Identifying Candidate Salivary Oral Cancer Biomarkers: Accurate protein quantification and analysis on LTQ type mass spectrometers

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Getiria Innocent Onsongo

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Dr. John V. Carlis & Dr. Timothy J. Griffin

May, 2011

Acknowledgements

I am grateful to all those without whom it would not have been possible to write this doctoral dissertation, to only some of whom it is possible to give particular mention here.

Above all, I would like to thank my mother who encouraged me to pursue a doctoral degree despite circumstances that made getting a job an appealing alternative. My father, sister and brothers for their unequivocal support throughout my graduate school career especially my older brother Auka who did the heavy lifting that enabled me to pursue a doctoral degree.

This dissertation would not have been possible without the tremendous help and support of my principal advisor, Prof. John V. Carlis. This guidance includes writing and presentation help. Also, I am thankful for the support and encouragement of my co-advisor Prof. Timothy J. Griffin. The access to resources in the Griffin laboratory, are those that most bioinformatics researchers can only dream of. The ability to conduct experiments with the help of post-doctoral researchers in the lab to test and evaluate computational techniques was invaluable to the success of my thesis work.

I owe my deepest gratitude to Rinal Ray and my close group of friends (chales). They have been my family away from home. I am thankful for Rinals patience, personal support and encouragement at all times. Most of all, Rinal was an excellent pro bono editor-in-chief. The phone calls and emails with words of encouragement from Edward Donkor, Kagabo Ngiruwonsanga and Fui Tsikata helped me stay focused. Additionally, Daniel Osei-Kuffuor was a great partner in crime through the Ph.D. program.

I am thankful to Shana Watters for showing me the ropes and helping me adjust to graduate school.

I most grateful to my friends and colleagues Bridget McInnes, Ebbing de Jong, Sricharan Bandhakavi and Matthew D. Stone who had been through the process and gave me great moral support and encouragement.

Members of Carlis and Griffin labs: Susan Van Riper, Daniel Feldman and Joel Kooren, thank you. I would also like to acknowledge the system staff in the Department of Computer Science and Engineering at the University of Minnesota, Twin Cities for their technical support. Special thanks to Kent Mein who offered tremendous technical support.

Last but not least, I would like to thank Libby Shoop, my undergraduate honors advisor for introducing me to the area of bioinformatics and encouragement to pursue a doctorate in Computer Science.

Dedication

To my loving mother and role model, Pacifica Moraa Ogeto.

Abstract

Cancer is one of the leading causes of death worldwide accounting for around 13% of all deaths. Oral cancer in one of the more common cancers occurring more frequently than leukemia, brain, stomach, or ovarian cancer. Unfortunately, the 5-year survival rate for oral cancer has not significantly improved in the past 30 years and remains at approximately 50%, in part, due to lack of reliable diagnostic biomarkers for early detection. It is estimated, if diagnosed and treated early, survival rates for oral cancer would significantly improve to between 80% and 90%. We need reliable reliable biomarkers for diagnosis and early detection of oral cancer.

Recent developments in high-throughput proteomics techniques have made it possible to detect and identify low abundance proteins in complex biological fluids such as saliva. These low-abundance proteins could be a source of the elusive reliable biomarkers needed to improve survival rates for oral cancer. Limiting the widespread use of these proteomics techniques is lack of an accurate protein relative quantification technique.

A typical high-throughput experiment identifies several thousand proteins with several hundred differentially abundant proteins. The cost of validating candidate biomarkers prevents validation of each differentially abundant protein to identify promising candidate biomarker. We need computational techniques to identify promising candidate biomarkers.

This two-part dissertation presents: 1) a new technique for accurate protein relative quantification implemented in freely-available, open-source software (LTQ-iQuant) and 2) relational database operators for analyzing differentially abundant proteins to identify promising candidate biomarkers.

Linear ion trap mass spectrometers, such as the hybrid LTQ-Orbitrap, are a popular choice for isobaric-tags based shotgun proteomics because of their advantages in analyzing complex biological samples. Coupled with orthogonal fractionation techniques, they can be used to detect low abundance proteins extending the range for detecting possible biomarkers. Limiting the widespread use of this combination for quantitative proteomics studies is lack of a technique tailored to LTQ type instruments that accurately reports protein abundance ratios, and is implemented in an automated software pipeline. This thesis presents a new technique implemented in a freely-available, open source software that fulfills this need.

A major limitation of existing computational techniques when using high-throughput techniques is results that are too broad to be practically useful. A lot of the ‘potential’ disease-specific biomarkers discovered have been found not to be specific to the disease being studied. They either belong to biological categories that change in response to infection or tissue injury, or are proteins whose changes are induced by other stresses such as medication and diet. This thesis extends the relational database engine to enable use of biological pathways to identify promising candidate biomarkers. Using biological pathways to analyze high-throughput data avoids results that are too broad to be practically useful.

Protein differential abundance often is the criteria used to identify candidate biomarkers in high-throughput discovery-based biomarker studies. However, protein quantity by itself might not be the salient marker parameter. Protein function is often dependent on post-translational modifications such as phosphorylation and gylcosylation. By only using differential abundance to identify candidate biomarkers, we are limiting our ability to identify reliable biomarkers. We further develop new operators that in addition to using user specified pathways, use post-translational modification information to analyze high-throughput data. For the first time, we demonstrate feasibility of using post-translational modifications with relational database operators to analyze high-throughput proteomics data.

Collectively, this work will facilitate the search for reliable biomarkers. LTQ-iQuant will make LTQ instruments and isobaric peptide tagging accessible to more proteomics researchers providing a new window into complex biological fluids. Relation operators will provide a systematic way of bridging the gap between unbiased data driven approach and hypothesis driven approach to prioritize candidate biomarkers.

Contents

Acknowledgements **i**

Dedication **iii**

Abstract **iv**

List of Tables **ix**

List of Figures **xii**

1 Introduction **1**

2 Relative Protein Quantification **13**

2.1 Introduction 13

2.2 Background 14

2.3 Related Work 17

2.4 Experimental Procedures 21

2.4.1 Cell culture and sample preparation 21

2.4.2 Mass Spectrometry Analysis and Database Searching: MALDI TOF/TOF 4800 22

2.4.3 Mass Spectrometry Analysis and Database Searching: LTQ-Orbitrap XL 23

2.4.4 Computing Trend Lines 25

2.5 Quantification technique 26

2.5.1 Effect of reporter ion intensity on accuracy of reported ratio . . . 26

2.5.2 Developing an accurate quantification technique 31
2.5.3 Assessing statistical significance 34
2.5.4 Summary of quantification technique 38
2.6 Evaluating new quantification technique 39
2.6.1 Comparison of the new quantification technique to Mascot . . . 39
2.6.2 Comparison of LTQ-Orbitrap data to 4800 MALDI TOF/TOF data 40
2.7 Software Implementation 45
2.8 Discussion 47

3 Relational Database Operators 51
3.1 Operators for Binary Relation Biological Data 51
3.1.1 Illustration Data 52
3.1.2 Operator Overview 56
3.1.3 Transitive Edge Operators 57
3.1.4 Path Edge Operators 78
3.1.5 Rank Node 88
3.1.6 BuildGoSlim 96
3.2 Operators for Biological Reaction Data 101
3.2.1 Data Model 101
3.2.2 Illustration Data 104
3.2.3 Operator Overview 106
3.2.4 Compute Transitive Start Node - Node Pair 106
3.2.5 Compute Start & End Node Restricted Reaction Path 108

4 Analysis of High-throughput Proteomics Data 110
4.1 Analysis of proteomics data using *BuildGoSlim* 110
4.1.1 Related Work 111
4.1.2 Application 112
4.2 Identifying promising candidate biomarkers 116
4.2.1 Background 117
4.2.2 Related Work 118
4.2.3 Application 120

5 Conclusion and Future Work **132**

References **141**

Appendix A. Mapping Reactome **159**

Appendix B. Reaction Data Operators **167**

Appendix C. ComputeMinDistTransitiveEdge **171**

C.0.4 ComputeMinDistTransitiveEdge: Modified transitive closure algorithm 172

C.0.5 Computing Shortest Transitive Edge Relation: *MinJoinLogarithmicTC* 174

C.1 Experiment 179

C.2 Discussion and Conclusion 182

Appendix D. Data Tables **184**

List of Tables

2.1 Comparison Table of Protein Quantification Software and Techniques . 20

2.2 Checking normality 38

3.1 Sample data for high-throughput experiment data measuring protein abundance levels in different conditions (disease states). We purposefully did not include *Cell Growth* in this sample data to better reflect experimental conditions where some proteins are not identified by the high-throughput technology being used. 53

3.2 Differentially abundant proteins between pre-malignant and malignant states 53

3.3 Edge relation for Figure 3.3 56

3.4 Protein in pathway associated with cancer development 56

3.5 **Compute Transitive Edge Operators:** Operators used to identify protein-protein interactions 58

3.6 Abbreviated short names together with their corresponding proper result relation names for precedence charts A1 and A2. 61

3.7 Result relation of using relations *edge* (Table 3.3) input relation to *Compute MinDist Transitive Edge* 66

3.8 Result relation of using relations *edge* (Table 3.3) and 2 as input parameters to *Compute Distance Restricted Transitive Edge* 68

3.9 Result relation of using relations *edge* (Table 3.3), *StartNode* (Table 3.2) and 2 as input parameters to *Compute Distance & StartNode Restricted Transitive Edge* 74

3.10 Operator Summary: Compute Path Edge operators 79

3.11 Result relation of using relations *edge* (Table 3.3), *StartNode* (Table 3.2) and 2 as input parameters to *Compute inNode Restricted Path Edge* . . 89
3.12 Operator Summary: Rank Node 90
3.13 Transitive edges for differentially abundant proteins interacting with the cell growth pathway 95
3.14 Result relation for computing number of paths, mean distance and rank for differentially abundant proteins connected to cell growth 95
3.15 Result relation of using (Table 3.3, Table 3.2 and Table 3.4) as the input parameters to *rank node* 95
3.16 Schema for GRAPH_PATH 97
3.17 Schema for GOSLIM 97
3.18 Schema for ASSOCIATION 97
3.19 *Edge* 102
3.20 *Node* 102
3.21 Relations for Compound Edge Relational Model in Figure 3.9 103
3.22 Relations for Compound Edge Relational Model in Figure 3.9 using Reactions (a) and (d) in Figure 3.10 as sample data. 105
3.23 Operators for analyzing high-throughput data using reaction data. . . . 107
3.24 Result relation for *Compute Transitive Start Node - Node Pair* when using *Edge Participation* and *Start Node* as inputs. 108
3.25 Start & End Node Restricted Reaction Path 109
4.1 Result relation of using Rank Node to prioritize differentially abundant soluble proteins in oral cancer study 124
4.2 Result relation of using *compute path edge* to find each path with P02679 as Start Node and protein P31946 as End Node and distance less than or equal to 2 125
4.3 Annotation information of prioritized list of candidate biomarkers obtained from published literature (www.uniprot.org) 126
4.4 Differentially abundant proteins present in Reactome database 129
4.5 Select attributes and rows from the result relation of *Compute Transitive Start Node & Node pair* 130
D.1 Pathways in Reactome associated with cancer development. 184

D.2 Transitive Edges for Differentially Expressed Soluble proteins interacting with pathways associated with cancer development. 186

D.3 Soluble Saliva Proteins Differentially expressed between pre-malignant and malignant oral lesions. 188

List of Figures

2.1 iTRAQ Reagent 15

2.2 *Simultaneous comparison of multiple states using iTRAQ Reagent* . . . 16

2.3 Computing Trend Lines 25

2.4 *Experimental set up illustrating analysis of peptides from a yeast whole cell lysate labeled with iTRAQ reagents and mixed to known ratios of 10:5:2:1 for, respectively, reporter ion masses of 114, 115, 116 and 117. It was designed to examine effect of intensity on accuracy of reported ratios and accuracy of quantification techniques by comparing reported ratios to the expected ratio* 27

2.5 **Relationship between intensity and reported peptide ratio for peptides combined at a known ratio of 2:1**. X-axis is $log_2(I_{114} \times I_{115} \times I_{116} \times I_{117})$ and Y-axis is $log_2(\frac{I_{114}}{I_{115}})$ and $log_2(\frac{I_{116}}{I_{117}})$ where $I_{114}, I_{115}, I_{116}$ and I_{117} correspond to the intensities of 4-plex iTRAQ reporter ions. Horizontal line represents log_2 of the expected ratio, 2:1 ($log_2[2]$).**Trend line 1** is a non-linear least squares fit of the form $y = 2.24 \times e^{(\frac{x}{69.47})}$ fitting ratios greater than 2 and **Trend line 2** is a non-linear least squares fit of the form $y = 0.40 \times log_2(\frac{x}{10.12})$ fitting ratios less than 2. 28

2.6 **Relationship between intensity and reported peptide ratio for peptides combined at a known ratio of 5:1**. X-axis is $log_5(I_{114} \times I_{115} \times I_{116} \times I_{117})$ and Y-axis is $log_5(\frac{I_{114}}{I_{116}})$ and $log_5(\frac{I_{115}}{I_{117}})$ where $I_{114}, I_{115}, I_{116}$ and I_{117} correspond to the intensities of 4-plex iTRAQ reporter ions. Horizontal line represents log_5 of the expected ratio, 5:1 ($log_5[5]$). **Trend line 1** is a non-linear least squares fit of the form $y = 1.64 \times e^{(\frac{x}{41.68})}$ fitting ratios greater than 5 and **Trend line 2** is a non-linear least squares fit of the form $y = 0.41 \times log_5(\frac{x}{0.55})$ fitting ratios less than 5. **NOTE**: log_5 was used in place of log_2 (in Figure 2.5) in order to have a plot with the expected ratio represented by the horizontal line y=1. 29

2.7 **Relationship between intensity and reported peptide ratio for peptides combined at a known ratio of 10:1**. X-axis is $log_1 0(I_{114} \times I_{115} \times I_{116} \times I_{117})$ and Y-axis is $log_1 0(\frac{I_{114}}{I_{115}})$ and $log_2(\frac{I_{116}}{I_{117}})$ where $I_{114}, I_{115}, I_{116}$ and I_{117} correspond to the intensities of 4-plex iTRAQ reporter ions. Horizontal line represents $log_1 0$ of the expected ratio, 10:1 ($log_1 0[10]$).**Trend line 1** is a non-linear least squares fit of the form $y = 1.45 \times e^{(\frac{x}{39.42})}$ fitting ratios greater than 10 and **Trend line 2** is a non-linear least squares fit of the form y=0.019x + 0.68) fitting ratios less than 10. **NOTE**: log_{10} was used in place of log_2 (in Figure 2.5) in order to have a plot with the expected ratio represented by the horizontal line y=1. 30

2.8 *Graph showing bins on X-axis, containing 100 different peptide abundance ratios each, in order of bins containing the peptides with the lowest product of reporter ion intensity values (bin 1) to the highest (bin 7). The Y-axis shows the respective weights assigned to each bin and used to generate a weight matrix.* 32

2.9 *Graph showing analysis used to determine optimal bin size. X-axis is number of proteins, Y-axis is bin size. A value of 100 produced the highest number of proteins quantified with less than 10% error.* 33

2.10 *A normal Q-Q plot comparing normal distributed data on the vertical axis to a standard normal population on the horizontal axis.* 36

2.11 *A normal Q-Q plot comparing exponentially distributed data on the vertical axis to a standard normal population on the horizontal axis.* 36

2.12 *A normal Q-Q plot comparing isobaric based peptide tagging data on the vertical axis to a standard normal population on the horizontal axis.* . . 37

2.13 *A normal Q-Q plot comparing normal distributed data on the vertical axis to a standard normal population on the horizontal axis.* 37

2.14 **Comparison of our quantification technique to quantification techniques offered by Mascot.** *Comparison of $\frac{I_{114}}{I_{115}}$ abundance ratios determined by each technique (2:1 expected ratio). Our technique is denoted as* ***"Intensitybased Weighted Avg"****. The other three techniques shown are those offered by Mascot.* 41

2.15 **Comparison of our quantification technique to quantification techniques offered by Mascot.** *Comparison of $\frac{I_{114}}{I_{117}}$ abundance ratios determined by each technique (10:1 expected ratio). Our technique is denoted as* ***"Intensitybased Weighted Avg"****. The other three techniques shown are those offered by Mascot.* 42

2.16 *Experimental set-up comparing LTQ-Orbitrap plus our technique to 4800 MALDI TOF/TOF plus Protein Pilot for protein quantification* 43

2.17 *Correlation analysis on proteins identified and quantified by both the LTQ-Orbitrap and 4800 MALDI TOF/TOF.* 44

2.18 **Correlation analysis on proteins identified and quantified by both the LTQ-Orbitrap and 4800 MALDI TOF/TOF**. *Protein abundance ratios with P value < 0.05 quantified by both our technique and Protein Pilot. Protein abundance ratios are plotted in ascending order of magnitude as measured by Protein Pilot from the 4800 MALDI TOF/TOF data (black dots) along with corresponding abundunce ratio for the same protein as measured by pur technique from LTQ-Orbitrap data (crosses)* 46

2.19 **Relative Quantification Software.** *A diagram showing the different components combined to form the software pipeline.* 48

3.1 Relational schema for binary relations 52

3.2 A portion of the gene ontology database. 54

3.3 Sample pathway (part of EGF pathway) 55

3.4 Execution times of *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases 76

3.5 Number of rows in intermediate JOIN relations for *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases . . . 77

3.6 Results for *BuildGoSlim* when using Figure 3.2 as sample Gene Ontology database, and the terms *developmental process, meristem maintenance, death* and *muscle attachment* as sample *GO Slim* terms. 101

3.7 Edge-Node Relational Model 102

3.8 STAT3 Activation 102

3.9 Compound Edge Relational Model 103

3.10 Part of EGF signaling pathway 104

4.1 Comparison of *post-DRC saliva* and *untreated saliva* using *BuildGoSlim*. *post-DRC saliva* denotes whole saliva proteins identified using hexapeptide beads for dynamic range compression. *untreated saliva* denotes whole saliva proteins identified without use of hexapeptide beads. 114

4.2 Comparison of HNSCC TIF and healthy buccal epithelia cell lysate from brush biopsy using the cellular components Gene Ontology terms. Identified proteins of HNSCC TIF (*solid bars*) and epithelial cell lysate (*open bars*) were categorized into different cellular localizations. Proteins with names associated with cytosol or cytoskeleton were grouped as cytoplasmic. Proteins with localizations for nucleus, mitochondria, the endoplasmic reticulum, the Golgi complex, peroxisomes, or lysosomes were grouped as organellar. Several proteins were grouped into more than one bin. Proteins with no Gene Ontology information or no clear localization were grouped in unclassified. 116

4.3 Illustration showing use of pathway and PPI data to determine proteins associated with known cancer pathways. 122

C.1 Sample Graph 173

C.2 LogarithmicTC 175

C.3 MinJoinLogarithmicTC 180

C.4 Execution times of *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases 181

C.5 Number of rows in intermediate JOIN relations for *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases . . . 181

Chapter 1

Introduction

Cancer, a complex and heterogeneous disease caused largely by abnormalities of the epigenome [1], is the second leading cause of death after heart diseases in the United States (US). Over half a million deaths are expected from cancer in the US alone in 2010[2]. If current trends continue, cancer will be the leading cause of death in the US in the 21st century [3]. These numbers are even greater worldwide where cancer is the leading cause of death accounting for 7.4 million deaths (around 13% of all deaths) in 2004 [4].

Historically there have been predictions of imminent success in finding a cure for cancer. In the early 1900s Dr. Roswell Park and his associates predicted that the discovery of the cause of cancer was "*just around the corner*" [5]. They established a cancer research institute in New York in 1898, the first instance in history of government involvement in cancer research. More recently, in 1971, in its report to congress advocating appropriation of funds to the National Cancer Institute (NCI), The Committee on Labor and Public Welfare concluded that cancer researchers "*are within striking distance of achieving the basic understanding of cancer cells*". This report led President Nixon to sign the National Cancer Act of 1971 into law that substantially increased funding to NCI [6]. Though measured in his optimism, in an article published in *SCIENCE* in 1986 the Nobel Laureate Renato Dulbecco predicted "*the next generation can look forward to exciting new tasks that may lead to a completion of our knowledge about cancer, closing one of the most challenging chapters in biological research*" [7].

This optimism together with the current trend in cancer deaths has led to significant resources being devoted towards finding a cure for cancer. Since 1971 NCI alone has spent over $105 Billion on cancer research. This is in addition to the amount spent by other government institutions such as the U.S National Institutes of Health (NIH) and does not include money spent by private companies and foundations. In the early 1970s, NCI accounted for nearly two-thirds of total cancer research funding in the US. By 1997, NCI provided less than half of total cancer research funding (46%) with industry's share steadily increasing from 2 percent in 1974 to 31 percent in 1997 [8]. At the MD Anderson Cancer Center, for example, Federal grants and contracts constitute less than 40% of their research budget and a fraction of the total amount spent on cancer related projects [9].

Unfortunately, despite this large allocation of resources towards cancer research, little progress has been made as measured by the increased proportion of deaths caused by cancer. Adjusted for age and size the death rate for cancer dropped a mere 5 percent between 1950 and 2005. In contrast, during the same period the death rate for heart diseases dropped by 64 percent and for cerebrovascular diseases by over 74 percent [2]. These outcomes led Bailar et al in a paper published in *The New England Journal of Medicine* to conclude "*the effect of new treatments for cancer on mortality has been largely disappointing*" [10].

Nevertheless, this allocation of resources has resulted in more cancer related studies as measured by the number of cancer related publications. In 2010 alone, cancer research worldwide resulted in about 40,000 published papers and is growing at about 2% per year [11]. These studies have helped to tremendously improve our understanding of cancer. From the naive assumption that cancer is an infectious disease [5], we now know it is a heterogeneous class of diseases caused primarily by environmental factors. 90 - 95% of cancer cases are due to lifestyle and environmental factors that lead to epigenetic changes and only about 5 - 10% of cancers are due to genetics [12]. This new knowledge has helped change the trajectory of cancer research. The initial approach of trying to find a single cure for cancer ("*silver bullet*") is increasingly being replaced by a multi-prong approach that includes encouraging lifestyle changes and improved early detection [13].

Encouraging lifestyle changes to reduce the risk of getting cancer significantly reduces cancer deaths. Tobacco use is responsible for over seventy percent (70%) of lung cancer deaths [14], the most common type of cancer worldwide, leading to an estimated 1.3 million deaths a year. A reduction in cigarette smoking in the US led to one of the biggest successes in primary prevention of cancer, a reduction of nearly 40% in cancer death rate in men between 1990 and 2006 [2, 13]. Therefore, encouraging lifestyle changes such us quiting smoking significantly reduces cancer deaths.

Similarly, improved early detection leads to significantly better prognosis [13]. If detected early, while still localized, survival rates for cancer improve drastically. For example, the 5-yr survival rate for breast cancer patients with, at initial diagnosis, early-localized disease is 98% compared to 27% for those whose cancer has already progressed to distant metastasis. For patients with Melanoma of the skin, the situation is even more distressing. The 5-yr survival rate for patients with, at initial diagnosis, early-localized disease is 98% as compared with a dismal 15% for those whose cancer has already progressed to distant metastasis [15, 16]. For colorectal cancer, the situation is equally as bad. When detected in its earliest stages, the survival rate is as high as 95% but drops drastically to only about 7% if its detected after it has metastasized [17]. Early detection is therefore critical to improved prognosis and has become a central pillar in cancer prevention.

To improve early detection, we need better diagnostic molecular markers, known as **biomarkers** [13]. "*A biomarker is a measurable indicator of a specific biological state, particularly one relevant to the risk of contraction, the presence or the stage of disease*" [18]. Elevated levels of the protein prostate-specific antigen (PSA), for example, often indicate the presence of prostate cancer and is used as a biomarker to detect the presence of prostate cancer before it progresses to distant metastasis (early detection).

Early detection and 5 yr survival rates have improved in a number of cancers. For leaukemia 5-year survival rate has improved from 42% in mid 1970s to 66% [15]. The situation is even better with breast cancer where 5-year survival rate has improved from 65% in the early 1960s to 89% [15]. For oral cancer, however, the situation remains grim. Its 5-year survival rate has not significantly improved in the past 30 years and remains at approximately 50% [19]. We need reliable biomarkers, or panel of biomarkers, for early detection of oral cancer.

Fortunately, recent advancements in high-throughput technology and methodologies are expanding the range of detection in complex biological fluids, such as saliva, providing new tools for discovery of biomarkers. Previously, characterization of proteins in complex biological fluids was limited to high abundance proteins due to the large *dynamic range* problem. *Dynamic range* is the ratio in amounts between the most abundant and least abundant compound in a fluid. In saliva, for example, 10 proteins account for nearly 98% of total salivary protein [20]. These high-abundance proteins obscure detection of low abundance proteins [21] that could be potential reliable biomarkers. This is similar to how *Burj Khalifa*, the tallest building in the world as of December 2010, would make it difficult to distinguish the difference in height between two people standing next to it when viewing the earth from outer space.

Based on the LTQ line of mass spectrometers [22, 23], Griffin et al developed a novel three-step peptide fractionation strategy (IEF/strong cation exchange/reverse-phase) that helps overcome the dynamic range problem [24]. This novel technique is more sensitive and capable of identifying low-abundance proteins providing a new window into complex biological fluids thus presenting a new tool for identification of potential biomarkers.

Not-withstanding the dynamic range problem, saliva has unique properties that make it an optimal fluid for oral cancer biomarker discovery. It is easy to collect in relatively large amounts from patients in a non-invasive manner making it an ideal bodily fluid for clinical studies [25, 26, 27]. For cancers of the oral cavity, the close proximity and direct contact of affected tissues with salivary fluid makes it an obvious choice for biomarker discovery studies [27]. Additionally, there is increasing evidence that saliva contains biomarkers that show changes with oral cancer progression [28, 29].

This combination of novel high-throughput mass spectromerty based proteomics techniques and saliva as a source of candidate biomarkers provides a unique opportunity for identifying reliable biomarkers for oral cancer progression that is yet to be fully exploited. Preventing adoption of this combination in biomarker discovery studies is lack of software for automated protein quantification needed to identify promising candidate biomarkers and computational techniques to prioritize these promising candidate biomarkers for follow up validation studies[30]. Such software would meet the following criteria: 1) be compatible with centroided LTQ tandem mass spectrometry data; 2) employ a technique accounting for reporter ion intensities, critical for accurate protein quantification [22, 23, 31, 32]; and 3) packaged in a freely-available and flexible software pipeline. No available software currently meets these criteria.

When using high-throughput techniques in biomarker discovery studies, it is not uncommon to obtain a list of a few hundred differentially abundant proteins as candidate biomarkers. Because of practical limitations, it is not possible to validate each candidate biomarker necessitating computational techniques to prioritize them [33]. Existing computational techniques have one major drawback that limits their utility in prioritizing candidate biomarkers. They produce results that are too broad to be practically useful [34]. For example, an analysis of proteins identified from a collection of tumor interstitial fluid from a head and neck squamous carcinoma [35] using Ingenuity Pathway Analysis (Ingenuity Systems, www.ingenuity.com) reveals over 25 molecular and cellular functions significantly associated with the dataset. These functions include *Protein Degradation* and *Carbohydrate Metabolism* which have no documented specificity to tumor development. This drawback has been attributed to the disappointing low number of reliable biomarkers despite an exponential increase in number of proteins identified as ‘potential’ biomarkers [36].

This thesis presents a software, LTQ-iQuant, which meets the criteria for an ideal software for automated protein quantification on LTQ line of instruments. It also presents relational database operators for analyzing high-throughput proteomics data using biological ontologies and biological pathways to avoid results that are too broad to be practically useful.

LTQ-iQuant implements a new technique that accurately reports protein abundance ratios. Using an iTRAQ-labeled standard mixture, this new technique was compared to the commercial software Mascot, the best available option for quantifying isobaric peptide data on the LTQ line of instruments. LTQ-iQuant performed better than Mascot’s non-weighted averaging and median peptide techniques, and equal to its weighted averaging technique. LTQ-iQuant was also compared to Protein Pilot on the 4800 MALDI TOF/TOF, the *defacto* instrument and software standard for iTRAQ-based proteomics, by analyzing an iTRAQ-labeled stem cell lysate. These results illuminated two points. First, for proteins quantified by both instruments, results obtained with the 4800 MALDI TOF/TOF and Protein Pilot were comparable to results with the

LTQ-Orbitrap and LTQ-iQuant, validating the accuracy of this new technique. Second, the comparison showed that LTQ-Orbitrap and LTQ-iQuant identifies and quantifies significantly more proteins than comparable analysis on the 4800 MALDI TOF/TOF. This finding is especially significant since the 4800 instrument is currently considered the best option for large-scale iTRAQ-based quantitative analysis [37]. A key advantage of LTQ-iQuant is the capability of users to input their own training data, enabling customization to individual instrument performance, different isobaric tagging methods (e.g., TMT) and possibly even emerging instrumental operation methods (e.g., Orbitrap HCD [38, 39] or ETD [40, 41]). In summary, LTQ-iQuant should make the powerful combination of isobaric peptide tagging and the LTQ line of instruments attractive in a wide-variety of quantitative proteomic studies such as biomarker discovery.

When using high-throughput proteomics techniques such as those based on the hybrid LTQ-Orbitrap in discovery-based biomarker studies, it is necessary to use computational techniques to analyze results. These techniques produce lists of hundreds or a few thousand differentially abundant proteins as candidate biomarkers. To identify the most promising candidate biomarkers additional validation analyses are needed. These validation analyses, such as western blots, are expensive and time-consuming necessitating prioritizing techniques. In addition to cost of reagents and time spent by skilled researchers such as post-doctoral fellows to validate candidate biomarkers, samples used for validation are invaluable. Often these samples, especially those for different diseased states, are available in very limited amounts that cannot be replaced limiting the total number of validations that can be done.

Additionally, in these discovery-based studies, data is analysed without a biological hypothesis with the aim of identifying patterns or proteins that usefully discriminate among groups of persons with different diagnosis, prognosis or response to therapy[42, 43, 44]. Results obtained often have no clear biological meaning. Consequently, it is imperative that researchers use appropriate techniques with the necessary statistical power capable of distinguishing real observed changes from random acts of chance. This is particularly significant in high-throughput studies where proteins are simultaneously analyzed for differential abundance (multiple hypothesis testing problem [45]). In these studies, the probability of a type I error (false positives) where proteins are wrongly identified as being differentially differentially abundant is significantly higher [46].

Confounding the problem of multiple hypothesis testing is the issue of insufficient sample size [47]. Typically in a single experiment hundreds or even thousands of proteins are simultaneously examined as potential predictors for a small number of outcomes. To maintain the same statistical power as that of a similar experiment where only one gene or protein is being examined the sample size needs to be increased [48]. In clinical epidemiology, a parallel problem, the rule of thumb is to have at least 10 events for each variable being examined. This translates to roughly 10 samples per gene or protein in a high-throughput experiment suggesting tens of thousands of samples are needed which is clearly not practical [49].

Several techniques have been developed that purport to overcome the problem of small sample size in high-throughput studies [50, 51, 52]. However, it has been shown these techniques are highly dependent on the objective of an experiment and do not generalize to all high-throughput experiments [49]. Additionally, when performing multiple hypothesis testing correction, it is important to factor in dependence between genes or proteins [53]. Considering our knowledge of biological functions is largely incomplete, especially with respect to cancer development, one cannot ascertain dependence between genes or proteins and hence it is practically impossible to apply these techniques in an approariate manner. Furthermore, some of these statistical techniques are not applicable to the quantitative proteomics techniques in question [24], with great promise as tools for biomarker discovery [35, 29]. This quantitative proteomics technique pools samples resulting in loss or reduction of biological replicates which is characteristically different from microarray experiments where different chips are used for different samples.

It is therefore no surprise that despite an exponential increase in the number of proteins that have been discovered and presented as 'potential' biomarkers for specific diseases, the number of US Food and Drug Administration (FDA) - approved protein tests is decreasing [54, 55]. Most of this proteins do not progress beyond the initial discovery phase [17] and often do not hold on subsequent studies.

A historical look at epidemiology, a more mature field that encountered similar problems of uncertain and often contradictory results, might help tackle this paradox. Epidemiologists study patterns of health and illness and associated risk factors at the population level [56] while biomarker researchers study patterns of illness and associated risk factors at the individual level. Similar to biomarker discovery where purported candidate biomarkers often fail to progress past the initial discovery stage, epidemiology had a history of numerous studies claiming to have identified an association between a risk factor and the development of a disease only to be disputed by subsequent studies. For example, a study of electric utility workers in the United States in 1995 suggested a possible link between electomagnetic fields (EMF) from power supplies and brain cancer which contradicted previous studies done in Canada and France that found no link between EMF and cancer [57]. This was not an isolated instance. Taubes et al [57] lists other numerous past examples of contradicting epidemiological studies.

Because of this uncertainity, most prominent journals such as *SCIENCE* and the *New England Journal of Medicine* adopted a rule of thumb for epidemiological studies in the early 1990s where before taking any study seriously, it would have to show a very strong association between disease and a risk factor and a highly plausable biological mechanism. This rule of thumb was adopted after the realization that use of sophisticated mathematical and statistical techniques to control for uncertainity and effects of biases cannot compensate for the limitation of the data [57]. This limitation of the data is clearly demonstrated by the controversy of *carcinoembryonic* (CEA) antigen biomarker.

In a study done in 1969, CEA was claimed to be nearly 100% sensitive and 100% specific to colorectal cancer screening diagnosing 35 out of 36 patients known to have colorectal cancer [58]. Subsequent research, however, showed significantly different outcomes with less promising results. CEA was not as effective a biomarker as initially thought[59, 60]. This non-reproducibility of initial CEA results was a result of the type of samples used in the initial experiment [42, 61]. Individuals initially studied had extensive cancer whereas individuals who were later studied were in earlier stages of cancer development. The type of sample used in a study thus significantly affects results and the ability to generalize those results. Often "*ideal samples*" that would make it possible to discriminate among groups of persons using statistical techniques are not available resulting in use of less than ideal samples, a limitiation of the data.

Because of these practical limitations e.g., lack of *"ideal"* samples or lack of enough number of samples, adopting a rule of thumb similar to that adopted by epidemiologists might help overcome this paradox of decreasing number of FDA approved biomarkers with an increase in the number of ‘potential’ biomarkers. In addition to developing sophisticated mathematical and statistical techniques to tease out proteins and biomarkers that usefully discriminate among groups of persons with different diagnosis, prognosis or response to therapy we need tools that identifier biomarkers with a highly plausable biological mechanism.

This thesis presents relational operators that summarize results of high-throughput studies and identify proteins with known association to a specific disease development thus providing a tool for inferring plausible biological mechanism. They utilize biological ontologies, user specified pathways and pathway reaction data in pathway databases. Pathways are selected based on their known association to the disease or condition being studied.

Using pathways with known association to the disease or condition being studied has one other major advantage. Majority of the ‘potential’ disease-specific biomarkers discovered so far have been found not to be specific to the disease being studied [36]. Many have been found to either belong to biological categories that change in response to infection or tissue injury or are proteins induced by other stresses such as medication and diet and may have absolutely no relationship to the disease of interest. By using pathways with known association to a given disease, we exclude these proteins whose changes are triggered by immune system response or other stresses such as diet and medicine.

Several pathways have been identified as playing key roles in development of complex diseases such as cancer [62, 63, 64, 65] and have previously been studied as therapeutic targets for diseases supporting their use in prioritizing candidate biomarkers. Based on this observation that genes or proteins responsible for development of cancer are expected to interact with disease causing pathways, computational tools capable of elucidating interaction between candidate biomarkers and pathways can be used to identify and prioritize biomarkers [33]. To this end, we developed operators that utilize user specified pathways to prioritize candidate biomarkers.

To test these operators, we analyzed a dataset of salivary proteins differentially expressed between pre-malignant and malignant oral lesions. Six proteins were identified as candidate biomarkers worth of validation studies. A literature search reveals these high priority candidate biomarkers interact with proteins implicated in cancer development highlighting their potential utility as biomarkers demonstrating the effectiveness of these operators [33].

Protein differential abundance often is the criteria used to identify candidate biomarkers in high-throughput discovery-based biomarker studies. However, protein quantity by itself might not be the salient marker parameter [54]. Protein function is often dependent on phosphorylation, gylcosylation, other post-translational modifications, location in the cell and/or the location of the tissue and other gene products such as microRNAs. Recent studies have shown protein post-translational modifications and microRNA expression levels can lead to disease development [66, 67]. Kruck et al showed increased phosphorylation rather than protein overexpression leads to activation of mTOR in renal cell carcinoma [66]. Garzon et al make a case for targeting microRNAs in anticancer therapies based on their ability to concurrently target multiple effectors of pathways involved in cell differentiation, proliferation and survival [67]. By only using differential abundance to identify candidate biomarkers, we are limiting our ability to identify reliable biomarkers.

We further develop new operators that in addition to using user specified pathways, use reaction data such as enzymes, RNA and post-translational modification information to identify and prioritize candidate biomarkers. Reactome contains post-translational modification information and information about other gene products such as RNAs that can be used to identify and prioritize candidate biomarkers. Using EGFR signaling pathway, a commonly targeted pathway by anticancer drugs [68], we test the functionality of these operators. For the first time, we demonstrate feasibility of using post-translational modifications and reaction data with relational database operators to analyze high-throughput proteomics data.

Use of database operators has several other advantages over developing application software. First they shift the burden of analysis to the database management system resulting in improved productivity and performance [69]. Second database operators enable execution of complex queries useful for prioritizing candidate biomarkers. Third database operators make it possible to repeatedly perform complex analysis enabling refinement of the prioritized list of candidate biomarkers. Finally with database operators, developed techniques can be easily integrated with application software.

In summary, the main contributions of this thesis are as follows: 1) operators for analyzing candidate biomarkers using biological pathways; 2) operators that enable manual validation of results obtained; and 3) a datamodel and operators that enable use of reaction data such as enyzmatic information and post-translational modifications to analyze candidate biomarkers.

Collectively, these operators will help overcome one of the main challenges of high-throughput computational techniques; provide a systematic way of bridging the gap between unbiased data driven approach and hypothesis driven approach to prioritize candidate biomarkers worth of more expensive and time consuming validation studies [33].

Disciplines are distinguished partly for historical reasons and reasons of administrative convenience ... We are not students of some subject matter, but students of problems. And problems may cut right across the borders of any subject matter or discipline. - **Karl Popper** [70]

This work was done in collaboration with researchers from Department of Biochemistry, Molecular Biology and Biophysics, Department of Oral Medicine, Diagnosis and Radiology, Department of Biomedical Informatics and Computational Biology, Department of Biostatistics, Department of Pediatrics, and Minnesota Supercomputing Institute. This broad interdisciplinary expertise provided essential functionalities without which development of an accurate quantification technique and validation of operators for prioritizing candidate biomarkers would not have been possible. For example, the *yeast standard mixture* used to confirm dependence of collective reporter ion intensity on accuracy of reported ratios was generated by a post-doctoral fellow in Dr. Griffins Lab [23].

The remainder of this dissertation is organized as follows. Chapter 2 presents LTQ-iQuant, a software that implements a technique for accurate protein relative quantification on LTQ line of instruments. First, it presents background material needed to understand relative protein quantification on LTQ line of instruments using isobaric tags. Second, related work on protein relative quantification and how it compares to LTQ-iQuant is discussed. Third, experimental procedures for the different studies done to develop and evaluate LTQ-iQuant are outlined. Fourth the new technique that accurately quantifies proteins on LTQ line of instruments is presented together with implementation in a freely-available software pipeline, LTQ-iQuant. Finally, a discussion on specific contribution and impact of this work is presented.

Chapter 3 presents operators and a new datamodel for analyzing high-throughput proteomics data. This chapter first presents *Compute Transitive Edge* operators, *Compute Path Edge* operators, *Rank Node* operator and *BuildGoSlim* operator. These operators use biological graph data to analyze high-throughput proteomics data. A datamodel that represents biological reactions as multiple input, multiple output and possible multiple co-edge nodes together with new operators (*Compute Transitive Start Node - Node Pair* and *Compute Start & End Node Restricted Reaction Path*) that use this datamodel are presented. This new datamodel captures information lost when using binary protein-protein interactions to represent biological pathways.

Chapter 4 evaluates operators presented in Chapter 3. Chapter 5 discusses the specific contributions and the overall conclusion of this dissertation. As with any research, this work has generated new research questions. Chapter 5 concludes by proposing future work to address this new research questions.

This thesis contains published work that has been reproduced with permission by Copyright ACM [33], Copyright IEEE [71] and Copyright Wiley-VCH Verlag GmbH & Co. KGaA [30]. Reused worked as been cited accordingly as per the copyright terms of use.

Chapter 2

Relative Protein Quantification

2.1 Introduction

Linear ion trap mass spectrometers such as the LTQ line of instruments offer sensitivity, versatility and reliability, making them a popular choice for shotgun proteomic studies [23, 22]. Coupled with a novel three step fractionation technique developed by Griffin et al [23], these LTQ line of instruments are more sensitive than techniques that use a two step fractionation technique. Collectively they provide a new window into complex biological fluids such as saliva capable of identifying low-abundance proteins. These low-abundance proteins could potentially be a source of the elusive reliable biomarkers needed to improve survival rate for oral cancer.

Limiting the widespread use of this powerful combination for large-scale quantitative proteomic studies is the lack of a technique for accurate protein relative quantification on LTQ line of instruments that is implemented in a freely-available and flexible software pipeline making it available to the wider proteomics community. This chapter presents a technique that accurately determines protein abundance ratios from LTQ-derived data and is implemented in a freely-available and flexible software pipeline called LTQ-iQuant. Such software would enable identification of candidate biomarkers using the sensitive, versatile and reliable LTQ line of instruments.

The remainder of this chapter is organized as follows. Section 2.2 presents background material needed to understand relative protein quantification on LTQ line of instruments using isobaric tags. Section 2.3 discusses related work and compares it to LTQ-iQuant. Section 2.4 contains experimental procedures for the different studies done to develop and evaluate LTQ-iQuant. Section 2.5 presents a new technique that accurately determines protein abundance ratios from LTQ-derived data as demonstrated by studies presented in section 2.6. Section 2.7 presents implementation details of this technique in LTQ-iQuant. Finally, section 2.8 discusses the contribution and potential impact of work presented in this chapter.

2.2 Background

For quantitative proteomic studies multiplexed isobaric peptide tagging reagents such as the iTRAQ reagent [72] or tandem mass tags [TMT] [73] offer flexibility and ease of use [74, 75]. Multiplexed isobaric peptide tagging is a technique used to quantify proteins from different sources in one single experiment. It uses isobaric reagents which consist of a peptide reactive group, a reporter group, and a balance group as shown in Figure 2.1. The peptide reactive group (PRG) covalently links the isobaric tag with lysine side chains and N-terminal group of a peptide (Applied Biosystems). The reporter group produces reporter ions used to estimate peptide amounts from different samples and the balance group ensures total mass of the reagents is the same (isobaric).

The isobaric nature of these tags is what enables quantification of proteins from different sources in one single experiment using *tandem mass spectrometry*. Tandem mass spectrometry, also know as MS/MS, refers to multiple steps of mass spectrometry analysis with some form of fragmentation occuring in between the steps [76]. For relative quantitative proteomics, these tags allow for labeling of samples with reagents of the same mass but after fragmentation in tandem mass spectrometry mode, gives rise to reporter ions with distinct masses. Proteins from different samples are lysed to their constituent peptides using the enzyme trypsin. Isobaric tags with different reporter ion masses are then attached to peptides from different samples. The tagged peptides are combined and analyzed using tandem mass spectrometry. Using trypsin avails peptides lysine side chains and N-terminal groups that react with the peptide reactive group to form a covalent bond.

Figure 2.2 illustrates use of 4plex iTRAQ for simultaneous comparison of salivary

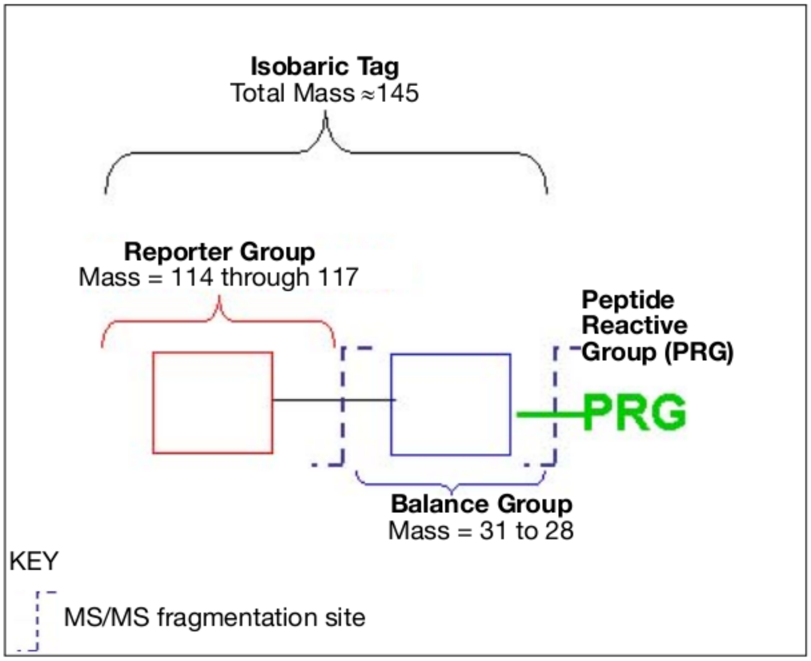
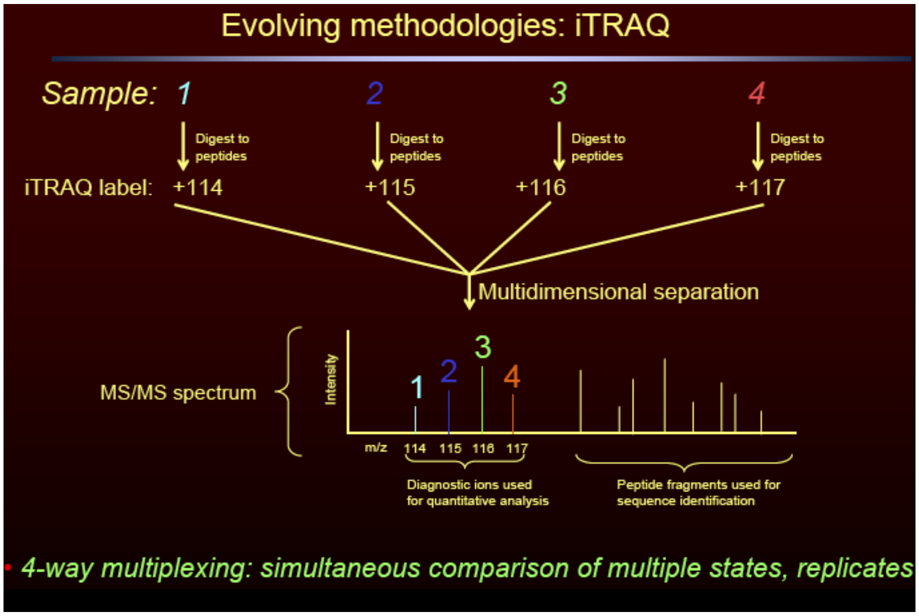


Figure 2.1: iTRAQ Reagent

protein amounts from different stages of oral cancer progression. Salivary protein samples from normal, pre-cancer, cancer and post-treatment patients are digested using trypsin and peptides labeled with, respectively, iTRAQ reagents with reporter ion masses of 114, 115, 116 and 117. The peptides are then combined and analyzed using tandem mass spectrometry. The first mass spectrometry analysis isolates a peptide from many entering a mass spectrometer. The second mass spectrometry analysis fragments the peptide. These fragments are characterized and used to identify peptides which are in turn used to identify proteins present in the samples (see " *Peptide fragments used for sequence identification* " in Figure 2.2).

In the second mass spectromety analysis, along with fragmenting and isolating amino acid fragments, isobaric tags are also fragmented separating the reporter group from the balancer group. Reporter ions from the reporter group are isolated and their ion counts used to estimate peptide amounts (see " *Diagnostic ions used for quantitative analysis* " in Figure 2.2). Because peptides from different sources are labeled using reagent with differing reporter ion masses, the relative amounts of these unique reporter ions are used



Evolving methodologies: iTRAQ

Sample: 1 2 3 4

Digest to peptides

iTRAQ label: +114 +115 +116 +117

Multidimensional separation

MS/MS spectrum

Diagnostic ions used for quantitative analysis

Peptide fragments used for sequence identification

- 4-way multiplexing: simultaneous comparison of multiple states, replicates

Figure 2.2: *Simultaneous comparison of multiple states using iTRAQ Reagent*

to estimate relative peptide amounts. Relative peptide amounts are then aggregated and used to infer relative protein amounts.

Until recently, use of LTQ line of instruments in shotgun proteomics had been limited due to the " *1/3 rule* ". The *1/3 rule* refers to a limitation of ion traps where the upper limit on the ratio between precursor mass-to-charge ratio (m/z) and the lowest trapped fragment ion is approximately 0.3. In practical terms, fragment ions of a parent ion with m/z 900 will not be detected if their m/z is below 300 [77]. To see why this limitation of ion traps poses a challenge in iTRAQ-based quantitative proteomics, consider the average length of short trypic peptides is about 8.4 residues [78]. The peptide HLKTEAEMK, with 9 residues, has an average molecular weight of 1086.28 and a monoisotopic weight of 1085.55 [79]. According to the 1/3 rule, fragments for the singly charged peptide HLKTEAEMK with m/z below 363 cannot be detected excluding use of iTRAQ reagents (with total mass of 145) for isobaric based quantitative proteomics on ion traps.

Development of pulsed Q dissociation (PQD) overcame the "1/3 rule" [80] for tandem mass spectrometry operation on the LTQ, enabling detection of low *m/z* fragments derived from isobaric tagged peptides. Several groups have shown the effectiveness of LTQ operating with PQD for quantitative proteomic studies using isobaric peptide tagging [23, 81, 31]. Additionally, PQD operation on the hybrid LTQ-Orbitrap which offers improved mass accuracy and resolution compared to the standard LTQ has been shown to be effective for quantitative proteomics [22, 32]. Unfortunately, limiting the widespread use of isobaric peptide tagging with LTQ instruments is lack of a technique that accurately determines protein abundance ratios from LTQ-derived data, and is implemented in an automated software pipeline.

2.3 Related Work

For accurate quantification from isobaric peptide tagging data, regardless of the mass spectrometer used for analysis, reporter ion intensities must be considered. Furthermore, this quantification technique needs to be made available to the wider proteomics community. Ideally such software would, at the very least, meet the following criteria: 1) be compatible with centroided LTQ MS/MS data; 2) employ a technique accounting for errors introduced by low reporter ion intensities, critical for accurate protein quantification [23, 31, 22, 32] ; 3) compatible with different isobaric tagging; and 4) be packed in a freely-available and flexible software pipeline that makes it amenable to individual instruments and possibly even emerging instrumental operation methods (e.g., Orbitrap HCD [38, 39] or ETD [40, 41]). No available software currently meets these criteria.

Table 2.1 shows a comparison of LTQ-iQuant, the software presented in this chapter, with existing techniques and software for relative protein quantification. As demonstrated, LTQ-iQuant is superior to these techniques and software. It is the only software that meets criteria for ideal software for relative protein quantification on LTQ line of instruments.

Multi-Q [82], i-TRACKER [83], and RelEx [84] are not compatible with centroided data and do not account for errors introduced by low reporter ion intensities. Multi-Q is distributed as an executable for the Windows platform. It uses weighted average to determine protein relative abundance ratios but the weight is determined by peptide abundance and not intensity of reporter ions. i-TRACKER was developed for peptide relative quantification and relies on an input parameter from users as a threshold to exclude peptides with low ion counts. RelEx uses least-squares regression to determine peptide ion ratios which are then averaged to determine relative protein abundance. It was written in Visual Basic and C for the Windows platform.

Scaffold Q+ (marketed by Proteome Software), Libra [85], and the in-house software developed by [86] do not account for errors introduced by low reporter ion intensities. Scaffold Q+ assigns protein abundance based on the median ratio from aggregated peptides. Libra and the in-house software developed by [86] average peptide abundance ratios to determine protein relative abundance. Libra is constrained to those using the Trans-Proteome Pipeline while the in-house software developed by Schulze et al was not intended as software for general use by others.

Andreev et al [87] present an algorithm designed to take advantage of high resolution, mass accuracy and throughput of the hybrid mass spectrometer LTQ-FT by filtering less reliable measurements of peptide abundances. Filtering less reliable measurements of peptides, however, has the potential of excluding large portions of a data set when quantifying proteins. Low abundance peptides, which tend to have higher relative variability and hence have less reliable measurements, dominate data sets [88] Filtering to exclude less reliable measurements of peptides is therefore bound to exclude a large portion of the data set in quantification.

Mascot and Protein Pilot are commercial software. Mascot, developed by Matrix Science, offers several techniques for quantifying isobaric peptide tagging data from LTQ data, but is only available to researchers who have purchased it. The current Protein Pilot software version sold by Applied Biosystems for the analysis of iTRAQ isobaric peptide tagging data [89] is not compatible with data generated from LTQ instruments.

Karp et al [88] recently described software employing a technique based on variance stabilizing transformation accounting for errors due to low reporter ion intensities; however it relies on the use of Mascot for quantifying LTQ data.

The Kuster group [22, 32] accounted for outlier ratios that may stem from low reporter ion intensities using a linear regression analysis across aggregated peptides to calculate overall protein abundance ratios along with a confidence estimate. This technique provided comparable quantitative results to the same sample analyzed on a time-of-flight instrument, demonstrating its accuracy. However, this technique was implemented using in-house scripts not intended as software for general use by others. Similarly, Schulze et al [86] develop an in-house software that works on centroided data. However, unlike the in-house software developed by the Kuster group [22, 32], it does not account for for errors due to low reporter ion intensities.

Finally, Griffin et al [23] minimized errors from peptides producing low intensity reporter ions using a technique wherein reporter ion intensities were summed across aggregated MS/MS spectra to calculate protein abundance ratios [23, 31]. This technique was shown by [90] to be optimal amongst the algorithms tested which included the least squares approach. It is simple, computationally efficient, incorporates intensity into the estimates of fold change thus accounting for errors introduced by low reporter ion intensities, does not fail when only few peptides are identified (no singularities), and consistently provided the best estimates of protein relative quantification [90]. Similar to the in-house scripts developed by the Kuster group, this technique was implemented using in-house scripts not intended as software for general use by others. Additionally, because peptide intensities were summed resulting in a single ratio, used to determine relative protein abundance, standard statistics such standard deviation commonly used to assess statistical significance of protein abundance could not be computed.

To address this limitation, I developed a technique for relative protein quantification from LTQ generated isobaric peptide tagging data wherein abundance ratios from each identified peptide sequence are weighted proportional to the collective intensities of their reporter ions. Aggregated, weighted peptide ratios are then used to obtain a weighted average abundance ratio and confidence value for each protein. This technique was implemented and packaged in a freely available software pipeline (LTQ-iQuant) that automates analysis of large-scale isobaric peptide tagging data generated by LTQ-type instruments. It supports the use of mzXML-formatted data and therefore compatible with data generated from a wide-variety of database search programs. Users have the ability to input their own data enabling generation of peptide weighting matrix customized to individual instrument performance. As previously stated, Table 2.1 shows a comparison of LTQ-iQuant with existing techniques and software that demonstrates its superiority.

Table 2.1: Comparison Table of Protein Quantification Software and Techniques

Software	Reference	LTQ-Orbitrap Compatible	Centroided Data Compatible	Accounts for Low Intensity Data	Computes Statistical Significance	Customizable	Open Source / Free	Platform Independent
Multi-Q	[82]				✓			
i-TRACKER	[83]						✓	✓
RelEx	[84]	✓			✓		✓	
Scaffold Q+	Proteome Software	✓	✓		✓			✓
Libra	[85]	✓	✓		✓		✓	✓
Andreev et al	[87]	✓	✓	✓	✓		✓	
Mascot	Matrix Science	✓	✓	✓	✓			✓
Protein Pilot	[89]			✓	✓			✓
Karp et al	[88]	✓	✓	✓	✓			✓
Kuster Group	[22]	✓	✓	✓	✓		✓	
Schulze et al	[86]		✓		✓		✓	
Griffin et al	[23]	✓	✓	✓				
LTQ-iQuant	**[30]**	✓	✓	✓	✓	✓	✓	✓

2.4 Experimental Procedures

2.4.1 Cell culture and sample preparation

S. cerevisiae cell culture for generating standard iTRAQ reagent labeled mixture

S. cerevisiae cultures were grown, cells were lysed, and proteins were extracted, digested with trypsin, followed by peptide labeling with iTRAQ reagent (AB Sciex, Foster City, CA) at known relative abundances as previously described [23].

Human mesenchymal stem cell (MSC) culture for generating iTRAQ reagent labeled peptide mixture used to compare LTQ-Orbitrap and MALDI 4800 TOF/TOF instruments

Tissue collection and processing was approved by the Committee on the Use of Human Subjects in Research at the University of Minnesota. Bone marrow was donated from healthy adult patients and filters were obtained, and MSCs obtained. Resulting cells were plated at 1000 cells/cm^2 in a 6-well dish and incubated in MSC media for 7 days. For Osteocyte differentiation, cultures were incubated with 3mL of MEM-alpha supplemented with 10% FBS, 100nM dexamethasone (Sigma, St. Louis, MO), 0.2mM Ascorbic acid (Sigma), 10mM beta-glycerol phosphate (Sigma), and 1X Pen-Strep (Invitrogen, Carlsbad, CA). Media was refreshed at 3-4 day intervals for 5 weeks. For adipogenesis, the cultures were incubated in 3mL/well of IMDM (Invitrogen) supplemented with 10% FBS (Hyclone), 10% horse serum (Sigma), 1μM dexamethasone (Sigma), 5μg/mL insulin (Sigma), 12mM Glutamax (Invitrogen), 50μM Indomethacin (Sigma), 0.5μM 3-Isobutyl-1-methylxanthine (IBMX; Sigma), and 1X Pen-Strep (Invitrogen). Media was changed every 3-4 days for 5 weeks.

Protein isolation, digestion, and iTRAQ labeling

Proteins were extracted in buffer containing 0.5M triethylammonium bicarbonate, 0.05% SDS and 0.1% Triton X100. Samples were treated with benzonase and the buffer was exchanged with multiple rinses with 0.5M triethylammonium bicarbonate using a Microcon YM-3 filter (Millipore, Billerica, MA). Proteins were processed for trypsin digestion and iTRAQ analysis according to manufacturers protocol with supplied reagents. Peptides from selected sample types were labeled at 100 μg with separate iTRAQ chemical tags: 114 tag for undifferentiated MSC cells; 115 and 116 tags for adipocytic differentiation; and 117 tag for osteocytic differentiation. The total peptide mixture was purified with an MCX cartridge (Waters, Milford, MA) before 2D LC.

2.4.2 Mass Spectrometry Analysis and Database Searching: MALDI TOF/TOF 4800

LC-MALDI-MS of iTRAQ reagent labeled samples from un-differentiated and differentiated stem cells

The iTRAQ reagent labeled peptide mixture from above was separated in the 1^{st} dimension off-line on a strong cation exchange column as described previously [91]; peptides were collected in 3-minute intervals. One third of 14 selected SCX fractions were further separated in the second dimension using a Tempo LC MALDI spotting system (AB Sciex). Peptides were separated by capillary LC with an in-house packed C18 column, spotted on two LC MALDI plates and analyzed by MALDI-TOF/TOF MS as previously described [92]. After the initial MALDI-MS acquisition was completed, inclusion lists were made from the precursor ions from each LC run. The precursor *m/z* values were used as exclusion lists during a second level of MS acquisition (parameters identical to original analysis).

Database searching and relative protein quantification using Protein Pilot

Tandem mass spectra were extracted and analyzed with Protein Pilot version 2.0.1 software (AB Sciex), which uses the Paragon scoring algorithm [93]; a list of inferred proteins and relative protein abundances were reported. The ratios of 114.1, 115.1, 116.1, 117.1 *m/z* values (“reporter ions ”) provided relative protein abundance among sample types for select proteins. The protein database used for searches was a concatenated “target-decoy” version of human subset of the NCBI Ref Sequence (http://www.ncbi.nlm.nih.gov/RefSeq/) database (June 27, 2008; 38,017 protein entries) to which 109 ‘contaminant’ proteins were added (source: http://www.thegpm.org/crap/index.html). A total of 76,252 protein sequences comprised the target-decoy database. Search parameters included: 4plex peptide mode; quantification; cysteine fixed methyl methanethiosulfonate modification; trypsin enzyme; thorough search mode (which includes semi-trypsin peptides during the search);

biological modifications (includes > 220 post-translational and artifactual modifications) and minimum detected protein threshold 66% confidence (or Unused Protein Score > 0.47; see [94], for description of unused Protein Score). “Reporter ion” area for each peptide was obtained from the Protein Pilot Peptide Export and used for further computational analysis.

2.4.3 Mass Spectrometry Analysis and Database Searching: LTQ-Orbitrap XL

LC-MS/MS of iTRAQ reagent labeled yeast standard and samples from undifferentiated and differentiated stem cells

For the iTRAQ labeled yeast standard sample, 0.2 μg of total peptides were analyzed. For the differentiated stem cell sample, an equivalent third of each of the 14 SCX fractions generated as described above were subjected to stage-tip [95] clean up. Stage-tips were prepared with 2 punches of Empore (3M, St. Paul, MN) high performance extraction disks using a 22-gauge needle. Peptide samples were diluted to 60μL with 0.1% TFA in water. After conditioning of the stage-tips, peptides were loaded, washed with 60μL with 0.1% TFA in water, and eluted with 60μL 20/80/0.1 water/ACN/TFA. Eluates were dried by vacuum centrifugation. Peptides were re-suspended in 5μL of load buffer (98/2/0.1 water/ACN/FA). They were loaded onto a similar C18 column as described above for the 4800 MALDI TOF/TOF analysis, but having a pulled tip instead of a frit, using a micro AS autosampler and a nano1D-LC HPLC (Eksigent, Dublin, CA) at a flow rate of 1μL/min with load buffer. Peptides were eluted over a 90 min linear gradient of 2 to 40% ACN in water and 0.1% FA at a flow rate of 0.25μL/min. The column was rinsed with 20/80/0.1 water/ACN/FA for 10 min and prepared for the next sample with rinsing with load buffer for 10 min. Peptides were analyzed by nano-electrospray using an LTQ-Orbitrap XL (Thermo Scientific, Waltham, MA). The spray voltage was 1.75 kV and capillary temperature was 160 °C. Full scans were obtained at 60,000 resolutions in the orbital trap over a range of 360 to 1800 m/z. The lock mass feature was enabled for the orbital trap using the m/z values of 371.1012, 445.1200, and 519.1388. The AGC values for the full scan were 500 ms or 1E^6 charges. The 5 most intense ions determined from the full scan were selected for fragmentation.

Fragmentation was performed with PQD using collision energy of 31%, activation time of 0.1 ms, and Q of 0.7 and data was obtained from 2 microscans. Dynamic exclusion lists were generated for up to 500 ions for 60s using an *m/z* range of -0.6 to 1.2. Charge state rejection was enabled for undetermined and +1 charge states. All MS/MS spectra were collected in centroid mode.

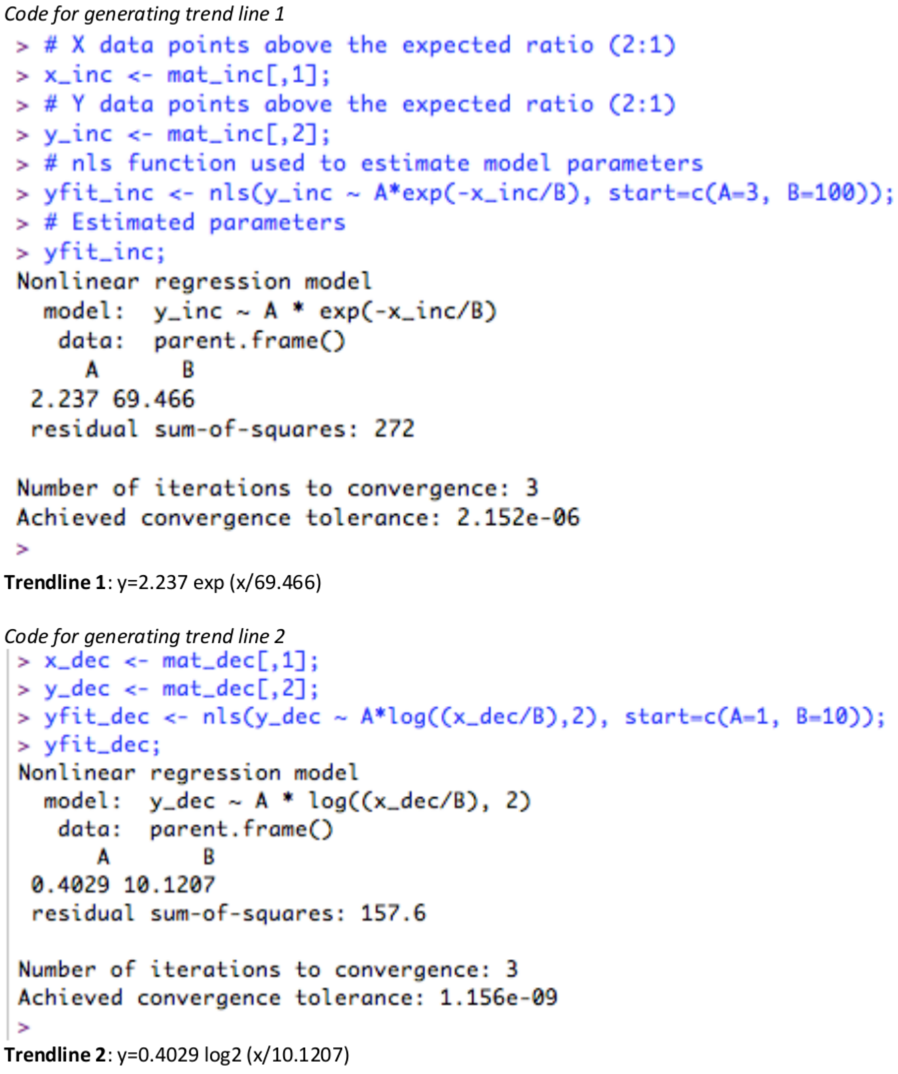
Database searching

mzXML files were created from .RAW files using ReAdW, .dta files extracted via extract_msn followed by database searching using Sequest v.27 rev. 12. For the yeast samples, MS/MS spectra were matched against peptide sequences from a database containing protein sequences translated from 6139 open reading frames in the *S. cerevisiae genome*, with a reversed-sequence version of the same database appended to the end of the forward version for the purpose of false discovery rate (FDR) estimations. For the differentiated stem cell samples, MS/MS spectra were searched against the same reversed database as above for the MALDI 4800 analysis. Parent mass tolerance and fragment mass tolerance were set to 1.0 and 0.8 amu respectively. Partial trypsin digestion with 2 missed cleavages was selected. Fixed modifications of iTRAQ reagent on Lys and N-terminal peptides and Methyl methanethiosulfonate (MMTS) on Cys and the variable modification of oxidation of Met were selected. Peptide probabilities from PeptideProphet were determined with Scaffold (Proteome Software) and identifications were filtered using 7 ppm precursor ion tolerance, 2 tryptic termini, and 5% peptide probability. For the yeast standard iTRAQ reagent labeled data used to develop and test our quantification technique, protein matches were filtered to an estimated peptide false positive rate below 1%.

For database searching of the iTRAQ labeled yeast standard mixture used for comparison with Mascot (Matrix Science, London, UK), a .mgf file was generated using Mascot Daemon v. 2.2.2. This data was searched using Mascot [96] using the same yeast database and modifications as above for the Sequest analysis. Trypsin was chosen as the digest enzyme and the precursor tolerance and fragment ions tolerances were set to 0.2 Da and 0.8 Da respectively. Protein quantification from reporter ions was done using either "average" (no weighting), "median" or "weighted" methods in Mascot. For each identified and quantified protein from the Mascot analysis, we extracted the relevant peptide and reporter ion data from the mzXML file used to generate the .mgf. We then analyzed this data using our intensity-based weighted average technique for our comparison to Mascot.

2.4.4 Computing Trend Lines

Figure 2.3 shows a session of the statistical package R captured to illustrate its use of non-linear least squares fit to compute trend lines for Figures 2.5, 2.6 and 2.7.



Code for generating trend line 1

```
> # X data points above the expected ratio (2:1)
> x_inc <- mat_inc[,1];
> # Y data points above the expected ratio (2:1)
> y_inc <- mat_inc[,2];
> # nls function used to estimate model parameters
> yfit_inc <- nls(y_inc ~ A*exp(-x_inc/B), start=c(A=3, B=100));
> # Estimated parameters
> yfit_inc;
Nonlinear regression model
  model:  y_inc ~ A * exp(-x_inc/B)
   data:  parent.frame()
     A      B 
 2.237 69.466 
 residual sum-of-squares: 272

Number of iterations to convergence: 3 
Achieved convergence tolerance: 2.152e-06 
>
```

Trendline 1: y=2.237 exp (x/69.466)

Code for generating trend line 2

```
> x_dec <- mat_dec[,1];
> y_dec <- mat_dec[,2];
> yfit_dec <- nls(y_dec ~ A*log((x_dec/B),2), start=c(A=1, B=10));
> yfit_dec;
Nonlinear regression model
  model:  y_dec ~ A * log((x_dec/B), 2)
   data:  parent.frame()
      A       B 
 0.4029 10.1207 
 residual sum-of-squares: 157.6

Number of iterations to convergence: 3 
Achieved convergence tolerance: 1.156e-09 
>
```

Trendline 2: y=0.4029 log2 (x/10.1207)

Figure 2.3: Computing Trend Lines

2.5 Quantification technique

2.5.1 Effect of reporter ion intensity on accuracy of reported ratio

For LTQ-derived isobaric peptide tagging data using PQD operation, relative abundance ratios calculated from lower intensity reporter ions are less accurate than those with higher intensities [23, 22, 31, 32]. Because low abundance peptides tend to dominate data sets [88], they cannot be discarded during quantification. Any protein quantification technique must therefore account for variability introduced by these lower intensity reporter ions. Before developing such a technique, one that considers ion intensities, we sought to confirm this relationship via asystematic analysis of peptides combined at different known amounts.

Tryptic peptides from a yeast whole cell lysate were labeled with iTRAQ reagents and mixed to known ratios of 10:5:2:1 for, respectively, reporter ion masses of 114, 115, 116 and 117. We refere to this combined mixture as the *yeast standard mixture*. MS/MS scans were performed in PQD mode using optimized instrumental parameters [23] and the data was processed as described in section 2.4. Results from this analysis were used to investigate the relationship between intensity of reporter ions and accuracy of reported ratios. Figure 2.4 illustrates this experimental set up.

As depicted in Figure 2.5, results of this experiment confirmed the relationship between reporter ion intensities and accuracy of measurement. Relative abundance ratios calculated from lower intensity reporter ions are less accurate than those with higher intensities. To generate Figure 2.5, we first needed to devise a means to categorize each peptide according to its collective reporter ion intensity. Each peptide was categorized based on the product of all four reporter ion intensities, which we deemed more effective than arbitrarily categorizing based on the intensities of individual reporter ions, where it is unclear as to which of the different reporter ions should be considered. Those peptides having relatively low intensities for all four reporter ions would be expected to have less accurate measurements. The X-axis represents the product of intensities on a log_2 scale, or $log_2(I_{114} \times I_{115} \times I_{116} \times I_{117})$, where $I_{114}, I_{115}, I_{116}$ and I_{117} correspond to reporter ion intensities of isobaric tags for peptides identified from our yeast standard mixture. The Y-axis represents measured abundance ratios (on log_2 scale) for $\frac{I_{114}}{I_{115}}$ and $\frac{I_{116}}{I_{117}}$, which are expected to be at ratios of 2:1 in our standard mixture. Each point

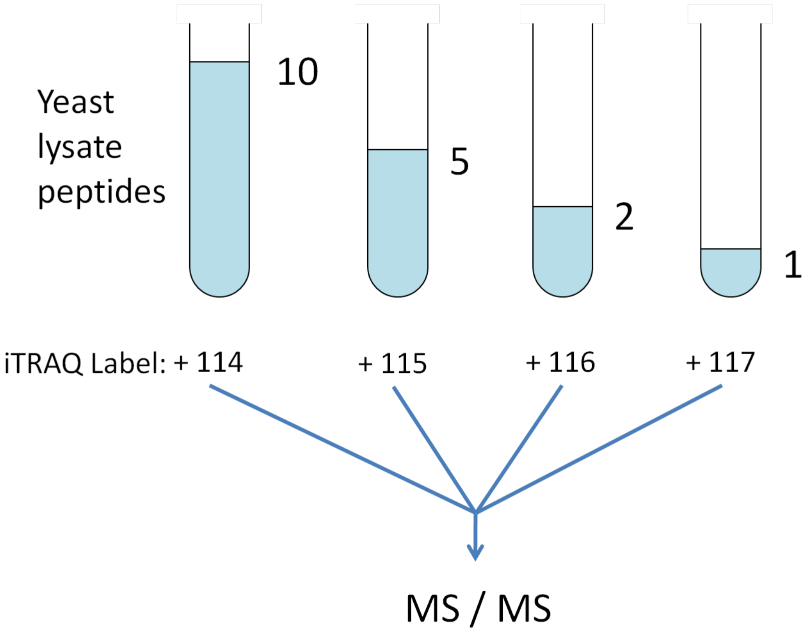


Figure 2.4: *Experimental set up illustrating analysis of peptides from a yeast whole cell lysate labeled with iTRAQ reagents and mixed to known ratios of 10:5:2:1 for, respectively, reporter ion masses of 114, 115, 116 and 117. It was designed to examine effect of intensity on accuracy of reported ratios and accuracy of quantification techniques by comparing reported ratios to the expected ratio*

represents one scan's $\frac{I_{114}}{I_{115}}$ or $\frac{I_{116}}{I_{117}}$ ratio. The **horizontal line** at y = 1 represents the expected value for each measured ratio ($log_2[2] = 1$). **Trend line 1** and **Trend line 2**, show that, with decreasing product of reporter ion intensities, measured peptide abundance ratios diverge from their known values. Section 2.4.4 describes how these trend lines were computed. Notably, a similar analysis plotting measured ratios from other combinations of iTRAQ reporter ions from the standard mixture against their expected ratios (5:1 or 10:1) showed similar results (Figures 2.6 and 2.7). These results indicate a consistent dependence of abundance ratio accuracy on the product of reporter ion intensities, regardless of the magnitude of the abundance ratio being measured (2:1, 5:1 or 10:1).

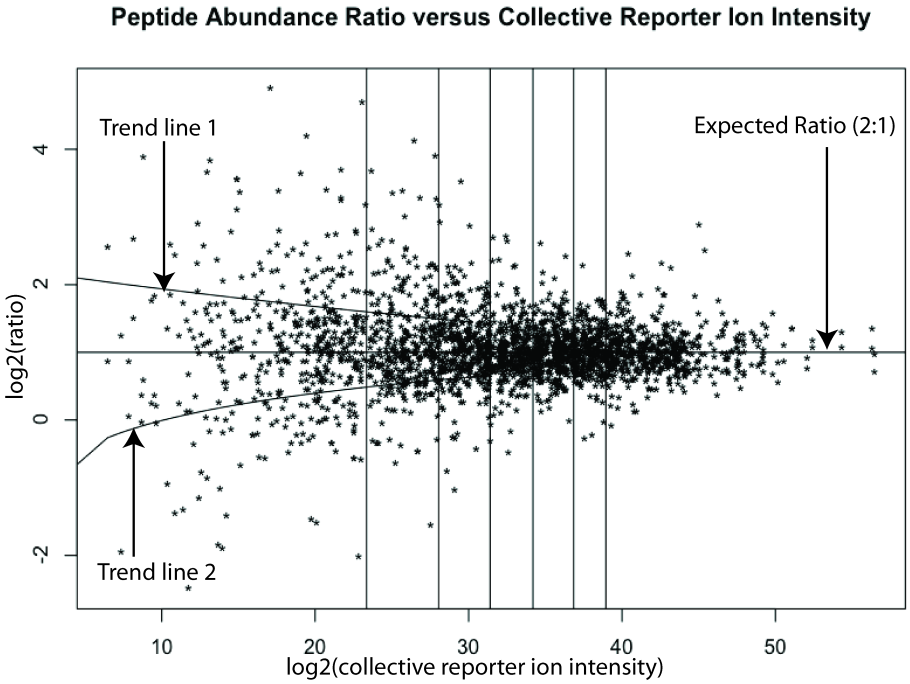


Figure 2.5: **Relationship between intensity and reported peptide ratio for peptides combined at a known ratio of 2:1**. X-axis is $log_2(I_{114} \times I_{115} \times I_{116} \times I_{117})$ and Y-axis is $log_2(\frac{I_{114}}{I_{115}})$ and $log_2(\frac{I_{116}}{I_{117}})$ where $I_{114}, I_{115}, I_{116}$ and I_{117} correspond to the intensities of 4-plex iTRAQ reporter ions. Horizontal line represents log_2 of the expected ratio, 2:1 ($log_2[2]$).**Trend line 1** is a non-linear least squares fit of the form $y = 2.24 \times e^{(\frac{x}{69.47})}$ fitting ratios greater than 2 and **Trend line 2** is a non-linear least squares fit of the form $y = 0.40 \times log_2(\frac{x}{10.12})$ fitting ratios less than 2.

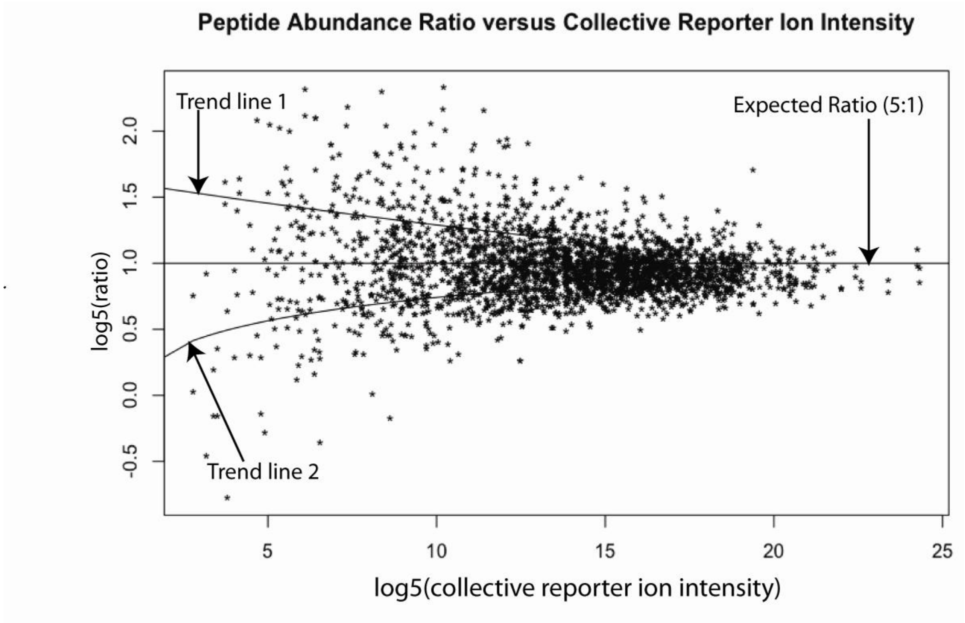


Figure 2.6: **Relationship between intensity and reported peptide ratio for peptides combined at a known ratio of 5:1**. X-axis is $log_5(I_{114} \times I_{115} \times I_{116} \times I_{117})$ and Y-axis is $log_5(\frac{I_{114}}{I_{116}})$ and $log_5(\frac{I_{115}}{I_{117}})$ where $I_{114}, I_{115}, I_{116}$ and I_{117} correspond to the intensities of 4-plex iTRAQ reporter ions. Horizontal line represents log_5 of the expected ratio, 5:1 ($log_5[5]$). **Trend line 1** is a non-linear least squares fit of the form $y = 1.64 \times e^{(\frac{x}{41.68})}$ fitting ratios greater than 5 and **Trend line 2** is a non-linear least squares fit of the form $y = 0.41 \times log_5(\frac{x}{0.55})$ fitting ratios less than 5. **NOTE**: log_5 was used in place of log_2 (in Figure 2.5) in order to have a plot with the expected ratio represented by the horizontal line y=1.

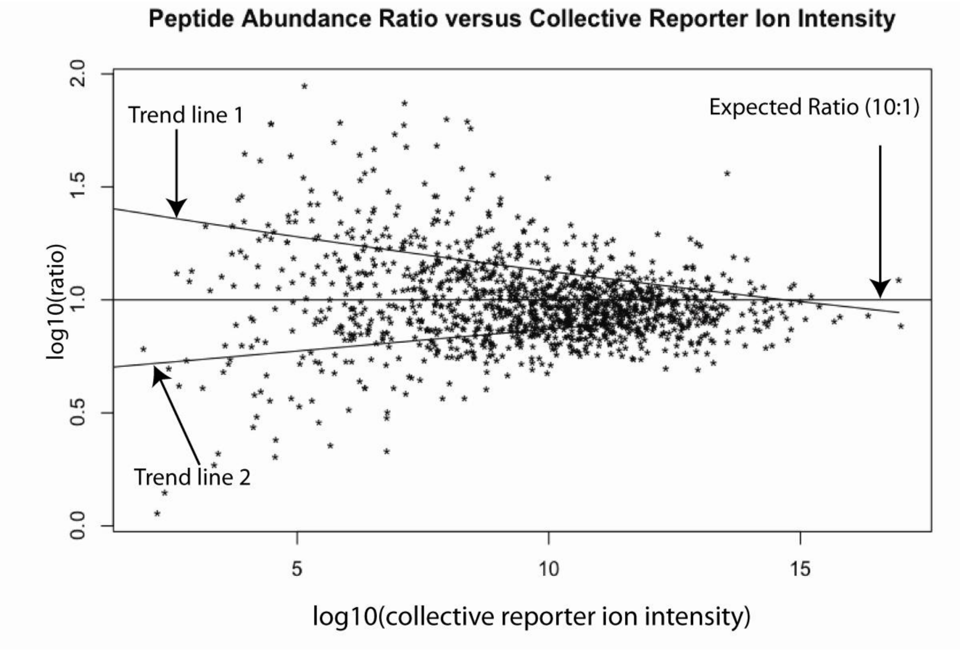


Figure 2.7: **Relationship between intensity and reported peptide ratio for peptides combined at a known ratio of 10:1**. X-axis is $log_10(I_{114} \times I_{115} \times I_{116} \times I_{117})$ and Y-axis is $log_10(\frac{I_{114}}{I_{115}})$ and $log_2(\frac{I_{116}}{I_{117}})$ where $I_{114}, I_{115}, I_{116}$ and I_{117} correspond to the intensities of 4-plex iTRAQ reporter ions. Horizontal line represents log_10 of the expected ratio, 10:1 ($log_10[10]$).**Trend line 1** is a non-linear least squares fit of the form $y = 1.45 \times e^{(\frac{x}{39.42})}$ fitting ratios greater than 10 and **Trend line 2** is a non-linear least squares fit of the form y=0.019x + 0.68) fitting ratios less than 10. **NOTE**: log_{10} was used in place of log_2 (in Figure 2.5) in order to have a plot with the expected ratio represented by the horizontal line y=1.

2.5.2 Developing an accurate quantification technique

Motivated by results of section 2.5.1, which confirm dependence of abundance ratio accuracy on product of reporter ion intensities, we sought to develop a protein quantification technique based on intensity weighting of peptide data. We devised a technique that assigns weights to peptide abundance ratios proportional to its product of reporter ion intensities, aggregates the peptides with their corresponding proteins, and then calculates a weighted average to obtain the protein abundance ratio thus accounting for variability introduced by low intensity reporter ions. To reduce the likelihood of observed abundance change being due to chance alone, a P-value is computed to assess statistical significance.

Assigning weights to peptide abundance ratios

To determine weights of reported peptide ratios based on intensity weighting of peptide data, a sample from the yeast standard mixture was independently analyzed and used to generate a weight matrix. Identified peptides were sorted from lowest to highest products of reporter ion intensities and grouped into bins each containing 100 identified peptides as shown by the **vertical lines** in Figure 2.5 and plotted in Figure 2.8. We tried different values for the bin size and found that 100 produced the best results (see Figure 2.9). The width of each bin was determined by the range of products of reporter ion intensities corresponding to the 100 peptides contained in the bin. For example, bin 1 in Figure 2.8 contained the peptides with the 100 lowest products of reporter ion intensities (first column from the left in Figure 2.5), bin 7 contained peptides with the 100 highest products of reporter ion intensities (last column in Figure 2.5). We measured how well the experimentally determined ratio for each peptide within a bin deviated from the expected ratio to assign an error, and averaged the square of the error across all 100 peptides to calculate a variance for each bin. Bins with low variance were assigned a higher weight, while bins with a high variance were assigned a lower weight. The Y-axis in Figure 2.8 shows the relative weights assigned to each bin. A 2-by-N weight matrix with the bin range in the first column and the weight of the bin in the second column was constructed. N is the total number of bins. This weight matrix is used to assign default weights to peptides when quantifying a sample whose protein relative abundance is unknown.

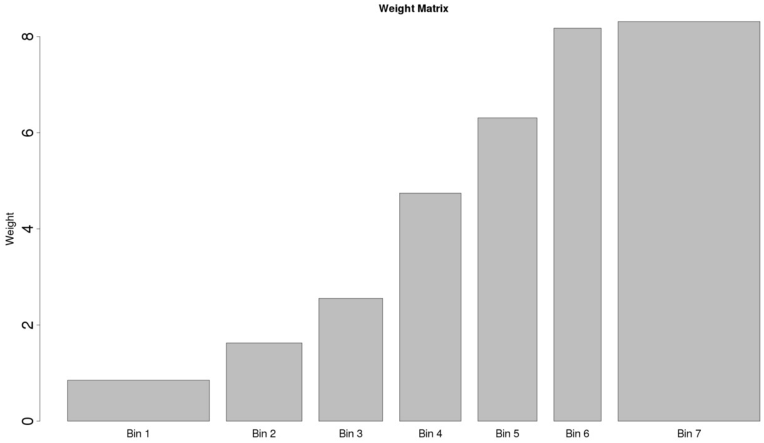


Figure 2.8: *Graph showing bins on X-axis, containing 100 different peptide abundance ratios each, in order of bins containing the peptides with the lowest product of reporter ion intensity values (bin 1) to the highest (bin 7). The Y-axis shows the respective weights assigned to each bin and used to generate a weight matrix.*

Before settling on use of a weight matrix to determine default weights for peptide ratios, I explored the use of a function but ultimately decided to use the weight matrix due to the reasons presented below. Looking at Fig 2.8, the weights assigned to each bin reflects an approximately linear relationship. An obvious question would therefore be, why bother using bins? The training data used to generated the weight matrix could be used to generate a linear function for assigning weights to peptides. However, using a linear function assumes all future data will follow this trend, an assumption that might not hold. Use of a weight matrix enabled implementation of software easily customizable to user's specific instrument in case their training data does not show a linear relationship, a key advantage of our software. If a function was used instead of a weight matrix, and a users training data deviates from a linear relationship, users would have to determine an alternative, more appropriate function to use instead of a linear function making the process of customizing the software much more complex.

Furthermore use of a function instead of a weight matrix will likely underestimate or overestimate weights of peptides in regions with few data points. The training data used was not evenly distrubuted along the X-axis and it is unlikely any training data set would be evenly distributed (see Figures 2.5, 2.6 and 2.7). As a result, function

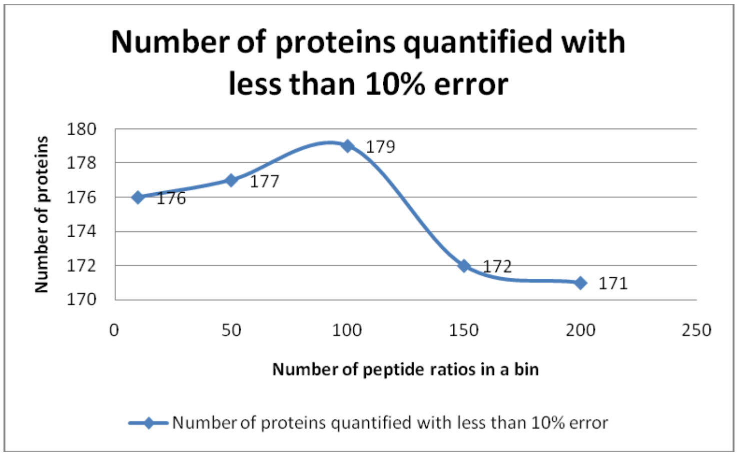


Figure 2.9: *Graph showing analysis used to determine optimal bin size. X-axis is number of proteins, Y-axis is bin size. A value of 100 produced the highest number of proteins quantified with less than 10% error.*

parameters tuned using these training data sets will reflect behavior of data points in high density regions which contain most of the data points. If data points in the low density region follow a different trend, the function will either underestimate or overestimate weights of peptides in these low density regions.

In contrast, the weight matrix is generated using bins whose weights are determined by data points in each bin. The weight assigned to any given bin is dependent on data points in the bin and independent of data points in other bins. Consequently, high density regions have no influence on weights assigned to bins in lower density regions.

Working in log scale

With a weight matrix in hand, it is tempting to immediately proceed to calculating weighted average to determine protein abundance. However, this would be a mistake. The standard approach for computing weighted average [97] does so in linear space. Working in linear space to aggregate peptide ratios fails the inversion test [89], an important property in differential abundance studies. To see why the inversion property is significant, consider the following example of three peptides identifying a protein with ratios of 10, 10 and 0.1, where the ratio is defined as (cancer/normal). In linear space, the average is 6.7 (average $= \frac{10+10+0.1}{3} = 6.7$) indicating abundance for this protein is increasing in the transition from normal to cancer cells. With the same peptides, if we want to instead find the ratio (normal/cancer), the inverse, we expect the ratio to be less than one ($ratio < 1$) indicating a decrease in protein abundance. Using the same approach, however, the average is not less than 1 (average = $\frac{0.1+0.1+10}{3} = 3.7$). In both cases protein relative abundance is increasing, a contradiction since one should be the inverse of the other. 6.7 is not the inverse of 3.7 ($6.7 \neq \frac{1}{3.7}$) and hence working in linear space fails the inversion test.

When transformed to log space, the average ratios are inverses of each other.

ratio = cancer/normal: average = $2^{(\frac{\log_2(10)\ +\ \log_2(10)\ +\ \log_2(0.1)}{3})} = 2.154435$

ratio = normal/cancer: average = $2^{(\frac{\log_2(0.1)\ +\ \log_2(0.1)\ +\ \log_2(10)}{3})} = 0.4641589$

Calculated in log space, the result of averages passes the inversion test ($2.154435 = \frac{1}{0.4641589}$).

Aggregating peptide abundance ratios to determine protein ratios

Having established averages calculated in log space pass the inversion test, we can now determine protein abundance ratios in log space. First, we compute peptide abundance ratios ($peptide\ abundance\ ratio = \log_2[\frac{I_{\text{reporter ion 1}}}{I_{\text{reporter ion 2}}}]$) where I = intensity. Second, we determine the weight for each peptide abundance ratio by calculating its collective reporter ion intensity and looking up its weight from the weight matrix. Third, we average the weighted peptide abundance ratios in $\log_2$ scale to obtain protein abundance ratios. Finally, we transform these ratios back into linear space to obtain ratios which are either greater than 1 if abundance is increasing or less than 1 if abundance is decreasing.

2.5.3 Assessing statistical significance

Calculating P value

To reduce the likelihood of observed abundance change being purely due to chance, we compute a P-value for each protein abundance ratio obtained by aggregating peptide abundance ratios. P-value is a standard statistical metric in hypothesis testing that measures the probability of a *null hypothesis* being true [89]. It ranges from 0 to 1 and the smaller it is, the less likely it is the null hypothes is true. In our case, the null hypothesis is *" protein abundance does not change between the two conditions being compared "*. This *null hypothesis* is **not** true when the difference in abundance is real. Therefore, a P value very close to 0 (typically less than 0.05) would lead us to reject the *null hypothesis* and conclude the difference in protein abundance is real and not a random event occuring due to chance. We use a two sided student t-test to compute P value.

Checking normality

The t-test statistic makes certain assumptions about the data being tested with the main one being the data follows a normal distribution. If this assumption does not hold, P values calculacted using t-test statistics are less informative and can even be misleading. Before employing use of students t-test statistic, we checked sample data from an actual experiment to assess normality of isobaric based peptide tagging data on LTQ type instruments. We used sample data from an iTRAQ experiment measuring protein abundance at different stages of oral cancer progression [24]. Failure of the isobaric based peptide tagging data to pass the normality test would suggest use of t-test statistic might not be the best approach to assess statistical significance. Alternate test statistics less sensitive to normality of data might be preferable.

To test for normality, we used quantile-quantile (Q-Q) plots, a graphical method for comparing two probability distributions by plotting their quantiles against each other [98]. Quantiles of data being tested (sample data) are plotted against the theoretical quantiles of the standard normal distribution. If the data being tested is a sample from a normal distributed population, the points being plotted will fall roughly along the line y = x. Figure 2.10 illustrates how data from a normal distribution is plotted on a normal Q-Q plot. It compares randomly generated, normal distributed data on the vertical axis to a standard normal population (theoretical normal distribured data) on the horizontal axis. The linearity of the points confirm the randomly generated, normal distributed data is indeed normally distributed.

To illustrate how a Q-Q plot can be used to identify normally distributed data, we plotted a randomly generated, exponential data (Figure 2.11) and displayed it next to the Q-Q plot of a normally distributed data (Figure 2.10). As Figure 2.11 shows, the points follow a strongly nonlinear pattern indicating the data is not from a normal

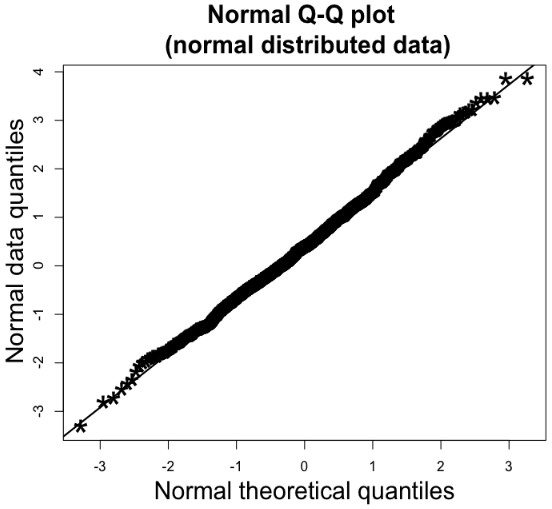


Figure 2.10: *A normal Q-Q plot comparing normal distributed data on the vertical axis to a standard normal population on the horizontal axis.*

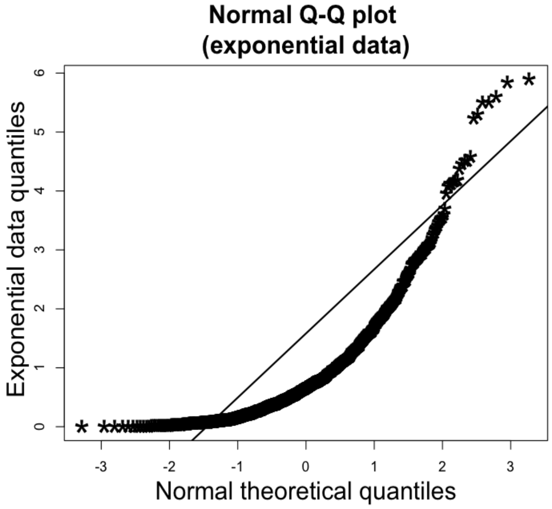


Figure 2.11: *A normal Q-Q plot comparing exponentially distributed data on the vertical axis to a standard normal population on the horizontal axis.*

distribution.

To test the sample data from an actual experiment, we computed a relative abundance ratio (cancer/healthy), plotted it on a Q-Q plot (Figure 2.12), and displayed it next to a standard normal distribution (Figure 2.10) for comparison. As the results show, the sample data from an actual isobaric based peptide tagging data experiment does follow a normal distribution. The distribution of points in Figure 2.12 is as linear as the distribution of points in Figure 2.10.

A graphical method for comparing two probability distributions such as the Q-Q plot has several advantages. It is easy to use, convenient to intepret and can be effective in detecting large deviations from the underlying assumption of normality. However, since the process relies heavily on visual interpretation, some subjectivity cannot be avoided and may, at times, lead to ambigious results especially in situations where the underlying distribution may be mildly skewed or symmetric but nonnormal [99]. To dismiss any glimmer of doubt as to whether or not the sample data from an actual

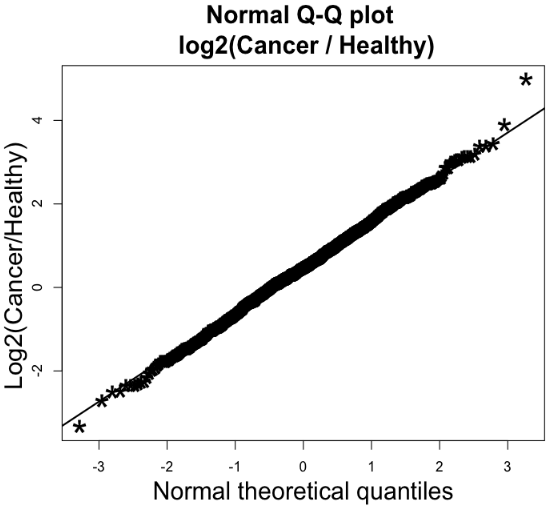


Figure 2.12: *A normal Q-Q plot comparing isobaric based peptide tagging data on the vertical axis to a standard normal population on the horizontal axis.*

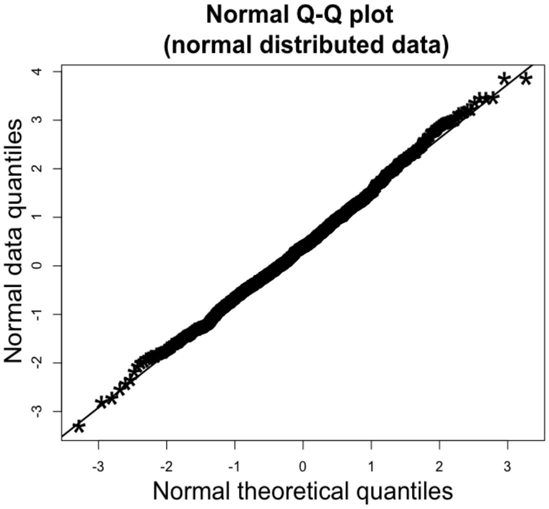


Figure 2.13: *A normal Q-Q plot comparing normal distributed data on the vertical axis to a standard normal population on the horizontal axis.*

isobaric based peptide tagging experiment follows a normal distribution, we tested the sample for normality using a non-graphical method, Shapiro-Wilk test.

Shapiro-Wilk test, published in 1965, tests the *null hypothesis* that " *a sample came from a normally distributed population* " [100]. If the resulting P value is less than the chosen α (e.g., 0.05 or 0.01), the null hypothesis is rejected concluding the data is not from a normally distributed population. We performed a Shapiro-Wilk test on the sample isobaric based peptide tagging data in Figure 2.12, exponential data in Figure 2.11) and normally distributed data in Figure 2.10. Table 2.5.4 shows results of these tests.

As table 2.5.4 shows, using the popular significance level ($\alpha = 0.05$), the null hypothesis cannot be rejected in both the sample data and normally distributed data. For the exponetial data, the P value is significantly smaller (2.2e-16) so we reject the null hypothesis confirming the data is not normally distributed. These results further confirming the sample isobaric based peptide tagging data passes the normality test.

Shapiro-Wilk Test		
Data	**W**	**P value**
Normal (Fig 2.10)	0.9976	0.168
Exponential (Fig 2.11)	0.8061	2.2e-16
Sample (Fig 2.12)	0.9988	0.7617

Table 2.2: Checking normality

2.5.4 Summary of quantification technique

In sections 2.5.1, 2.5.2 and 2.5.3, we present the technique developed for accurate protein relative quantification on LTQ line of instruments. Below, we summarize the main steps.

Step 1: Determine peptide abundance ratio

Peptide Abundance Ratio = $log_2 \frac{I_{reporter\ ion\ 1}}{I_{reporter\ ion\ 2}}$ where I = intensity.

Step 2: Determine weight of each peptide

Peptide Weight = $f(log_2(I_{114} \times I_{115} \times I_{116} \times I_{117}))$

where f = function that looks up weight from weight matrix.

Step 3: Determine protein abundance ratio in log_2 scale

Temp Protein Abundance Ratio = $\frac{\sum_{j=1}^{N} Peptide\ Abundance\ Ratio_j \times Peptide\ Weight_j}{\sum_{j=1}^{N} Peptide\ Weight}$

where N = number of peptides identifying a protein

Step 4: Determine protein abundance ratio

Protein Abundance Ratio = $2^{Temp\ Protein\ Abundance\ Ratio}$

Step 5: Calculate P value for protein abundance ratio

$\bar{x}$ = Weighted Average = Temp Protein Abundance Ratio

Var($\bar{x}$) = weighted variance

Z = $\frac{\bar{x}}{\sqrt{Var(\bar{x})}}$

P value = $pnorm(-|Z|) \times 2$

2.6 Evaluating new quantification technique

With our quantification technique in hand, we sought to compare the accuracy of using this technique to other techniques for protein quantification from isobaric peptide tagging data. This section presents results from these comparisons demonstrating the improved accuracy of our technique.

2.6.1 Comparison of the new quantification technique to Mascot

We started by comparing our quantification technique to that of the program Mascot, because it is arguably the best commercial option for quantifying LTQ-generated data, offering several different techniques for quantifying isobaric peptide tagging data: non-weighted averaging of aggregated peptide abundance ratios, use of median peptide abundance ratio for assigning protein abundance ratio, and weighted averaging based on summation of reporter ion intensities across aggregated peptides (see Mascot online Help manual, http://www.matrixscience.com/help/quant_statistics_help.html). Using our standard yeast mixture data from the LTQ-Orbitrap, we first used Mascot for sequence database searching to match MS/MS spectra to peptide sequences (see section 2.4), which were then aggregated by protein from which they are derived. For the comparison, we only considered proteins identified by three or more unique peptides. Using the same peptide data, we then quantified each protein using the different techniques offered by Mascot and our weighted averaging technique. We compared our results only for protein abundance ratios deemed statistically significant by Mascot and high confidence proteins quantified by our technique (P value < 0.05). We then determined the number of proteins quantified by each different technique in the standard mixture at varying relative errors ($\leqslant$10%, $\leqslant$20%, or $\leqslant$30%) when comparing measured abundance ratios to the expected ratios. We chose to compare results for the ratio of iTRAQ reporter ions 114:115 in our standard mixture (expected to be 2:1; Figure 2.14) and the ratio of reporter ions 114:117 (expected to be 10:1; Figure 2.15).

Examination of the results in Figure 2.14 reveals that our technique quantifies about the same number of proteins with high accuracy ($<$ 10% error) compared to Mascot's weighted averaging technique when measuring relatively small (2-fold) abundance differences. Our technique quantifies 50% more proteins than Mascot's median technique, and about 20% more than non-weighted averaging. When the error for quantified proteins is increased to 30%, the numbers of proteins quantified by each technique equalizes, indicating that the four techniques perform comparably when considering less accurate data.

Examination of the results in Figure 2.15 reveals that our technique also quantifies about the same number of proteins with high accuracy ($< 10\%$) as Mascot's weighting and median techniques when measuring a larger abundance difference (10-fold). The non-weighted averaging technique performs slightly worse than the other techniques at high accuracy. The 10:1 abundance ratio is the most challenging to measure accurately, due to the 10-fold less abundance from the iTRAQ 117 reporter ion, which will be detected with low signal-to-noise ratio in many of the MS/MS spectra. Similar to the comparisons in Figure 2.14, as the % error increases, all four techniques perform comparably, although both weighting techniques still quantify slightly more proteins than either the median or non-weighted averaging.

It should be noted that our technique has an advantage over the weighted average technique used by Mascot: we assign a P value as a measure of confidence of each quantified protein, while Mascot's technique does not. Assigned P values in our software implementation provide the user flexibility to apply different levels of stringency when interpreting their results without the need for re-analysis at different sensitivities (α values).

2.6.2 Comparison of LTQ-Orbitrap data to 4800 MALDI TOF/TOF data

Results in section 2.6.1 demonstrated that the technique used by LTQ-iQuant performs as well as the option offered by Mascot for quantifying isobaric peptide tagging data from LTQ instruments, providing accurate quantification of proteins in complex mixtures. Next, we sought to answer an important question: how does analyzing LTQ-Orbitrap iTRAQ data with LTQ-iQuant compare to analyzing 4800 MALDI TOF/TOF data with Protein Pilot, the current de facto standard for large-scale iTRAQ-based quantitative proteomics? Because the use of the LTQ and isobaric tagging with LTQ-iQuant represents a new method for quantitative proteomics, we felt a comparison to the currently accepted method for analysis of isobaric peptide tagging was warranted to further

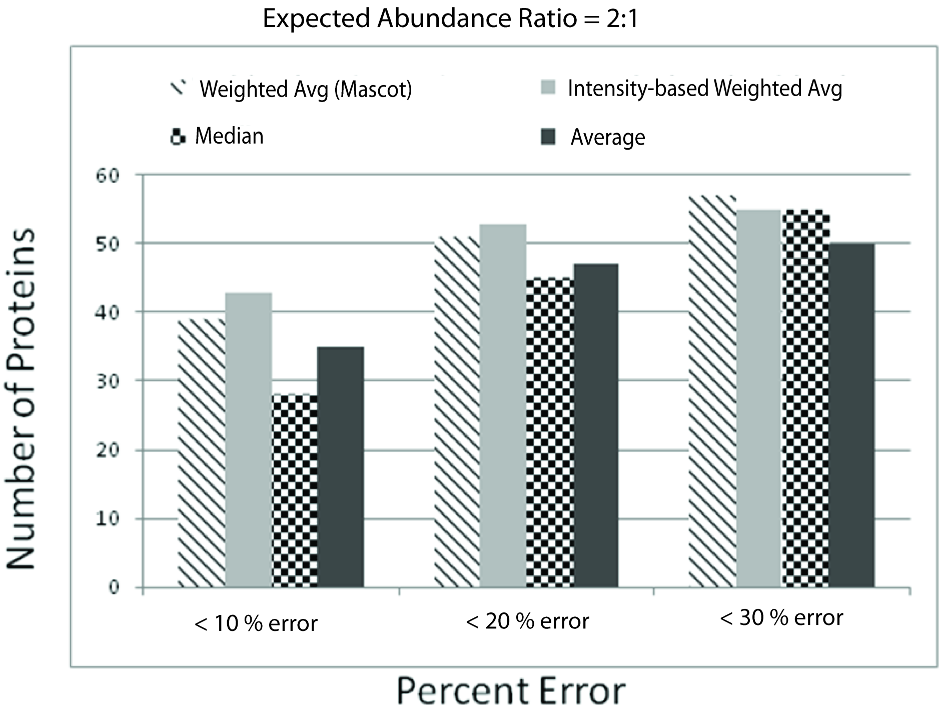


Figure 2.14: **Comparison of our quantification technique to quantification techniques offered by Mascot.** *Comparison of* $\frac{I_{114}}{I_{115}}$ *abundance ratios determined by each technique (2:1 expected ratio). Our technique is denoted as* ***"Intensitybased Weighted Avg"****. The other three techniques shown are those offered by Mascot.*

evaluate its performance. To that end we analyzed an iTRAQ labeled mixture comparing proteins derived from un-differentiated and differentiated stem cell lysates on both instruments using comparable analysis methods. Figure 2.16 shows this experimental setup. Details of the experiment are in section 2.4.

We first separately quantified the proteins identified on either instrument. Supplemental Table 1 provides all relevant information on the proteins identified from either instrument. For proteins identified by two or more distinct peptides, the LTQOrbitrap identified 1638 proteins while the MALDI 4800 TOF/TOF identified 657 proteins. The estimated protein FDR was 0% and 0.15%, for the LTQOrbitrap and MALDI 4800 TOF/TOF data, respectively. 630 proteins were identified in common between the

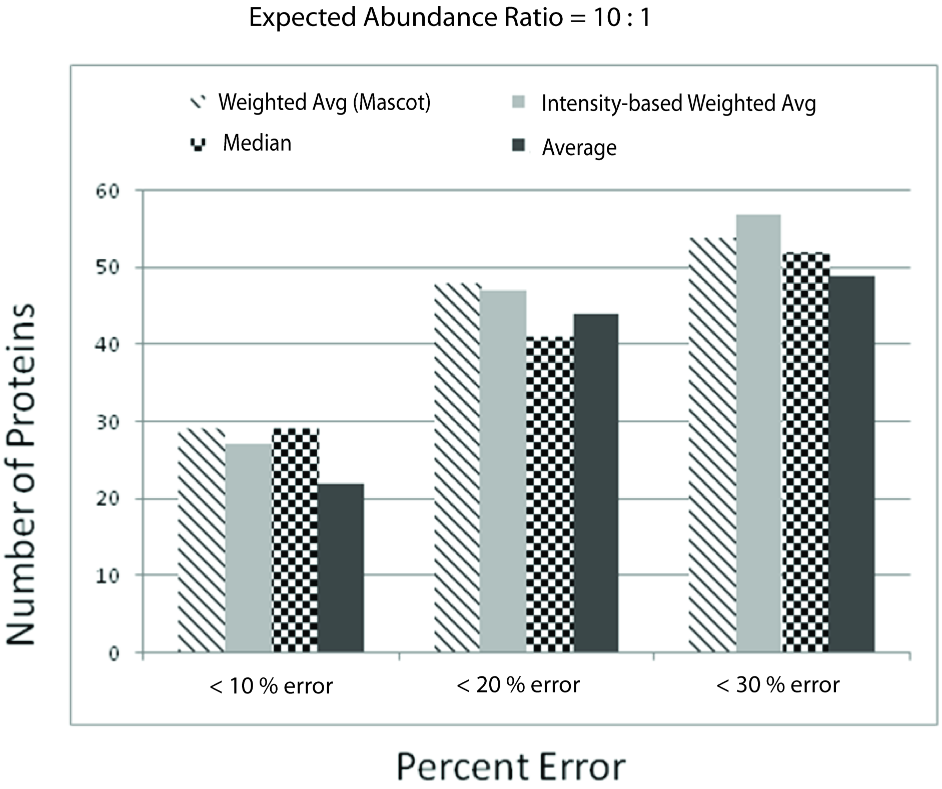


Figure 2.15: **Comparison of our quantification technique to quantification techniques offered by Mascot.** *Comparison of* $\frac{I_{114}}{I_{117}}$ *abundance ratios determined by each technique (10:1 expected ratio). Our technique is denoted as* ***"Intensitybased Weighted Avg"****. The other three techniques shown are those offered by Mascot.*

two instruments. Because we used 4plex iTRAQ reagents for labeling stem cell samples, three separate abundance ratios relative to the I_{114} signal ($\frac{I_{114}}{I_{115}}$, $\frac{I_{114}}{I_{116}}$ and $\frac{I_{114}}{I_{117}}$) were determined for each identified protein and assigned a P value. Focusing on proteins identified by three or more peptides, the LTQ-Orbitrap plus LTQ-iQuant quantified 910 abundance ratios ($P < 0.05$) while the MALDI 4800 TOF/TOF plus Protein Pilot quantified 784 abundance ratios ($P < 0.05$).

Next we assessed how well LTQ-iQuant and Protein Pilot agreed with each other when quantifying the same protein (Figures 2.17(a), 2.17(b) and 2.18). We plotted a correlation graph for the common 142 protein abundance ratios that were determined

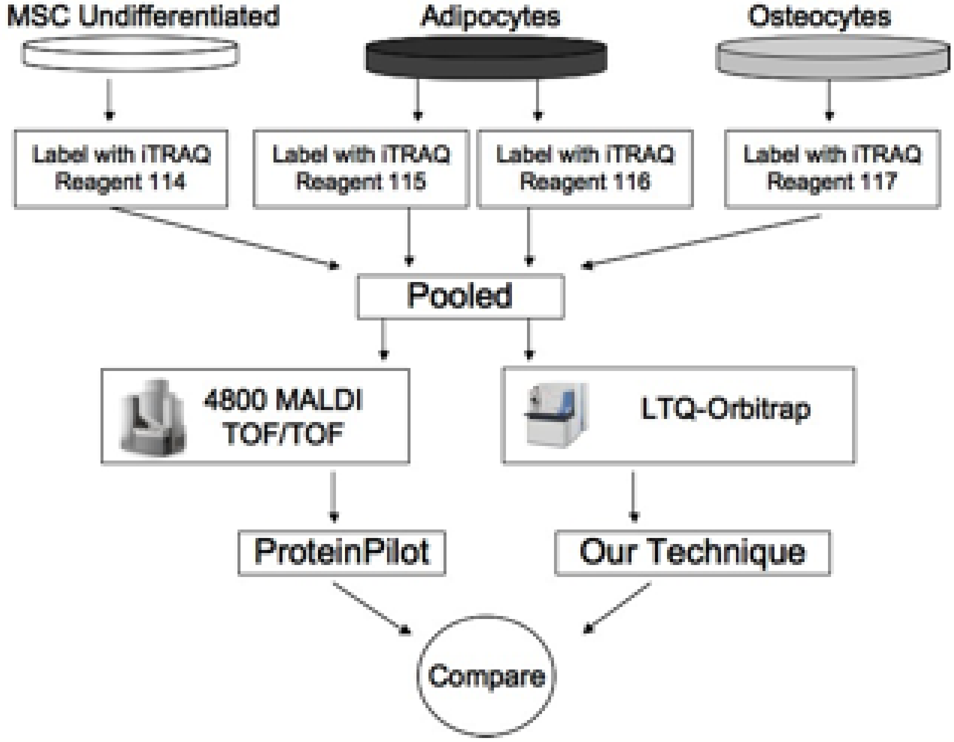
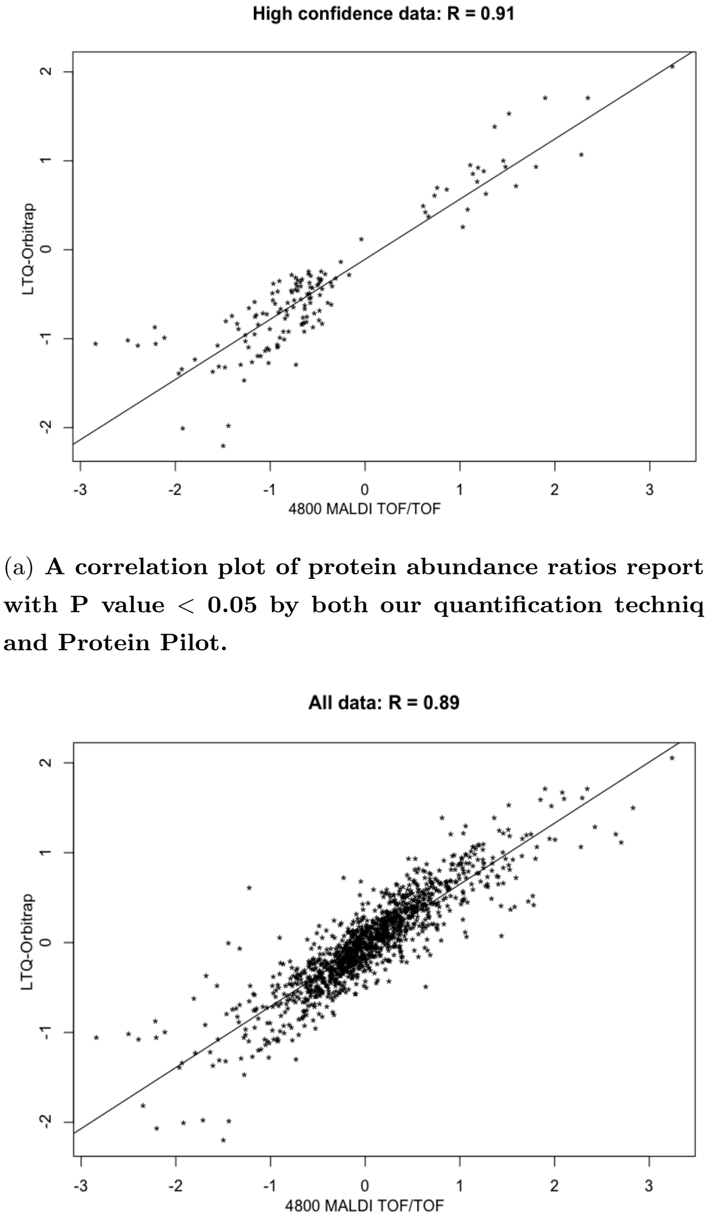

MSC Undifferentiated
Adipocytes
Osteocytes
Label with iTRAQ Reagent 114
Label with iTRAQ Reagent 115
Label with iTRAQ Reagent 116
Label with iTRAQ Reagent 117
Pooled
4800 MALDI TOF/TOF
LTQ-Orbitrap
ProteinPilot
Our Technique
Compare

Figure 2.16: *Experimental set-up comparing LTQ-Orbitrap plus our technique to 4800 MALDI TOF/TOF plus Protein Pilot for protein quantification*

with high confidence (P value < 0.05) by both Protein Pilot and LTQ-iQuant. The measured data between the two instruments and software programs showed positive correlation, with an R value of 0.91 (Figure 2.17(a)). We also plotted a correlation graph of ratios for all protein abundance ratios common to both data sets irrespective of assigned P value (Figure 2.17(b)). Even for this lower confidence data, abundance ratios obtained from the LTQ-Orbitrap and LTQ-iQuant was still largely in agreement with those same ratios from the MALDI 4800 TOF/TOF (R = 0.89).

Although the results in Figure 2.17(a) showed reasonably good positive correlation for proteins quantified by both instruments, the R value of 0.91 indicated that there was some level of disagreement between the two datasets. To better characterize the nature of this disagreement, we compared the magnitude and direction of measured relative



(a) **A correlation plot of protein abundance ratios reported with P value < 0.05 by both our quantification technique and Protein Pilot.**

(b) **A correlation plot for all protein ratios identified in common between the LTQ-Orbitrap and the 4800 MALDI TOF/TOF, with no P value threshold**

Figure 2.17: *Correlation analysis on proteins identified and quantified by both the LTQ-Orbitrap and 4800 MALDI TOF/TOF.*

protein abundance ratio changes between the two datasets. For abundance ratios determined with high confidence (P value < 0.05) between both instruments (Figure 2.17(a)), we plotted these in order from low to high values as determined by Protein Pilot, along with their corresponding value as determined by LTQiQuant (Figure 2.18). The results of this plot showed that the direction of each protein abundance ratio (either increased or decreased abundance) was in agreement between the LTQOrbitrap and MALDI 4800 TOF/TOF for all quantified proteins, while there was some variation in the magnitude of abundance change between the two instruments contributing to the disagreement observed in the correlation graph in Figure 2.17(a).

From this comparison, it is clear that for proteins quantified by both instruments, results obtained with the 4800 MALDI TOF/TOF and Protein Pilot were in generally good agreement to results with the LTQ-Orbitrap and LTQ-iQuant. Additionally, the LTQ-Orbitrap analysis plus our technique identified and quantified almost 2.5 times more proteins than comparable analysis on the 4800 MALDI TOF/TOF. This finding is especially significant since the 4800 instrument is the currently considered the best option for large-scale iTRAQ-based quantitative proteomic analyses [37]. Based on our collective findings from the yeast standard mixture and the comparison to the MALDI 4800, users can trust the combination of the LTQOrbitrap and LTQiQuant to provide accurate and sensitive results for quantitative proteomics using isobaric peptide tagging.

2.7 Software Implementation

To make this technique accessible to the proteomics community, we implemented it in a software pipeline automating the analysis of large-scale isobaric peptide tagging data on LTQ-type instruments. Figure 2.19 shows the the pipeline's various components and workflow. The software was written using the statistical package R, programming language Perl, bash shell scripts and the querying language SQL. It contains a parser, relative quantification code, MySQL database and a bash script that runs the pipeline with the option of including a visualization program. Collectively, they make up its three main components: protein identification (Figure 2.19(a)), weight matrix generation (Figure 2.19(b)) and protein quantification (Figure 2.19(c)) as described below. The rectangle shapes represent software packages and the hexagon shapes represent data.

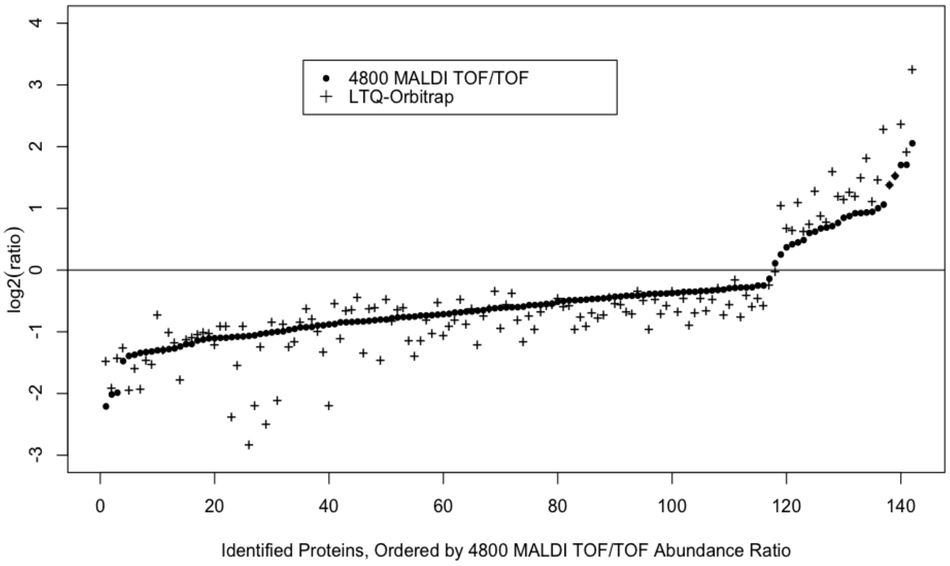


Figure 2.18: **Correlation analysis on proteins identified and quantified by both the LTQ-Orbitrap and 4800 MALDI TOF/TOF**. *Protein abundance ratios with P value < 0.05 quantified by both our technique and Protein Pilot. Protein abundance ratios are plotted in ascending order of magnitude as measured by Protein Pilot from the 4800 MALDI TOF/TOF data (black dots) along with corresponding abundunce ratio for the same protein as measured by pur technique from LTQ-Orbitrap data (crosses)*

First, as depicted in Figure 2.19(a), a sample is analyzed by mass spectrometry (LTQ-Orbitrap). Data representing spectra identified is extracted (.dta) and searched against a protein database. To search against the protein database, a reference set of peptides are generated based on the enzyme used in the experiment (trypsin in most cases). Theoretical spectra corresponding to these peptides are then generated and used as the reference. Spectra from the mass spectrometer are searched against the theoretical spectra to identify peptides in the sample (.out). These peptides are mapped to proteins and used to identify proteins in the sample. These identifications are then filtered to desired criteria (.xls). Reporter ions from isobaric tags extracted from the mass spectrometry output are paired with corresponding filtered identifications using a parser (PARSER). The resulting data containing identified proteins, their corresponding peptides together with reporter ion intensities are stored in a MySQL relational database

(DATABASE) for analysis.

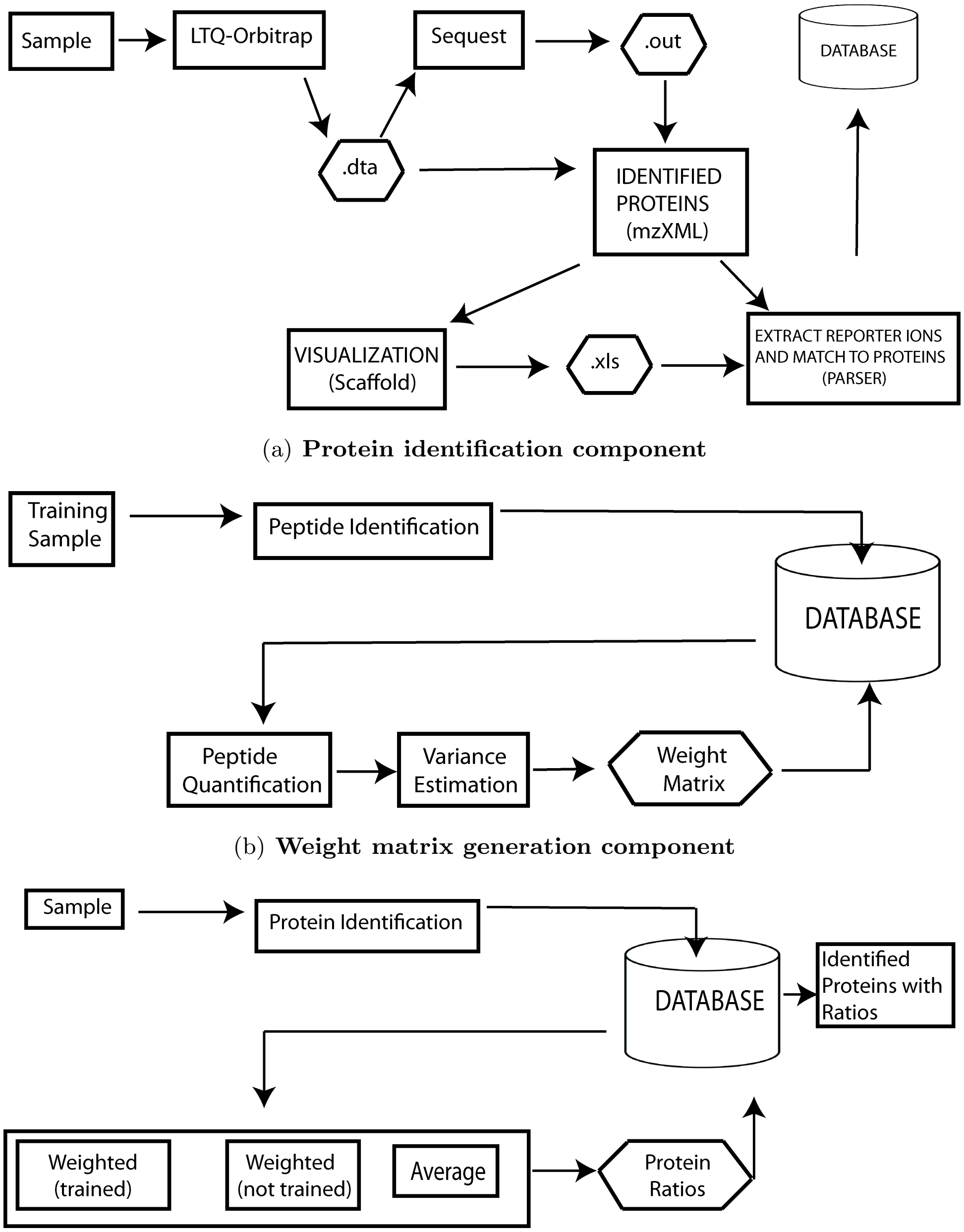
Next, a weight matrix, using default values based on our training data from section 2.5.2, or using custom values from user-generated training data, is stored in the database, as depicted in Figure 2.19(b). The training data goes through the identification process depicted in Figure 2.19(a). The resulting data, stored in a MySQL database, is analyzed using a software package implemented in R to generate the weight matrix in Figure 2.19(b). The rectangle shapes in Figure 2.19(b) represent software packages written in R for peptide quantification and generation of the weight matrix. This stored weight matrix is used to assign weights to experimental data whose protein abundance is unknown.

Finally, using aggregated, weighted peptide abundance ratios a weighted average abundance ratio for each protein is calculated using software packages written in R. For each computed protein ratio, a P value, which measures uncertainty and enables an assessment of the ratios statistical significance, is calculated. Computed ratios and their associated P values are stored in the MySQL database, as shown in Figure 2.19(c).

2.8 Discussion

Via a systematic evaluation LTQ-Orbitrap isobaric peptide tagging data analyzed using PQD operation, we have confirmed the dependence of peptide abundance ratio accuracy on reporter ion intensities. This finding led us to develop a quantification technique tailored to this data, which we implemented in an automated software pipeline. We proved that weighted averaging is superior to averaging without weighting, the technique used currently by commercially available software. We demonstrated the accuracy of our technique by comparison to the 4800 MALDI TOF/TOF using Protein Pilot.

Our peptide weighting technique differs from the technique successfully used by the Kuster laboratory to account for inaccuracies by weak reporter ion signals from LTQ data obtained using PQD operation. They used a linear regression analysis to account for outlier peptide ratios potentially produced from low intensity reporter ion signals [22, 32]. Although the linear regression method can be effective, we decided against using it in our technique for two reasons. For one, we were concerned about the validity of the assumptions made in such a regression, for example, a linear relationship



(a) **Protein identification component**

(b) **Weight matrix generation component**

(c) **Protein quantification component**

Figure 2.19: **Relative Quantification Software.** *A diagram showing the different components combined to form the software pipeline.*

between reporter ion intensities and constant variance. Others have demonstrated that the assumption of constant variance may not be valid and may introduce errors into quantification of isobaric peptide tagging data [90]. Our use of a weight matrix does not make any such assumptions on the data. Second, use of the linear regression and its underlying assumptions did not lend itself as well to our desire to make our technique amenable to user inputted training data for customizing the peptide weight matrix.

The comparison of our techniques results to 4800 MALDI TOF/TOF results illuminated two points. First, for proteins quantified by both instruments, results obtained with the 4800 MALDI TOF/TOF and Protein Pilot were comparable to results with the LTQ-Orbitrap and our technique, indicating our techniques accuracy. Second, the comparison showed that LTQ-Orbitrap analysis plus our technique identifies and quantifies significantly more proteins than comparable analysis on the 4800 MALDI TOF/TOF. This finding is especially significant since the 4800 instrument is the currently considered the best option for large-scale iTRAQ-based quantitative proteomic analyses [37]. Based on these findings, users can trust the combination of isobaric peptide tagging plus LTQ-Orbitrap analysis plus our technique to provide accurate and sensitive results for quantitative proteomics.

Besides automating our technique to make it amenable to largescale quantitative proteomic studies, the LTQ-iQuant software pipeline has several characteristics that make it attractive. Instead of distributing just binary code files, we have chosen to make the source code, and documentation, freely available. The source code can be accessed at
https://netfiles.umn.edu/users/onson001/www/LTQiQuant.html.
The pipeline has been developed using Java, making it platformindependent. It was been tested on Windows XP, Ubuntu 8.04 the Hardy Heron, and Mac OS X 10.4.11 platforms. LTQiQuant is mzXML compatible, which gives users the flexibility to employ the pipeline with different database search programs. It can be used for both 4plex and 8plex iTRAQ reagent labeling methods, accepts isotope purity correction factors, and is amenable to experimental data where not all four or eight iTRAQ labels are used (e.g., a binary sample comparison where only the 114 and 117 labels are used, etc.).

A key advantage of the software pipeline is the capability of users to input their own training data, enabling generation of a peptide weighting matrix customized to individual instrument performance, different isobaric tagging methods (e.g. TMT) and possibly even emerging instrumental operation methods (e.g. Orbitrap HCD [39, 38] or ETD [40, 41]. In our experience, the absolute intensities of reporter ions can vary between different LTQ instruments, and even on the same instrument under different tuning parameters. We would also expect that different isobaric tagging methods (e.g. TMT versus iTRAQ) would produce different absolute reporter ion intensities. Given these variations, the relationship of the product of reporter ion intensities on accuracy may differ from the relationship derived from the analysis of our standard iTRAQ labeled yeast mixture, which was used to generate our default peptide weights 2.8. Although our default weighting values may still produce acceptable results, the ability to create customized weights using our software pipeline and user-generated training data should help ensure the most accurate quantitative measurements for users. Also, we developed the pipeline using sound software engineering practices so that users could modify the pipelin.

In summary, our new technique and software pipeline make the powerful combination of isobaric peptide tagging and the LTQ usable in a wide-variety of quantitative proteomics studies.

Chapter 3

Relational Database Operators

This chapter presents relational database operators for analyzing high-throughput proteomics data using biological graph data and biological pathway data. Section3.1 presents operators that use biological graph data represented by sets of binary relationships. Section3.2 presents a datamodel for biological pathways and new operators that use this datamodel to analyze high-throughput proteomics data. Later in Chapter 4, these operators are evaluated by analyzing high-throughput proteomics in an oral cancer experiment.

3.1 Operators for Binary Relation Biological Data

Protein-protein interactions and the Gene Ontology database are examples of binary relation biological data. A protein-protein interaction occurs when two or more proteins bind together. The Gene Ontology database consists of ontolgoy terms used to categorize genes and gene products. These terms are organized in a hierarchical structure forming a Directed Acyclic Graph structure. Figure 3.1 shows a relational schema for storing binary relation data in relational database. Proteins in protein-protein interaction data are stored in the *Node* relation and interactions between two proteins are stored in the *Edge* relation. Ontology terms in the Gene Ontology database are stored in the *Node* relation and relationships between terms are stored in the *Edge* relation.

Before presenting these operators, we present sample data that will be used to describe the operators. A typical high-throughput experiment identifies several thousand

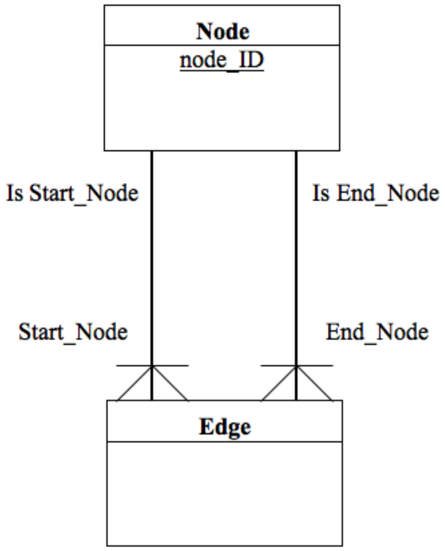


Figure 3.1: Relational schema for binary relations

proteins. The purpose of this sample data is to demonstrate functionality and not the utility of these operators.

3.1.1 Illustration Data

Table 3.1 is an example of high-throughput proteomics data analyzed to identify candidate biomarkers. It contains protein abundances for two disease conditions, premalignant and malignant. In these experiments, one of the initial tasks is to find differentially abundant proteins. Often, the null hypothesis is *"the abundance level between the two groups is the same"* i.e., the ratio (pre-malignant/malignant) = 1. For illustration purposes we will assume the filtering condition (ratio > 2.0 or ratio < 0.5) defines differentially abundant proteins. Using this condition the proteins **egf**, *ras*, *stat3*, and *amy1a* are in the list of differentially abundant proteins (Table 3.2).

In high-throughput experiments, it is useful for researchers to be able to group genes or proteins into broad biological categories that give a higher-level view of their function [71]. The Gene Ontology (GO) database provides a nomenclature for categorizing genes and gene products based on their function and cellular localization. GO consists of genes and gene products plus certain concepts, called terms, associated with them,

Protein	Pre malignant	Malignant	Ratio (pre/malignant)
egf	200	200	**0.21**
egfr	150	100	1.5
ras	500	100	**5.0**
stat1	180	100	1.8
stat3	100	400	**0.25**
amy1a	500	100	**5.0**
cRAF	90	100	0.9
jak1	80	100	0.8

Table 3.1: Sample data for high-throughput experiment data measuring protein abundance levels in different conditions (disease states). We purposefully did not include *Cell Growth* in this sample data to better reflect experimental conditions where some proteins are not identified by the high-throughput technology being used.

Differentially abundant protein	
Protein	Ratio
egf	0.21
ras	5.0
amy1a	5.0
stat3	0.25

Table 3.2: Differentially abundant proteins between pre-malignant and malignant states

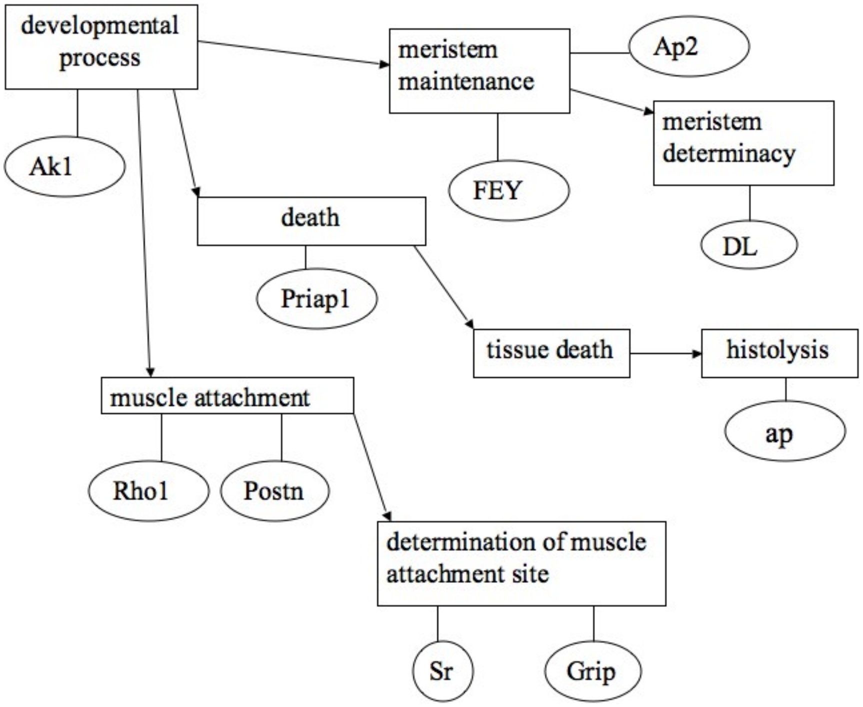


Figure 3.2: A portion of the gene ontology database.

and, in addition, other data that is not relevant here. GO organizes terms and parent-child relationships between terms into three separate ontologies for biological processes, molecular functions and cellular components. Each ontology forms a directed acyclic graph, DAG, with each node being a term and each parent-child relationship being a directed arc between distinct nodes. In GO each child term is a more specific process, function or component than each of its parent terms. An *association* connects a gene or gene product with the most specific possible term, and implicitly applies to the termś ancestors. Collectively, the genes and gene products associated with a term are called its *annotation*. Figure 3.2 shows a small portion of GO, with terms appearing inside rectangles, genes or gene product associated with a term appearing inside ellipses attached to its rectangle, and parent term - child term relationships appearing as arrows.

Another example of analysis of high-throughput data is, given a list of differentially abundant proteins, find those interacting with each biological pathway known to have a role in development of the disease being studied. To perform this analysis using

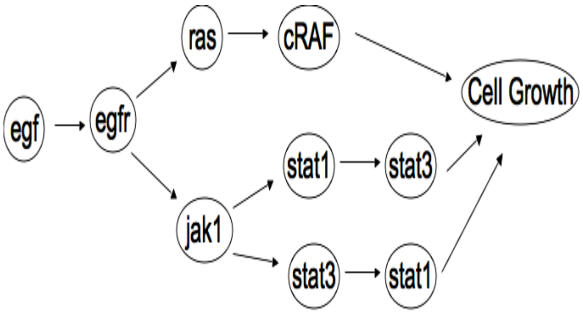


Figure 3.3: Sample pathway (part of EGF pathway)

relational database operators these proteins have to be stored in the database. Because relational databases are content-neutral, we will store these proteins in Table 3.2 in a relation named *node of interest* and refer to them as *nodes*.

A biological pathway is defined by the set of molecules and reactions in the pathway. Because each molecule in a pathway interacts with at least one other molecule i.e., biological pathways do not have orphan molecules, in abstract terms a biological pathway can be defined by a set of directed edges representing reactions in the pathway. We use directed edges to represent and store biological pathways in a relational database. Figure 3.3 shows a portion of the EGF signaling pathway. Table 3.3 shows a directed edge relation representing edges that define the pathway. The underline below the attribute names denotes identifying columns.

Uncontrolled cell growth being one of the defining characteristic of cancerous cells, for illustration purposes we will use the pathway *cell growth* as an example of a pathway known to have a role in development of oral cancer. Oral cancer will be the disease being studied. Again, because relational databases are content-neutral, we will store the pathway *cell growth* in a relation named *pathway_of_interest*. Table 3.4 represents this *pathway_of_interest*. Note, while the other nodes in Figure 3.3 are proteins, *cell growth* is a pathway containing more than one protein. For simplicity a single node is used to represent all the proteins in *cell growth* pathway.

Edge	
StartNode	EndNode
egf	egfr
egfr	ras
egfr	jak1
ras	cRAF
cRAF	Cell Growth
jak1	stat1
jak1	stat3
stat1	stat3
stat3	stat1
stat1	Cell Growth
stat3	Cell Growth

Table 3.3: Edge relation for Figure 3.3

Protein in pathway associated with cancer development
Protein
Cell Growth

Table 3.4: Protein in pathway associated with cancer development

3.1.2 Operator Overview

This section presents four different operators used to analyze high-throughput genomics and proteomics data. With the exception of the fourth operator, different variations of the same operator are presented for the different input parameters used with each operator.

The first operator, presented in Table 3.5, retrieves pairs of proteins connected by protein-protein interactions. It can be used to identify candidate biomarkers connected to pathways with known association to oral cancer. The second operator, presented in

Table 3.10, is used to determine each edge in a path connecting two proteins. This operator makes it possible for users to examine interactions between candidate biomarkers and biological pathways, a useful functionality that aids understanding of disease mechanism [101]. The third operator, presented in Table 3.12, is a macro that identifies the most promising candidate biomarkers based on their interactions with pathways associated with the disease being studied. The fourth operator *BuildGoSlim* uses the Gene Ontology database to categorize genes and protein into broad biological categories.

3.1.3 Transitive Edge Operators

The operator presented in Table 3.5 generates relations used to determine whether two proteins are connected (transitive edges). To avoid infinite loops, it has a cycle detection mechanism that does not permit cycles in transitive edges. By definition, input edges are transitive edges with distance one thus ensuring they are part of the result relation.

Compute ShortestDist Transitive Edge

The precedence charts " PRECEDENCE CHART A1 " and " PRECEDENCE CHART A2 " show one iteration of *Compute ShortestDist Transitive Edge*. The same basic steps are outlined in the pseudocode in Algorithm 3.1. For simplicity, abbreviated short names have been given for result relations in the precedence charts shown. However it is important to use proper result relation names that specify a relations **base, column modifier** and **row modifier**. For more details on importance of using proper result relation names see [102]. Table 3.6 list these abbreviated short names together with their corresponding proper result relation names.

Operator Summary

We use the predecence charts to describe these basic steps with corresponding operations in Algorithm 3.1 given in brackets. *Compute ShortestDist Transitive Edge* takes as input an *edge* relation defining a pathway database e.g., Table 3.3 and finds each shortest distance transitive edge betwen two proteins.

Circumstance				Result Relation Name			Result Relation Structure		
Operator	**Type**	**Relation inputs**	**Non-relation inputs**	**Base**	**Row modifier**	**Column modifier**	**Identifier**	**Width**	**Height**
Compute MinDist Transitive Edge	unary	edge	result_rel_name	no change	transitive	identifying columns + 1	edge	identifier set size + 1	≥ h
	“ *find each protein interacting with X other protein*” where X is an integer								
Compute Distance Restricted Transitive Edge	unary	edge	- result_rel_name - distance	no change	- transitive - distance	identifying columns + 1	edge	identifier set size + 1	≥ h
	“ *find each protein interacting with X other proteins using Y or fewer interaction*” where X and Y are integers								
Compute Distance & StartNode Restricted Transitive Edge	binary	- edge - StartNode	- result_rel_name - distance	edge	- transitive - StartNode - distance	identifying columns +1	edge	identifier set size + 1	≥ h
	“ *find each protein interacting with P using X or fewer interaction*” where P is a protein and X is an integers								

Table 3.5: **Compute Transitive Edge Operators:** Operators used to identify protein-protein interactions

Pre-processing

There is no standard datamodel for storing pathways in relational databases. Before invoking the operator, the pathway data has to be pre-processed to generate an *edge* relation similar to the one shown in Table 3.3. For some data sets e.g., HPRD [103] the task is straight-forward since its data is stored as protein-protein interaction pairs. For others such as Reactome [104] which uses a frame-based model, the task is not as straigh-forward.

Given this variability in choice of datamodels, we leave it to the user to transform pathway data to the appropriate format shown on Table 3.3. For this thesis, PL/SQL procedures were written that transform Reactome to the approriate format. These procedures together with details of how they transform Reactome to the appropriate format are presented in Appendix A.

Initialization Step

The first PROJECT in " PRECEDENCE CHART A1 " is used add a distance attribute to each edge in the input relation and initialize the relations R_N_delta, R_N_Minus_1_delta and TN ($R^N_\Delta, R^{N-1}_\Delta$ and TN: Lines 1 to 2, Algorithm 3.1). The relation R_N_delta (R^N_Δ) contains transitive edges newly generated at iteration N. R_N_Minus_1_delta (R^{N-1}_Δ) contains transitive edges newly generated at iteration N-1 TN (TN) will store each transitive edge generated and will be the final result relation.

The following GROUP, MATCH JOIN AND REDUCE initialize relations T1_max, A_B_pair and R_N_Minus_1_maxDelta ($T1, A\_B\_pair\ and\ R^{N-1}_{max\Delta}$: Lines 3 to 5, Algorithm 3.1). T1_max ($T1_{max}$) will contain *start node* and distance for longest distance transtive edge with *start node* as the input node. A_B_pair (A_B_pair) is a temporary relation used to generate R_N_Minus_1_maxDelta ($R^{N-1}_{max\Delta}$) which contains each longest distance transitive edge generated in iteration N-1.

Determining duplicate generating input rows

To avoid unnecessary joins *Compute ShortestDist Transitive Edge* determines apriori edges in input relations to joins that would generate transitive edges with distances greater than the shortest distance transitive edge for each two pair of nodes. Recall, because we are only interested in connectivity information, it is not necessary to compute each transitive edge between two nodes. Except the two symmetric either Match Joins (Lines 14 and 17, Algorithm 3.1), the operators in “ PRECEDENCE CHART A2 ”) are used to ensure input relations to these symmetric either Match Joins do not contains edges that will produce transitive edges with distance greater than the shortest distance transitive edge between two proteins. For details on how these operators eliminate tuples from the input relations see Appendix C.

Generating new transitive edges

New transitive edges are generated by the two Match Joins (Lines 14 and 18, Algorithm 3.1) at the bottom of page 1 of “ PRECEDENCE CHART A2 ”. The REDUCE operators (Lines 16 and 20, Algorithm 3.1) ensure duplicates are removed. Results of these two joins are temporarily stored in relations D1 and D2 (**D1** *and* **D2**) and then combined to form relation D1unionD2 (Line 22, Algorithm 3.1).

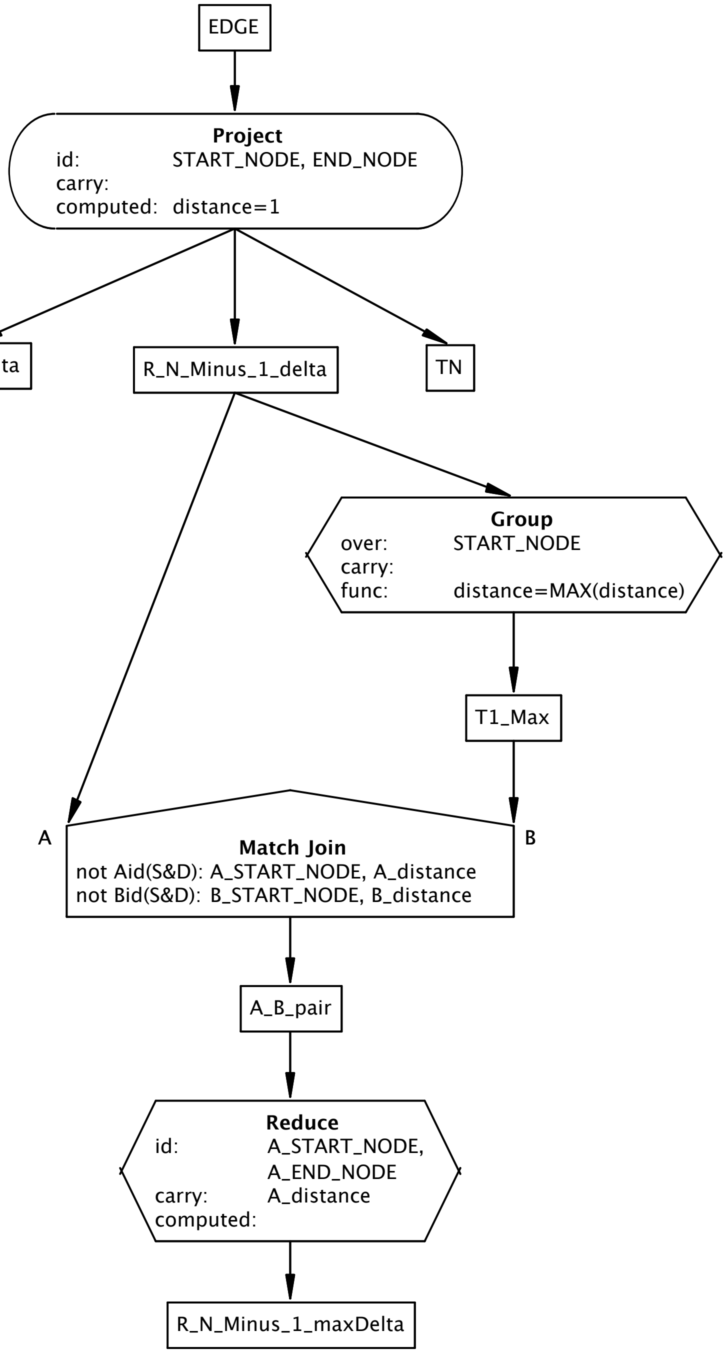
Updating output and intermediate relations

The other operators on page 1 of “ PRECEDENCE CHART A2 ” ensure no new transitive edges are generated with distances greater than the shortest distance transtitive edge. It is possible, however, to get transitive edges with distance equal to the shortest distance. The MINUS operator on page 2 of “ PRECEDENCE CHART A2 ” removes these transitive edges from the newly generated transitive edges. TN which contains the transitive edges generated so far is updated together with R_N_delta (R^N_Δ: Lines 21-22, Algorithm 3.1). If R_N_delta (R^N_Δ) contains zero rows, the operator terminates. Otherwise it continues until no new transitive edges are generated.

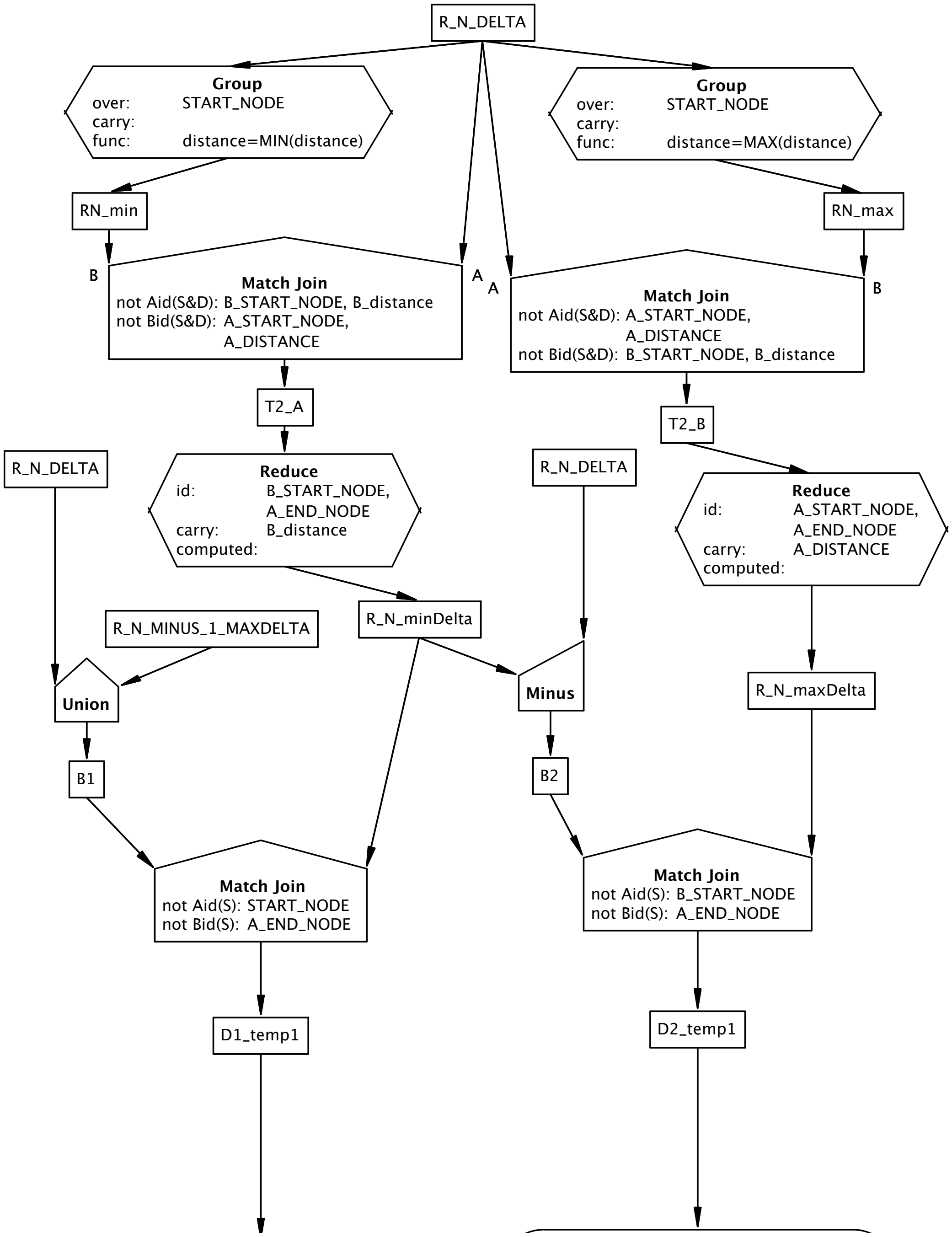
Result relation names for precedence charts A1 and A2	
Short Name	**Descriptive Long Name**
R_N_delta	new transitive edge with distance data (iteration N)
R_N_Minus_1_delta	new transitive edge with distance data (iteration N - 1)
TN	transtive edge (iteration N)
T1_Max	start_node with max distance data
A_B_pair	R_N_Minus_1_delta - start_node with max distance data pair
R_N_Minus_1_maxDelta	new trans edge with max dist data
RN_Min	start_node with min distance data
RN_Max	start_node with max distance data
T2_A / T2_B	new transitive edge with distance - start_node with min / max distance pair
R_N_minDelta	new transitive edge with min distance
R_N_maxDelta	new transitive edge with max distance
B1	new transitive edge with max distance (iteration N -1) and new transitive edge with distance (iteration N)
B2	new transitive edge with distance minus new transitive edge with min distance
D1_temp1	B1 - R_N_min Delta pair
D2_temp1	B1 - R_N_max Delta pair
D1_temp2	B1 - R_N_min Delta pair with updated distance
D2_temp2	B2 - R_N_max Delta pair with updated distance
D1 D2	newly generated transitive edge with distance data
D1unionD2	newly generated transitive edge with distance data
new_R_N_delta	new transitive edge with distance data (iteration N + 1)
new_TN	transtive edge (iteration N + 1)

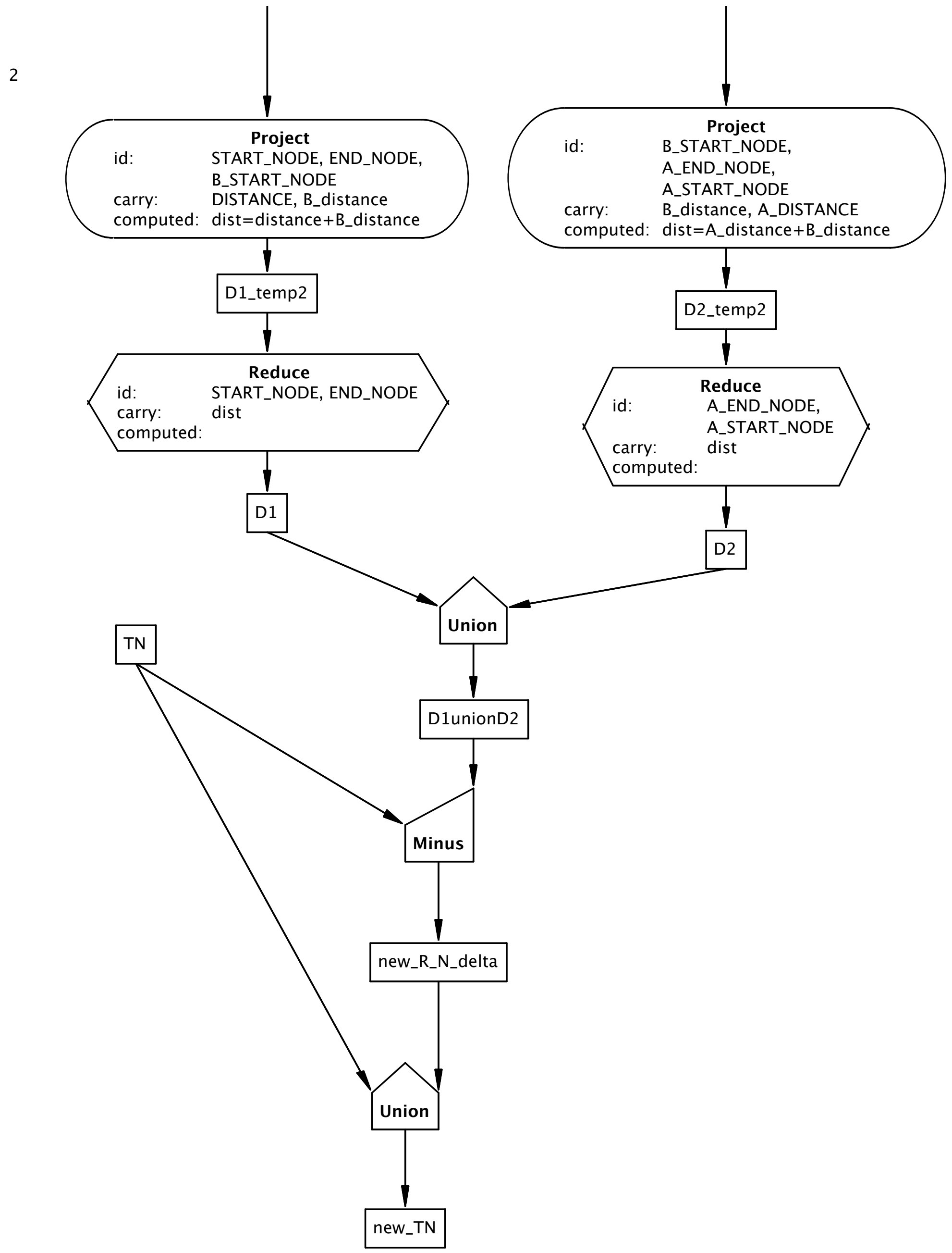
Table 3.6: Abbreviated short names together with their corresponding proper result relation names for precedence charts A1 and A2.

PRECEDENCE CHART A1



PRECEDENCE CHART A2




2
Project
id: START_NODE, END_NODE, B_START_NODE
carry: DISTANCE, B_distance
computed: dist=distance+B_distance
Project
id: B_START_NODE, A_END_NODE, A_START_NODE
carry: B_distance, A_DISTANCE
computed: dist=A_distance+B_distance
D1_temp2
D2_temp2
Reduce
id: START_NODE, END_NODE
carry: dist
computed:
Reduce
id: A_END_NODE, A_START_NODE
carry: dist
computed:
D1
D2
Union
TN
D1unionD2
Minus
new_R_N_delta
Union
new_TN

Algorithm 3.1 Compute ShortestDist Transitive Edge (Edge:operand relation; TN:result relation)

```
$TN := PROJECT(Edge, [start, end, dist = 1])$
$R_{\Delta}^{N-1} := TN, R_{\Delta}^{N} := TN$ {$R_{\Delta}^{N}$ will contain new tuples}
$T1 := GROUP(R_{\Delta}^{N-1}, [start, dist = MAX(distance)])$
$A\_B\_pair := JOIN(R_{\Delta}^{N-1}, T1, [R_{\Delta}^{N-1}.start = T1.start \wedge R_{\Delta}^{N-1}.dist = T1.dist])$
$R_{max\Delta}^{N-1} := REDUCE(A\_B\_pair, [R_{\Delta}^{N-1}.start, R_{\Delta}^{N-1}.end, R_{\Delta}^{N-1}.dist])$
while $R_{\Delta}^{N} \neq 0$ do
    $T1 := GROUP(R_{\Delta}^{N}, [start, dist = MIN(dist)])$
    $T2 := JOIN(R_{\Delta}^{N}, T1, [R_{\Delta}^{N}.start = T1.start \wedge R_{\Delta}^{N}.dist = T1.dist])$
    $R_{min\Delta}^{N} := REDUCE(T2, [R_{\Delta}^{N}.start, R_{\Delta}^{N}.end, R_{\Delta}^{N}.dist])$
    $T1 := GROUP(R_{\Delta}^{N}, [start, dist = MAX(dist)])$
    $T2 := JOIN(R_{\Delta}^{N}, T1, [R_{\Delta}^{N}.start = T1.start \wedge R_{\Delta}^{N}.dist = T1.dist])$
    $R_{max\Delta}^{N} := REDUCE(T2, [R_{\Delta}^{N}.start, R_{\Delta}^{N}.end, R_{\Delta}^{N}.dist])$
    $A1 := R_{min\Delta}^{N}, B1 := UNION(R_{\Delta}^{N}, R_{max\Delta}^{N-1})$
    $D1_{temp1} := JOIN(A1, B1, A1.end = B1.start)$
    $D1_{temp2} := PROJECT(D1_{temp}, A1.start, B1.end, dist = A1.dist + B1.dist)$
    $D1 := REDUCE(D1_{temp2}, start, end, dist)$
    $A2 := R_{max\Delta}^{N}, B2 := MINUS(R_{\Delta}^{N}, R_{min\Delta}^{N})$
    $D2_{temp1} := JOIN(A2, B2, A2.end = B2.start)$
    $D2_{temp2} := PROJECT(D2_{temp1}, A2.start, B2.end, dist = A2.dist + B2.dist)$
    $D2 := REDUCE(D2_{temp2}, start, end, dist)$
    $R_{\Delta}^{N-1} := R_{\Delta}^{N}, R_{max\Delta}^{N-1} := R_{max\Delta}^{N}$
    $R_{\Delta}^{N} := MINUS(UNION(D1, D2), TN), TN := UNION(T, R_{\Delta}^{N})$
end while
```

Compute ShortestDist Transitive Edge is useful when identifying candidate biomarkers connected to a pathway with known association to oral cancer. For example, it can be used to find which of the candidate biomarkers listed in Table 3.2 is connected to *Cell Growth*. Uncontrolled cell growth being one of the defining characteristics of cancerous cells, proteins interacting with *cell growth* could be potential reliable biomarkers. Note, *cell growth* is an abstraction for the different proteins in the *cell growth* pathway.

Minimum distance transitive edge for part of EGF pathway (Fig 3.3)		
Start Node	End Node	Distance
egf	egfr	1
egf	ras	2
egf	jak1	2
egf	cRAF	3
egf	stat1	3
egf	stat3	3
egf	Cell Growth	4
egfr	ras	1
egfr	jak1	1
egfr	cRAF	2
egfr	stat1	2
egfr	stat3	2
egfr	Cell Growth	3
ras	cRAF	1
ras	Cell Growth	2
cRAF	Cell Growth	1
jak1	stat1	1
jak1	stat3	1
jak1	Cell Growth	3
stat1	stat3	1
stat1	Cell Growth	1
stat3	stat1	1
stat3	Cell Growth	1

Table 3.7: Result relation of using relations *edge* (Table 3.3) input relation to *Compute MinDist Transitive Edge*

Table 3.7 shows the result relation of *Compute ShortestDist Transitive Edge* when using Table 3.3 as the input (*edge*) relation. Looking at the result relation (Table 3.7), *egf*, *ras* and *stat3* interact with *cell growth* and can be selected as promising candidate biomarkers.

Compute Distance Restricted Transitive Edge

The operator *Compute ShortestDist Transitive Edge* generated each minimum distance transitive edge betwen two proteins. Its result relation can be used to, for example, find each protein interacting with *cell growth* an important pathway when studying cancer. However, each protein is part of the same complex biological system, and at a global level all proteins interact with each other. To ensure results of any analysis are not too broad to be practically useful, it is necessary to restrict number of interactions permitted between candidate biomarkers and pathways associated with cancer development [33]. Using the result relation produced by Table 3.7, each protein in the sample pathway database interacts with *cell growth*. Consequently, using proteins that interact with *cell growth* as criteria for identifying promising biomarkers in the sample pathway database is meaningless since each protein in the sample database interacts with *cell growth*.

Compute Distance Restricted Transitive Edge, takes a *maximum distance* parameter and restricts transitive edges to those with distance less than the specified *maximum distance*. It takes as input an edge relation defining a pathway database e.g., Table 3.3 and a *maximum distance* as input parameters. It generates each minimum distance transitive edge in the pathway database terminating after transitive edges with distance equal to the *maximum distance* are generated. It uses the same set of operators in one iteration as *Compute ShortestDist Transitive Edge* (presented in the precedence charts labeled "*PRECEDENCE CHART A1*" and "*PRECEDENCE CHART A2*" or Algorithm 3.1) but terminates when transitive edges with distance equal to *maximum distance* are generated. *Compute Distance Restricted Transitive Edge* terminates when no new transitive edges are generated.

Table 3.8 shows the result relation of *Compute Distance Restricted Transitive Edge* when using Table 3.3 as input (*edge*) relation and 2 as *maximum distance* parameter. It contains the attributes <u>*start node*</u>*,* <u>*end node*</u> *and distance* with <u>start node</u> and <u>end node</u> as the identifying columns. As seen in Table 3.8, the result relation does not contain transitive edges with distance greater than 2. This result relation can be queried to determine proteins interacting with *Cell Growth* connected by a distance of at most 2 to identify promising candidate biomarkers. Unlike the result relation produced by *Compute ShortestDist Transitive Edge*, *egf* cannot be a promising candidate biomarker because in Table 3.8 it is not connected to *Cell Growth*.

Minimum distance transitive edge for part of EGF pathway (Fig 3.3) with maximum distance of 2		
Start Node	End Node	Distance
egf	egfr	1
egfr	ras	1
egfr	jak1	1
ras	cRAF	1
cRAF	Cell Growth	1
jak1	stat1	1
jak1	stat3	1
stat1	stat3	1
stat3	stat1	1
stat3	Cell Growth	1
stat1	Cell Growth	1
egf	ras	2
egf	jak1	2
egfr	cRAF	2
egfr	stat1	2
egfr	stat3	2
ras	Cell Growth	2

Table 3.8: Result relation of using relations *edge* (Table 3.3) and 2 as input parameters to *Compute Distance Restricted Transitive Edge*

Compute Distance & StartNode Restricted Transitive Edge

The purpose of the two operators presented, *Compute ShortestDist Transitive Edge* and *Compute Distance Restricted Transitive Edge*, is to generate result relations that can be used to identify promising candidate biomarkers associated with pathways known to have a role in cancer development e.g., *Cell Growth* for any protein in a pathway database. *Compute ShortestDist Transitive Edge* generates each shortest distance transitive edge between two proteins. *Compute Distance Restricted Transitive Edge* generates each minimum distance transitive edge with distance less than or equal to a user specified *maximum distance.* Looking at their result relations (Tables 3.7 and 3.8) most of the rows generated have **start nodes** not in the candidate biomarker list (Table 3.2). Most of the work done to generate these transitive edges is not necessary since these proteins are not in the candidate biomarker list. If the list of candidate biomarkers in known apriori, one can avoid unnecessary computations by restricting transitive edges to those with proteins in the candidate biomarker list as **start nodes**.

Operator Summary

Compute Distance & StartNode Restricted Transitive Edge, takes as input an *edge* relation defining a pathway database e.g., Table 3.3, a *start_node* relation containing candidate biomarkers e.g., Table 3.2 and a *maximum distance* parameter. It finds each minimum distance transitive edge with a node in the relation *start_node* as the start node and terminates after transitive edges with distance equal to the *maximum distance* are generated. The precedence chart labeled "*PRECEDENCE CHART B*" shows steps for one iteration of the operator. Algorithm 3.2) shows the same steps but in pseudocode.

Initialization Step

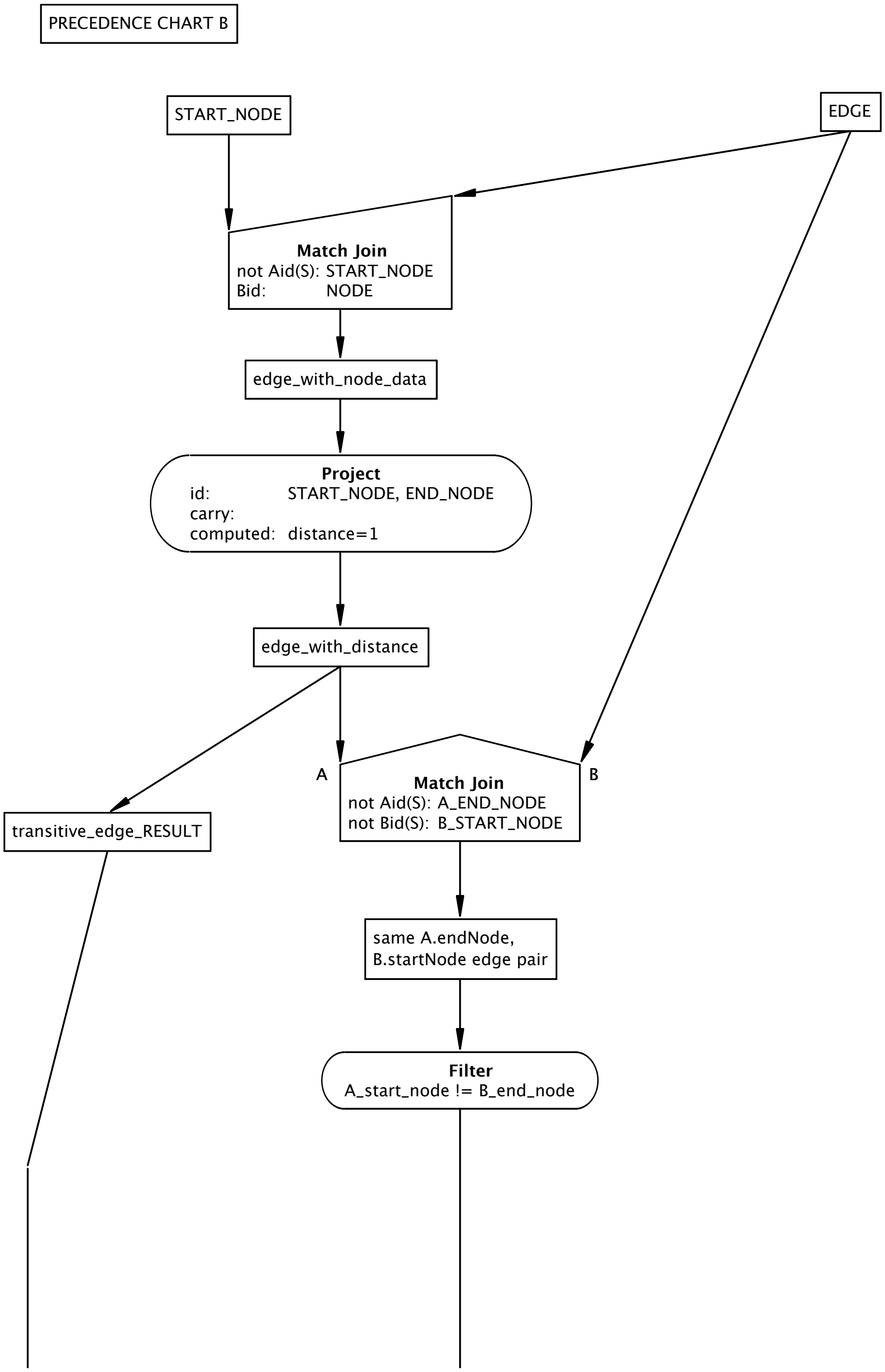
The first MATCH JOIN in " PRECEDENCE CHART B " is used filter the EDGE relation to restrict transitive edges to those with a *node* in START_NODE as the input node ["*edge_with_node_data*"] (R_{temp}: Line 1, Algorithm 3.2). If START_NODE contains the list of candidate biomarkers, this MATCH JOIN will ensure transitive edges generated are only for those with a protein in this list as the *start node.* The following PROJECT adds a distance attribute initialized to 1 to each edge in "*edge_with_node_data*" (R_A: Line 2, Algorithm 3.2). "*edge_with_node_data*" is then used to initialize the result relation ["*edge_with_node_data*"] (TN: Line 3, Algorithm 3.2).

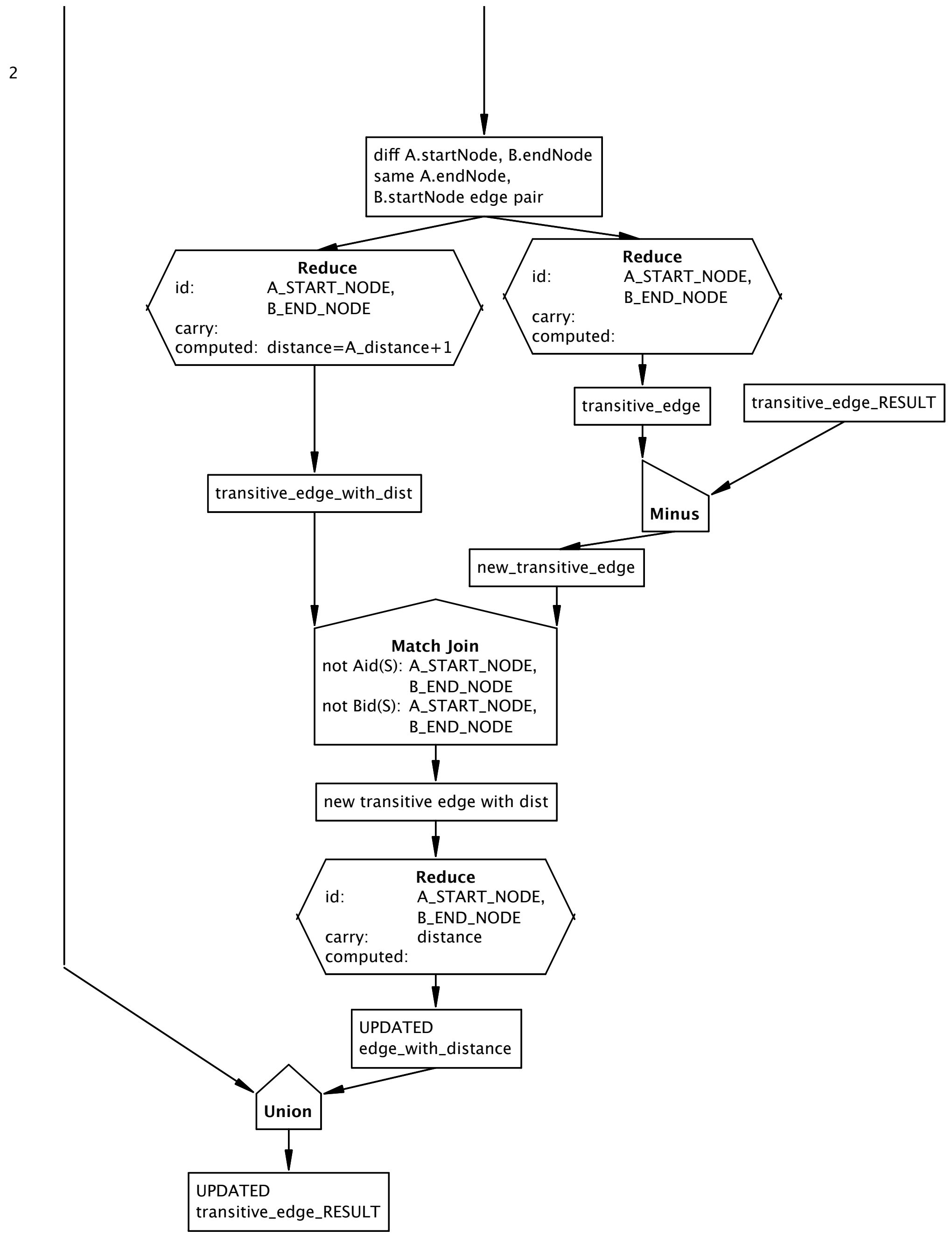
Generating new transitive edges

New transitive edges are generated by the symmetric either MATCH JOIN with EDGE and edge_with_distance as input relations ($T1$: Line 6, Algorithm 3.2). The FILTER is then used to ensure the START_NODE for relation A is not the same as the END_NODE for relation B thus preventing cycles ($T2$: Line 7, Algorithm 3.2). One REDUCE is used to increment the distance parameter by 1 (R_1: Line 8, Algorithm 3.2). The second REDUCE is used to produce a new relation containing the newly generated transitive edges but without the distance attribute ["*transitive_edge*"] (R_2: Line 9, Algorithm 3.2). The MINUS operator ensures the newly generated transitive edges are not connecting two nodes for which a transitive edge already exists in the result relation labeled transitive_edge_RESULT ["*new_transitive_edge*"] (*new_transitive_edge*: Line 10, Algorithm 3.2). The following MATCH JOIN adds the *distance* attribute to these new transitive edges ["*new transitive edge with distance*"] ($T3$: Line 13, Algorithm 3.2).

Updating output and intermediate relations

The REDUCE that takes in *new transitive edge with distance* as the input relation produces a relation *NEXT edge_with_distance* used to update the relation *edge_with_distance* to be used in the next iteration. The UNION adds these newly generated transitive edges to the result relation *UPDATED transitive_edge*. If *distance* parameter for each of the newly generated transitive edge is greater than or equal to *Max Distance* the operator terminates. Otherwise it iterates again with *NEXT edge_with_distance* as *edge_with_distance* and *UPDATED transitive_edge* as transitive_edge_RESULT.


PRECEDENCE CHART B
START_NODE
EDGE
Match Join
not Aid(S): START_NODE
Bid: NODE
edge_with_node_data
Project
id: START_NODE, END_NODE
carry:
computed: distance=1
edge_with_distance
A
B
Match Join
not Aid(S): A_END_NODE
not Bid(S): B_START_NODE
transitive_edge_RESULT
same A.endNode,
B.startNode edge pair
Filter
A_start_node != B_end_node


2
diff A.startNode, B.endNode
same A.endNode,
B.startNode edge pair
Reduce
id: A_START_NODE, B_END_NODE
carry:
computed: distance=A_distance+1
Reduce
id: A_START_NODE, B_END_NODE
carry:
computed:
transitive_edge
transitive_edge_RESULT
transitive_edge_with_dist
Minus
new_transitive_edge
Match Join
not Aid(S): A_START_NODE, B_END_NODE
not Bid(S): A_START_NODE, B_END_NODE
new transitive edge with dist
Reduce
id: A_START_NODE, B_END_NODE
carry: distance
computed:
UPDATED
edge_with_distance
Union
UPDATED
transitive_edge_RESULT

Algorithm 3.2 Compute Distance & StartNode Restricted Transitive Edge (Start_Node, Edge:operand relations; X:Max Distance; TN:result relation)

```
$R_{temp} := JOIN(Edge, Start\_Node, [Edge.Start\_Node = Start\_Node.node])$
$R_A := PROJECT(R_{temp}, [Start\_Node, End\_Node, dist = 1])$
$TN := R_A$
$STOP = 0$
while $STOP < X$ do
    $T1 := JOIN(R_A, Edge, [R_A.End\_Node = Edge.Start\_Node])$
    $T2 := FILTER(R_A, Edge, [R_A.End\_Node = Edge.Start\_Node])$
    $R_1 := REDUCE(T2, [R_A.Start\_Node, Edge.End\_Node, dist = R_A.dist + 1])$
    $R_2 := REDUCE(T2, [R_A.Start\_Node, Edge.End\_Node])$
    $new\_transitive\_edge := MINUS(R_2, Edge)$
    $A := R_1$
    $B := new\_transitive\_edge$
    $T3 := JOIN(A, B, [A.S\_Node = B.S\_Node \wedge A.E\_Node = B.E\_Node])$
    $NEW\_R_A := REDUCE(T3, [A.Start\_Node, A.End\_Node, A.distance])$
    $NEW\_TN := UNION(TN, NEW\_R_A)$
end while
```

Using Table 3.3, Table 3.2 and 2 as sample inputs for respectively *edge*, *start_node* and *maximum distance*, the task is to find each transitive edge that starts with either *ras, amy1a,* or *stat3* and has a maximum distance of 2. Note, because *amy1a* is not in our pathway (Figure 3.3), it is not considered by the operator despite being one of the differentially abundant protein. We deliberately chose to include a *amy1a* in the list of candidate biomarkers to demonstrates a limitations of using existing pathway databases, incomplete annotation. Most existing pathways databases are not fully annotated and sometimes lack some of the proteins in a users experiment.

Table 3.9 shows the result relation containing the attributes start node, end node *and distance* with start node and end node as the identifying columns. As seen in column *Start Node*, the operator only computes transitive edges for proteins in Table 3.2. Looking at Figure 3.3, ras is connected to *cRAF* by a direct edge and to *Cell Growth* via *cRAF. stat3* is connected to *stat1* and *Cell Growth* by direct edges. It is also connected

Minimum distance transitive edge for part of EGF pathway (Fig 3.3) with Start Node in Table 3.2 and maximum distance of 2		
Start Node	End Node	Distance
ras	cRAF	1
ras	cell growth	2
stat3	stat1	1
stat3	cell growth	1

Table 3.9: Result relation of using relations *edge* (Table 3.3), *StartNode* (Table 3.2) and 2 as input parameters to *Compute Distance & StartNode Restricted Transitive Edge*

to *Cell Growth* via *stat1*.

The operator also only retrieves the minimum distance transitive edges with *ras* and *stat3* as *start nodes*. Unlike *ras*, there are two *stat3* paths both connecting it to cell growth; the direct edge between *stat3* and *cell growth* and the path that goes through *stat1* (*stat3* → *stat1* → *cell growth*). The direct edge (*stat3, cell growth*) is the minimum distance transitive edge connecting the two nodes hence the absence of the transitive edge of distance 2 in the result relation.

The result relation generated by *Compute Distance & StartNode Restricted Transitive Edge* can now be queried to determine if *ras* and *stat3* are connected to *cell growth.* As shown in Table 3.7, *ras* and *stat3* both connected to *cell growth.*

Evaluation

Transitive closure database algorithms designed to answer connectivity queries have previously been studied [105]. These algorithms, however, have a limitation of exponential increase in space complexity with increased distance between proteins. *Compute ShortestDist Transitive Edge* presents an alternative to *transitive closure* for answering a class of complex biological queries that requires connectivity information but not the full transitive closure. A biological query such *" find each downstream protein interacting with glutamine synthetase using five or fewer interactions "* does not require *transitive closure*. Instead of generating each transitive edge between two nodes as is the case with *transitive closure*, it generates the shortest distance transitive edges which is sufficient to determine whether two proteins are connected.

In query optimization, selection of rows is sometimes done as early as possible, especially when join operations are involved. The rationale behind this rule is that a join operation can operate on a smaller table reducing work [106]. Using the same principle *Compute ShortestDist Transitive Edge* eliminates rows from input relations to join operations that would lead to transitive edges with distances greater than the shortest distance transitive edge. The smaller input relations leads to reduced join operations and better performance. Appendix C presents a detailed description of *Compute ShortestDist Transitive Edge*.

To demonstrate efficiency of *Compute ShortestDist Transitive Edge* over use of transitive closure algorithms to answer connectivity queries, we compare it to *LogarithmicTC* [107]. Results show *Compute ShortestDist Transitive Edge* has lower computational costs with much smaller result relations.

Experimental Setup

We used the two operators, *Compute ShortestDist Transitive Edge* and *LogarithmicTC* to generate shortest transitive edge relation for the Gene Ontology database. *Compute ShortestDist Transitive Edge* implements *MinJoinLogarithmicTC* This result relation is used later by the macro *BuildGoSlim* to group genes and gene products into broad biological categories that give a higher-level view of their function when analyzing results of a high-throughput experiment. We tested their performance as the size of the graph increased by varying the number of nodes in the input graph. The total size of intermediate JOIN results, together with execution times were recorded. Figures 3.4 and 3.5 present these results.

Figure 3.4 gives the execution times of *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases. The x-axis represents the number of nodes in the input graph. The y-axis lists the execution times (seconds) for the two operators.

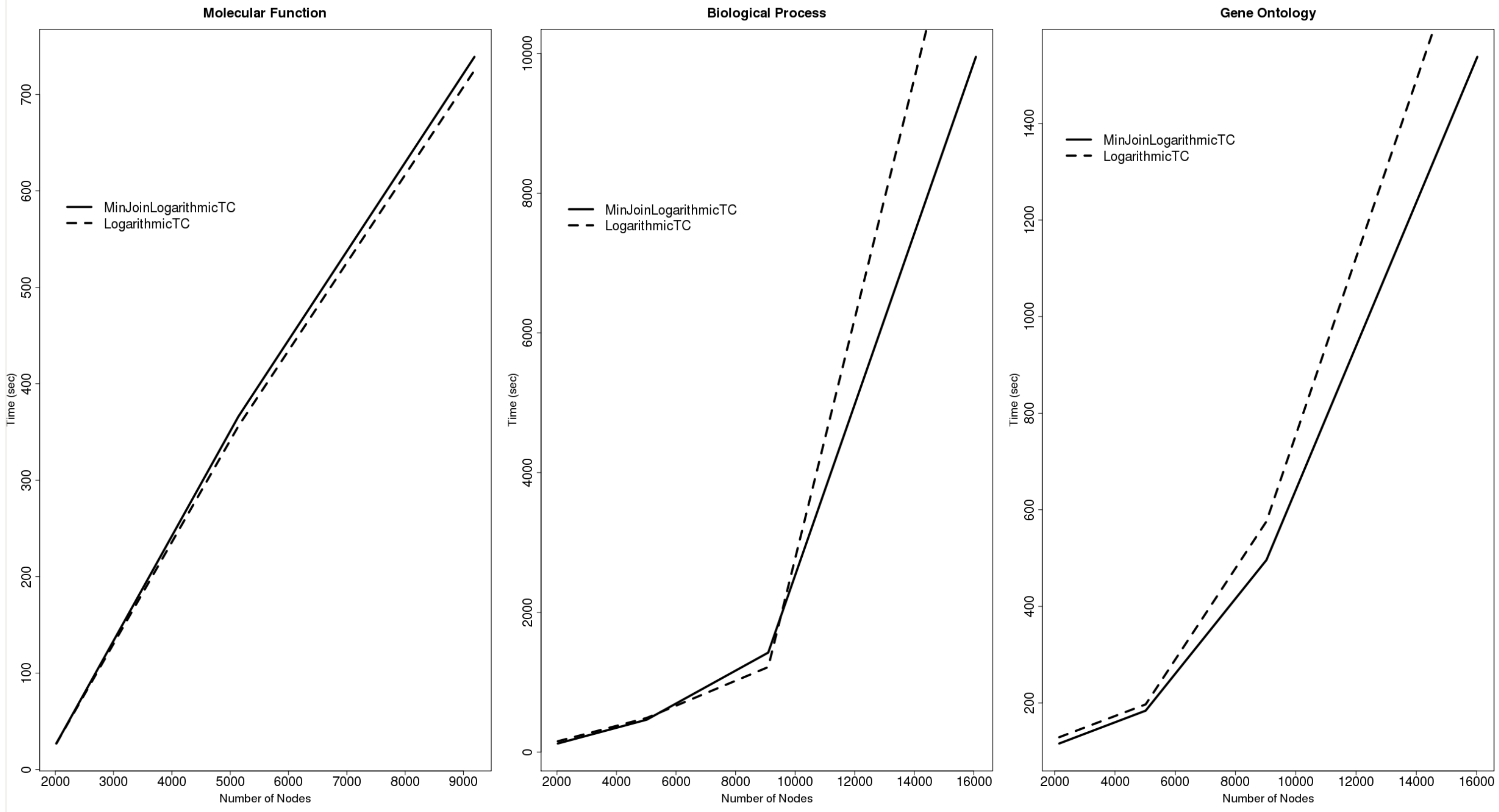


Figure 3.4: Execution times of *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases

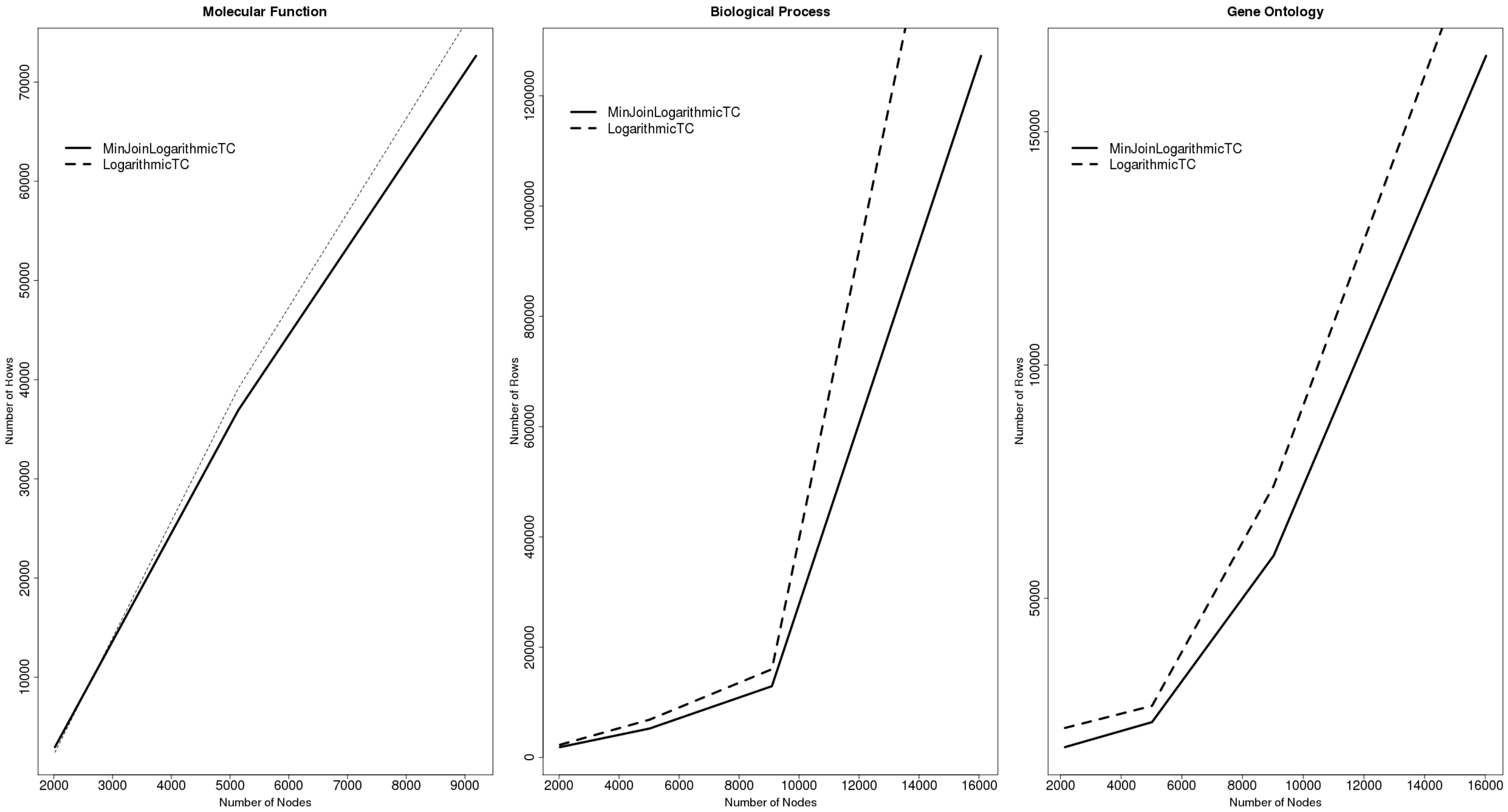


Figure 3.5: Number of rows in intermediate JOIN relations for *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases

Figure 3.5 gives the total size of intermediate JOIN relations for *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases. The x-axis represents the number of nodes in the input graphs. The y-axis lists the size of intermediate result relations before duplicate entries are removed.

For small sized graphs (9000 nodes or less), there was no discernable difference in the execution times betweeen *MinJoinLogarithmicTC* and *LogarithmicTC*. In fact, *LogarithmicTC* appeared to have lower execution times. *MinJoinLogarithmicTC* uses more temporary relations when computing the shortest transitive edge relations. The lower execution times by *LogarithmicTC* for small input size graphs could be a result of the overhead cost of *MinJoinLogarithmicTC* incurred by the higher number of temporary relations maintained by the operator. For small graphs, the overhead cost of creating and maintaining these temporary relations is greater than the time saved by *MinJoinLogarithmicTC*. However, as the size of the graph increases, *MinJoinLogarithmicTC* consistently outperforms *LogarithmicTC*. *MinJoinLogarithmicTC* not only generates the shortest transitive edge relation of a graph, a functionality needed to answer complex biological queries. It does so in a more efficient manner that does not generate duplicate entries in intermediate join relations.

3.1.4 Path Edge Operators

The operator presented in Table 3.10 generates relations used to retrieve each path edge in a path connecting pairs of proteins. Result relations produced are used to determine interactions that link a candidate biomarker to a pathway known to have a role in the disease being studied. Knowing the set of interactions and reactions that link a candidate biomarker to the pathway gives more information about the interaction and will help elucidate its role in disease development [101].

Compute Path Edge

Compute Path Edge takes as input an *edge* relation containing each edge in a pathway and generates a relation containing each path edge for each transitive edge (*path number*, *start node*, *end node*, *position*). The attributes *path number*, *start node* and *end node* identify each path edge.

The precedence chart below (“ PRECEDENCE CHART C ”) show one iteration

Circumstance				Result Relation Name			Result Relation Structure		
Operator	**Type**	**Relation inputs**	**Non-relation inputs**	**Base**	**Row modifier**	**Column modifier**	**Identifier**	**Width**	**Height**
Compute Path Edge	unary	edge	result_rel_name	from ID columns	path edge	path_ID in node, out node, position	path_ID, position	identifier set size + 2	≥ h
	“ *find each path edge in EGF signaling pathway* ”								
Compute inNode Restricted Path Edge	binary	edge inNode	result_rel_name max_distance	from ID columns	path edge in Node	path_ID in Node, out node, position	path_ID, position	identifier set size + 2	≥ h
	“ *find each path edge with beta-catenin as the in Node* ”								

Table 3.10: Operator Summary: Compute Path Edge operators

of *Compute Path Edge*. The same basic steps are outline in the pseudocode in Algorithm 3.3.

Initialization Step

The first PROJECT adds *path_id* (unique for each edge) and *position* (initialized to 1) attributes to each edge (R_1: Line 1, Algorithm 3.3). The REDUCE creates a *path_edge* relation with *path_id, start_node, end_node* and *position*) as the identifying columns (*path_edge*: Lines 2 and 3, Algorithm 3.3). *Path_edge* is used to initialize the result relation (*path_edge_RESULT*: Line 4, Algorithm 3.3).

Extending path by adding new path_edge

Recall, the purpose of *Compute Path Edge* is to retrieve edges defining the path used to determine transitive edges connecting two proteins. The input relation (*Edge*) contains the first edge in the path (*path_edge*). *Compute Path Edge* uses a series of join operations to determine the next path edge. In the first iteration, the MATCH JOIN combines *path_edge* and *edge* to obtain the next edge in the path ($T2$: Line 8, Algorithm 3.3). The FILTER operator is used to ensure the new path edge does not contain a loop connecting a node to itself. This FILTER is especially significant with signaling pathways where modification of a protein e.g., phosphorylation reaction results in a loop between a protein and itself ($T3$: Line 9, Algorithm 3.3).

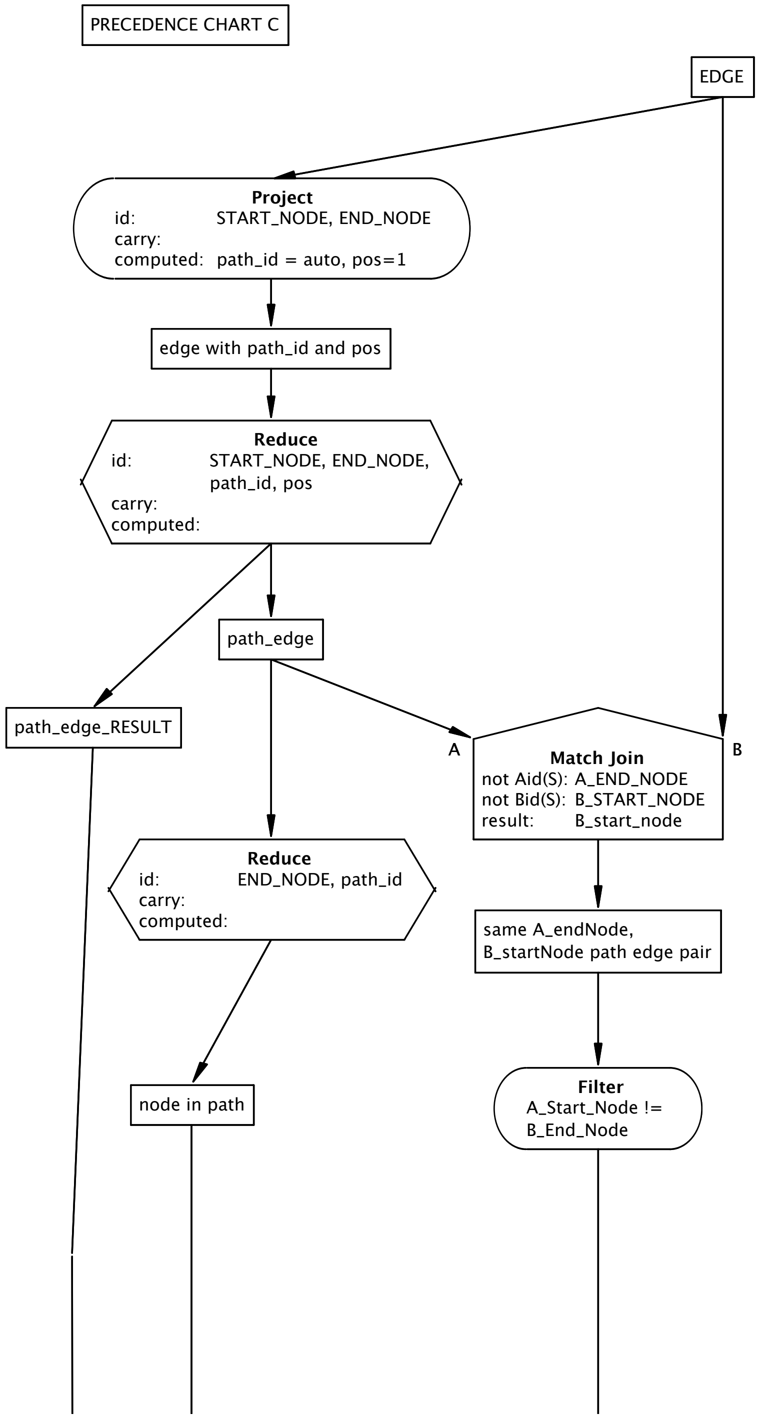
The left REDUCE, on page 2 extracts the end node for the new path edge ($T4$: Line 10, Algorithm 3.3). The MINUS immediately after this REDUCE is used to ensure the added path edge does not form a cycle with any of the other path nodes ($T6$: Lines 7 and 12, Algorithm 3.3).

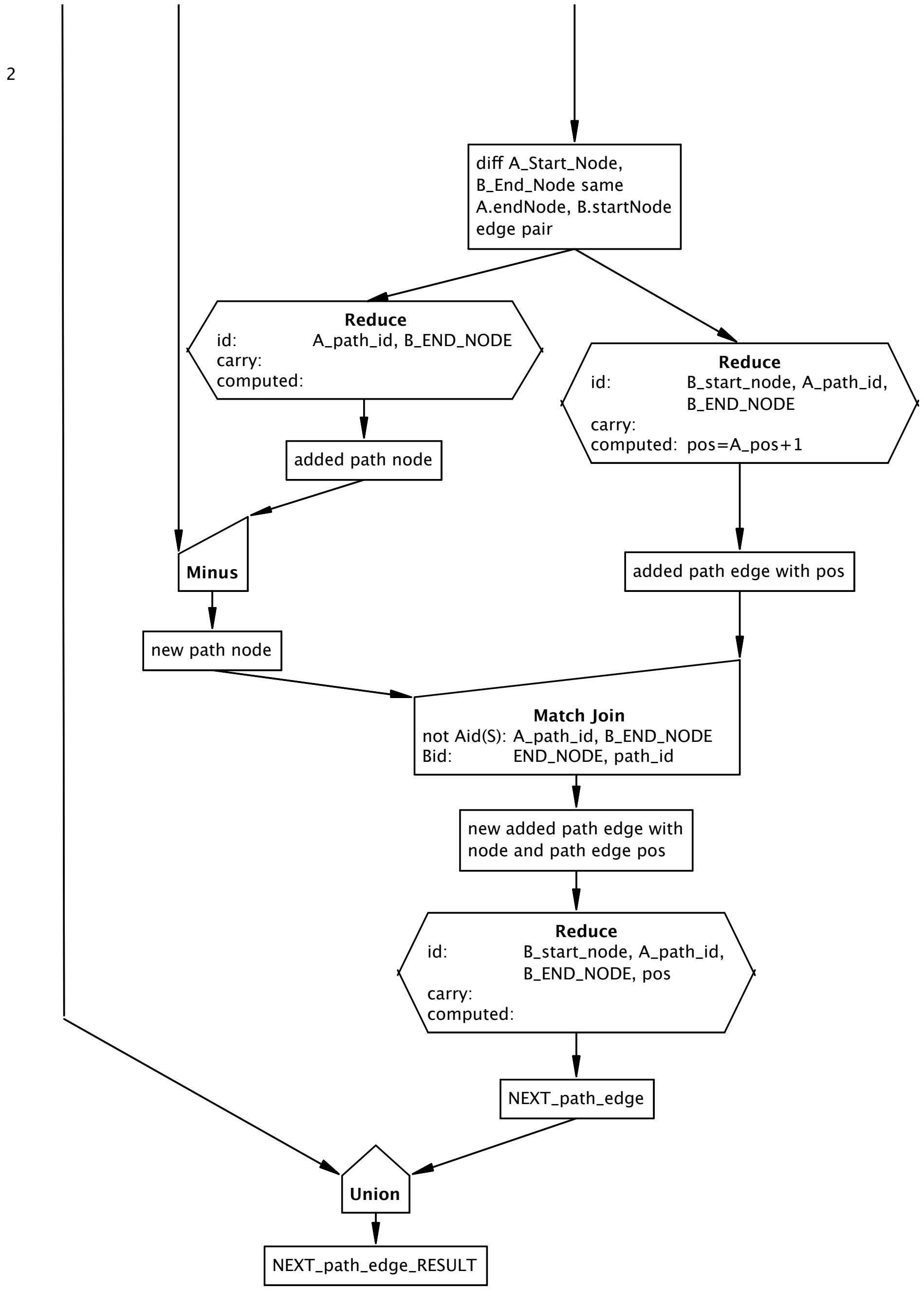
The right REDUCE, on page 2 assigns a position to the new path edge ($T5$: Line 11, Algorithm 3.3). The final MATCH JOIN creates a relation with information about the new *path_edge* ($T7$: Line 13, Algorithm 3.3).

Updating output and intermediate relations

Finally, the REDUCE at the bottom of page 2 generates the next *path_edge* ($T8$: Line 14, Algorithm 3.3). This relation is used to update *path_edge* to contain the new just added *path_edge* ($NEXT\_path\_edge$: Line 16, Algorithm 3.3). The UNION operator is the used to update the result relation to contain the new path edge

($NEXT\_path\_edge\_RESULT$: Line 17, Algorithm 3.3). If NEXT_path_edge has no rows, the operators terminates. Otherwise it continues until no new path edges are added.


PRECEDENCE CHART C
EDGE
Project
id: START_NODE, END_NODE
carry:
computed: path_id = auto, pos=1
edge with path_id and pos
Reduce
id: START_NODE, END_NODE, path_id, pos
carry:
computed:
path_edge
path_edge_RESULT
A
B
Match Join
not Aid(S): A_END_NODE
not Bid(S): B_START_NODE
result: B_start_node
Reduce
id: END_NODE, path_id
carry:
computed:
same A_endNode, B_startNode path edge pair
node in path
Filter
A_Start_Node !=
B_End_Node


2
diff A_Start_Node,
B_End_Node same
A.endNode, B.startNode
edge pair
Reduce
id: A_path_id, B_END_NODE
carry:
computed:
Reduce
id: B_start_node, A_path_id, B_END_NODE
carry:
computed: pos=A_pos+1
added path node
Minus
added path edge with pos
new path node
Match Join
not Aid(S): A_path_id, B_END_NODE
Bid: END_NODE, path_id
new added path edge with
node and path edge pos
Reduce
id: B_start_node, A_path_id, B_END_NODE, pos
carry:
computed:
NEXT_path_edge
Union
NEXT_path_edge_RESULT

Algorithm 3.3 Compute Path Edge (Edge:operand relation; TN:result relation)

```
$R_1 := PROJECT(Edge, [Start\_Node, End\_Node, path\_id = auto, dist = 1])$
$R_2 := REDUCE(R_1, [path\_id, start\_node, end\_node, pos])$
$path\_edge := R_2$
$path\_edge\_RESULT := R_2$
$STOP = 0$
while $STOP == 0$ do
    $T1 := REDUCE(path\_edge, [End\_Node, path\_id])$
    $T2 := JOIN(path\_edge\ A, Edge\ B, [A.End\_Node = B.Start\_Node])$
    $T3 := FILTER(T2, [A.Start\_Node \neq B.End\_Node])$
    $T4 := REDUCE(T3, [A.path\_id, B.End\_Node])$
    $T5 := REDUCE(T3, [A.path\_id, B.S\_Node, B.E\_Node, pos = A.pos + 1])$
    $T6 := MINUS(T5, T1)$
    $T7 := JOIN(T5, T6, [T5.path\_id = T6.path\_id \wedge T5.E\_Node = T6.E\_Node])$
    $T8 := REDUCE(T7, [A.path\_id, B.Start\_Node, B.End\_Node, pos])$
    $T9 := UNION(path\_edge\_RESULT, NEXT\_path\_edge)$
    $NEXT\_path\_edge := T8$
    $NEXT\_path\_edge\_RESULT := T9$
    $IF(NEXT\_path\_edge\ IS\ NULL)\ STOP\ = 1$
end while
```

Compute inNode Restricted Path Edge

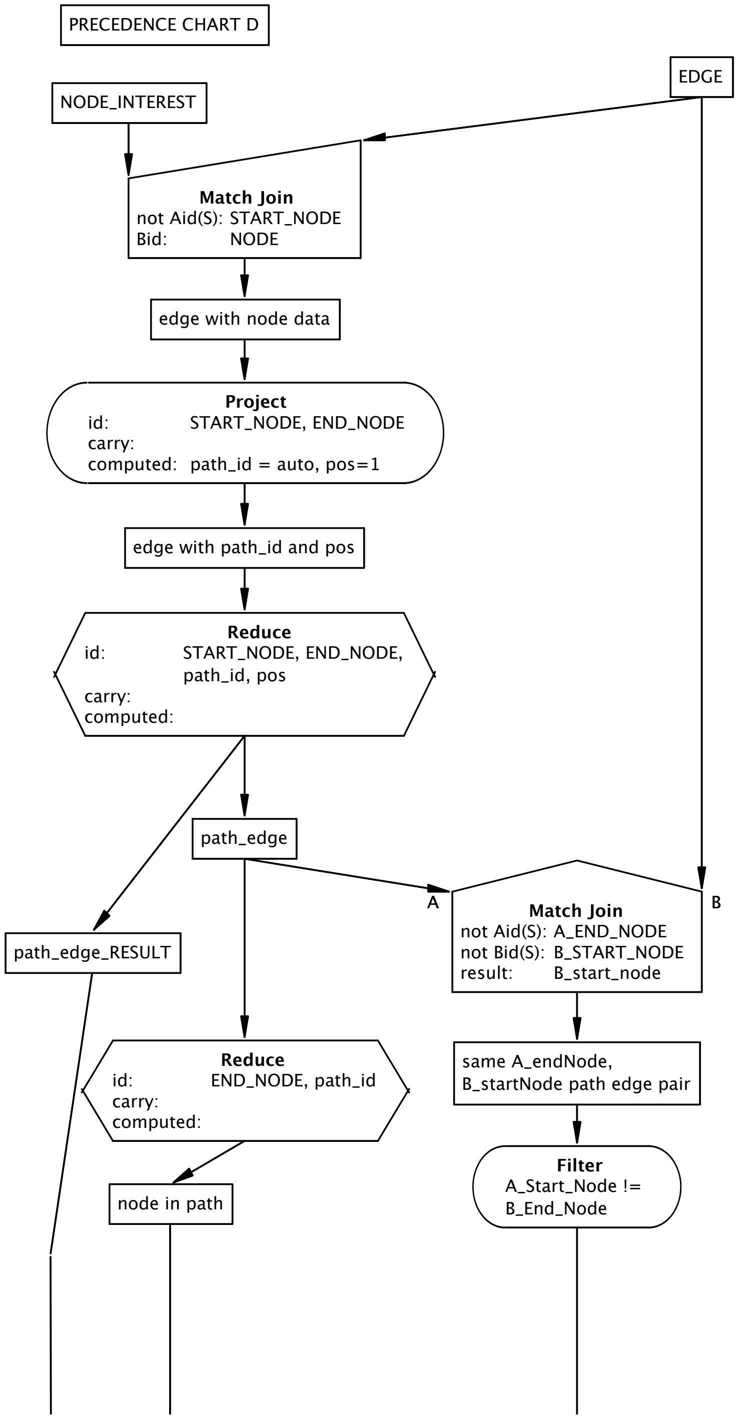
Compute Path Edge retrieves path edges for each transitive edge in a pathway database. In certain instances, however, e.g., when analyzing a set of differentially abundant proteins, it is not necessary to retrieve path edges between each pair of nodes in the pathway database. Retrieving path edges for transitive edges with one of the differentially abundant proteins as the *start node* is sufficient.

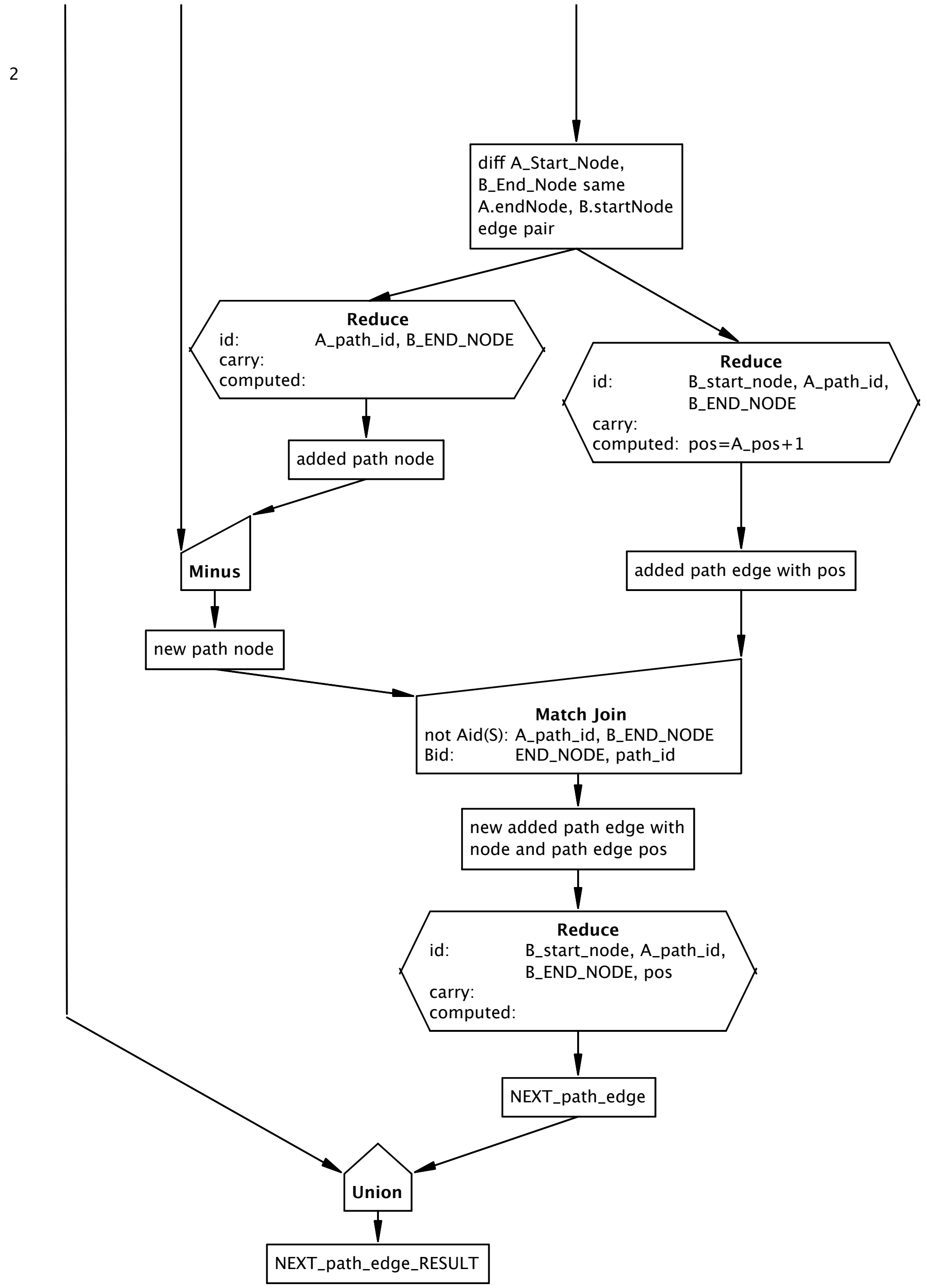
Compute inNode Restricted Path Edge, in addition to taking an *edge* relation as an input, also takes as input a *node* relation that will define the *start node* for each transtive edge for which a path edge is retrieved and a maximum distance parameter that limits the distance for these transitive edges. It produces the same relation as

Compute Path Edge. The precedence chart below (" PRECEDENCE CHART D ") shows one iteration of *Compute inNode Restricted Path Edge*. The same basic steps are outlined in the pseudocode in Algorithm 3.4. With the exception of the initialization step, *Compute inNode Restricted Path Edge* has the same set and sequence of operations as *Compute Path Edge*.

Initialization Step

The first MATCH JOIN is used to ensure each path edge is in a path with one of the differentially abundant proteins (*NODE_INTEREST*) as the first node in the path (P_1: Line 1, Algorithm 3.4). The rest of the operators are identical to *Compute Path Edge*.


PRECEDENCE CHART D
NODE_INTEREST
EDGE
Match Join
not Aid(S): START_NODE
Bid: NODE
edge with node data
Project
id: START_NODE, END_NODE
carry:
computed: path_id = auto, pos=1
edge with path_id and pos
Reduce
id: START_NODE, END_NODE, path_id, pos
carry:
computed:
path_edge
path_edge_RESULT
A
B
Match Join
not Aid(S): A_END_NODE
not Bid(S): B_START_NODE
result: B_start_node
same A_endNode, B_startNode path edge pair
Reduce
id: END_NODE, path_id
carry:
computed:
Filter
A_Start_Node != B_End_Node
node in path


2
diff A_Start_Node, B_End_Node same A.endNode, B.startNode edge pair
Reduce
id: A_path_id, B_END_NODE
carry:
computed:
Reduce
id: B_start_node, A_path_id, B_END_NODE
carry:
computed: pos=A_pos+1
added path node
Minus
added path edge with pos
new path node
Match Join
not Aid(S): A_path_id, B_END_NODE
Bid: END_NODE, path_id
new added path edge with node and path edge pos
Reduce
id: B_start_node, A_path_id, B_END_NODE, pos
carry:
computed:
NEXT_path_edge
Union
NEXT_path_edge_RESULT

Algorithm 3.4 Compute inNode Restricted Path Edge (Edge, Node_Interest :operand relations; X: Max Distance; TN:result relation)

```
$P_1 := JOIN(Edge\ A, Node\_Interest\ B, [A.Start\_Node = B.Node])$
$R_1 := PROJECT(P_1, [Start\_Node, End\_Node, path\_id = auto, dist = 1])$
$R_2 := REDUCE(R_1, [path\_id, start\_node, end\_node, pos])$
$path\_edge := R_2$
$path\_edge\_RESULT := R_2$
$STOP = 0$
while $STOP == 0$ do
    $T1 := REDUCE(path\_edge, [End\_Node, path\_id])$
    $T2 := JOIN(path\_edge\ A, Edge\ B, [A.End\_Node = B.Start\_Node])$
    $T3 := FILTER(T2, [A.Start\_Node \neq B.End\_Node])$
    $T4 := REDUCE(T3, [A.path\_id, B.End\_Node])$
    $T5 := REDUCE(T3, [A.path\_id, B.S\_Node, B.E\_Node, pos = A.pos + 1])$
    $T6 := MINUS(T5, T1)$
    $T7 := JOIN(T5, T6, [T5.path\_id = T6.path\_id \wedge T5.E\_Node = T6.E\_Node])$
    $T8 := REDUCE(T7, [A.path\_id, B.Start\_Node, B.End\_Node, pos])$
    $T9 := UNION(path\_edge\_RESULT, NEXT\_path\_edge)$
    $NEXT\_path\_edge := T8$
    $NEXT\_path\_edge\_RESULT := T9$
    $IF(NEXT\_path\_edge\ IS\ NULL)\ STOP\ = 1$
end while
```

Table 3.11 shows the result relation of using Table 3.3, Table 3.2 and 2 as sample inputs for respectively *edge*, *node_interest* and *maximum distance*, the task is to find each path edge for transtive edges with either *ras, amy1a,* or *stat3* as the start_node for the transitive edge with has a maximum distance of 2.

3.1.5 Rank Node

The operators presented in Tables 3.5 and BasicOpSumType2 are designed for functional analysis of high-throughput proteomics data. Identifying promising candidate biomarkers is an example of functional analysis of high-throughput proteomics data.

Minimum distance transitive edge for part of EGF pathway (Fig 3.3) with Start Node in Table 3.2 and maximum distance of 2		
Start Node	End Node	Distance
ras	cRAF	1
ras	cell growth	2
stat3	stat1	1
stat3	cell growth	1

Table 3.11: Result relation of using relations *edge* (Table 3.3), *StartNode* (Table 3.2) and 2 as input parameters to *Compute inNode Restricted Path Edge*

Rank Node is a macro that uses these operators to identify the most promising candidate biomarkers. It does so by finding each protein in a set of differentially abundant proteins connected to a user specified pathway with known association to the disease or condition being studied and uses these interactions to identify the most promising biomarkers.

Finding these interactions requires the scaling up of connectivity queries from between a pair of proteins to a pair of sets (candidate biomarkers and proteins in pathways known to be associated with the disease). Instead of finding each connection between **protein A** and **protein B**, the connection between each **differentially abundant protein** and each protein in a **pathways of interest** is sought.

Operator Summary

Rank Node has two main phases. The first phase finds each differentially abundant protein connected to a pathway associated with the disease or condition being studied thus identifying promising candidate biomarkers. The second phase ranks each protein, based on length and number of interactuions, thus prioritizing them for follow up validation studies. Table 3.12 summarizes *Rank Node*.

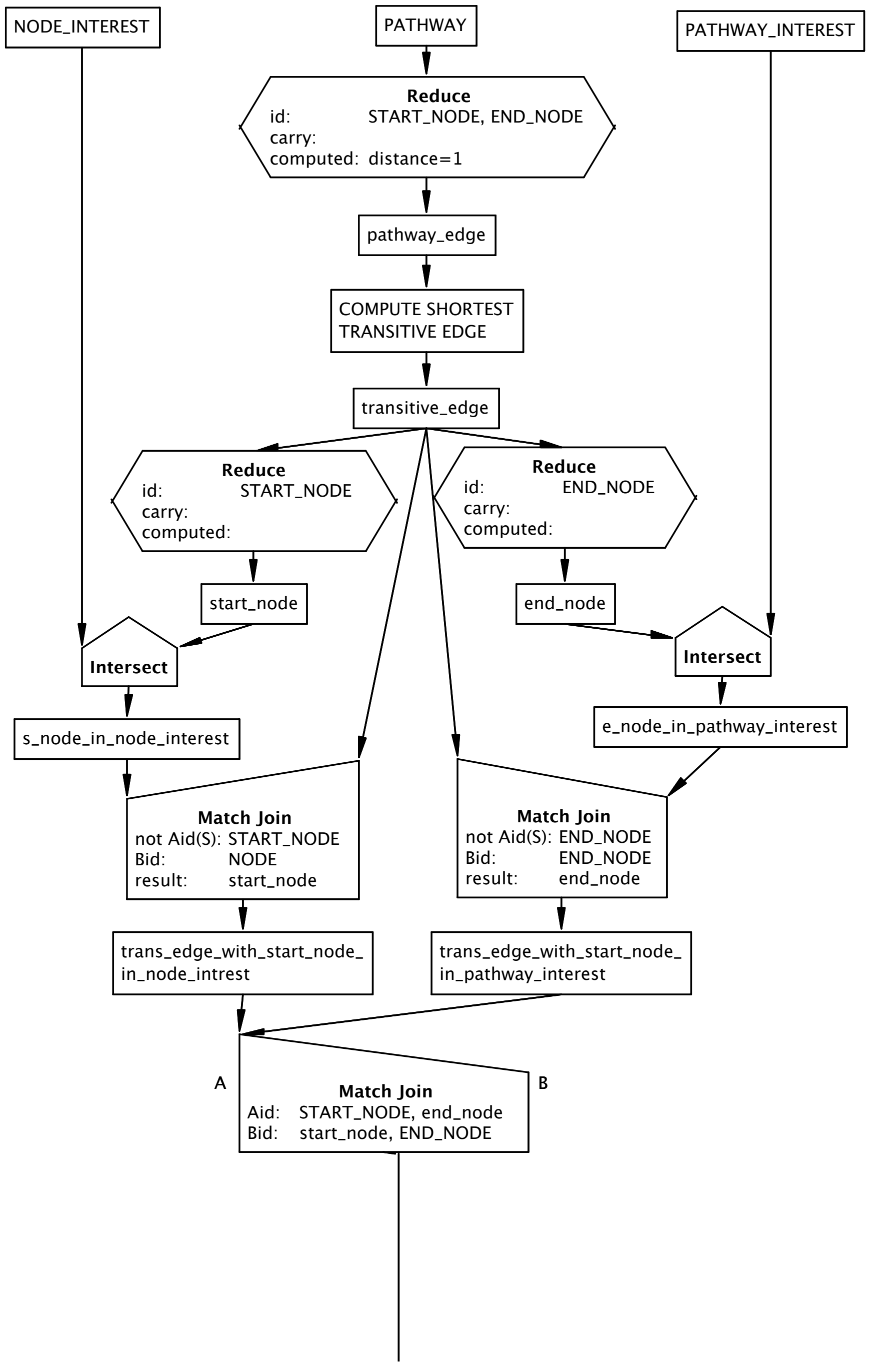
Circumstance				Result Relation Name			Result Relation Structure		
Operator	**Type**	**Relation inputs**	**Non-relation inputs**	**Base**	**Row modifier**	**Column modifier**	**Identifier**	**Width**	**Height**
Rank Node	multi	- edge - node - pathway	result_rel_name	from id columns	node	- node - rank	same as *node* relation	identifier set size + 1	≤ h

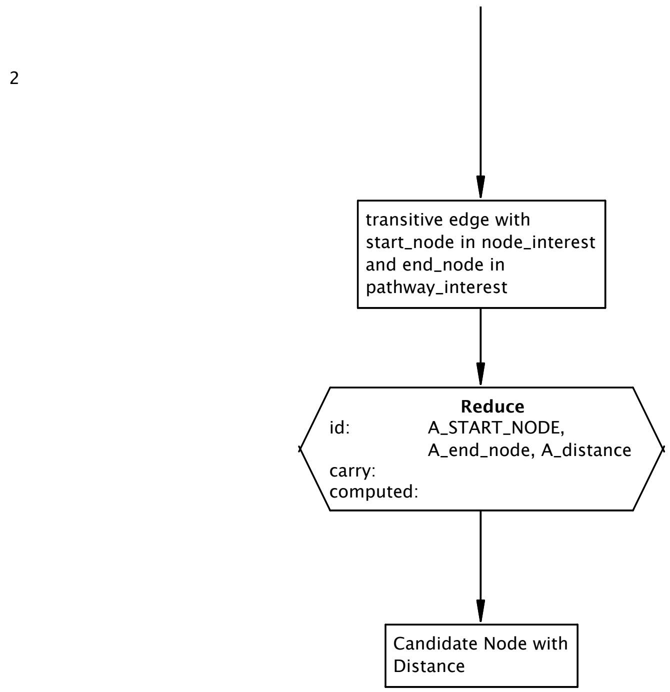
Table 3.12: Operator Summary: Rank Node

Phase One

PRECEDENCE CHART E shows phase one of *Rank Node*. The same steps are shown in the pseudocode in Algorithm 3.5. It takes as input an *edge* relation containing each edge in a pathway, a *node_interest* relation containing nodes of interest e.g., a set of differentially abundant proteins, a *pathway_interest* relation containing nodes in pathways of interest e.g., pathways associated with cancer development, and a *max distance* parameter. For each node in *node_interest*, it finds each path connecting it to nodes in *pathway_interest* with a distance less than or equal to *max distance*. It uses *Operator 1: Compute ShortestDist Transitive Edge* to find each transitive edge in a pathway. It then finds each node in *node_interest* connected to a pathway in *pathway_interest* with distance less than *max distance*. The REDUCE, INTERSECT and MATCH JOIN operators in PRECEDENCE CHART E (Algorithm 3.5) are used to find these nodes. Table 3.13 shows results of the first phase of the operator. *Stat3* is connected to *cell growth* by a direct edge (distance =1) and through stat3 (distance =2) hence the two entries in Table 3.13. The result relation *Candidate Node with Distance* (able 3.13) is used in phase two of *Rank_Node*.

PRECEDENCE CHART E




2
transitive edge with start_node in node_interest and end_node in pathway_interest
Reduce
id: A_START_NODE, A_end_node, A_distance
carry:
computed:
Candidate Node with Distance

Algorithm 3.5 Rank Node (Node_Interest, Pathway, Pathway_Interest, :operand relations; X: Max Distance; TN:result relation)

```
$R_1 := REDUCE(Pathway, [start\_node, end\_node, distance = 1])$
$T_1 := ComputeShortestDistTransitiveEdge(R_1)$
$T_2 := REDUCE(T_1, [Start\_Node])$
$T_3 := REDUCE(T_1, [End\_Node])$
$P_1 := INTERSECT(Node\_Interest, T_2)$
$P_2 := INTERSECT(Pathway\_Interest, T_2)$
$S_1 := MATCHJOIN(T_1\ A,\ P_1\ B, [A.Start\_Node = B.Start\_Node])$
$S_2 := MATCHJOIN(T_1\ A,\ P_2\ B, [A.End\_Node = B.End\_Node])$
$F := MATCHJOIN(S_1\ A,\ S_2\ B, [A.S\_Node = B.S\_Node \wedge A.E\_Node = B.E\_Node])$
$TN := REDUCE(F, [Start\_Node, End\_Node, distance])$
```

Phase Two

In the second phase, the operator first computes two columns; number of paths and mean distance. The nodes are ordered in descending order of number of paths followed by ascending order of mean distance. Each node is given a ranking in an ascending order starting with the protein with the highest number of paths and shortest mean distance. Displaying the rank associated with a row can be done using SQL but there is no straightforward way to do so [108]. Most bench biologists have limited experience with SQL. Bruso [109] developed an operator, PROJECT_RANK, for ranking rows in a database relation and could be used as a helper operator for Molecule Rank. To prevent the need to install PROJECT_RANK, or use SQL to rank the rows, the ranking operation was included as part of *Rank Node.* Table 3.14 shows result of computing the number of paths, mean distance and rank for each protein.

The start node and rank columns are projected as columns for the final result relation which contains differentially abundant proteins and their corresponding rank (Table 3.15). The rank attribute is determined using the number of paths associated with a given node and the mean length for these paths. We use the number of interactions and mean distance to rank the candidate biomarkers because previous findings from

Transitive edge with differentially abundant protein as Start Node and cell growth as End Node		
Start Node	End Node	Distance
stat3	cell growth	1
stat3	cell growth	2
ras	cell growth	2

Table 3.13: Transitive edges for differentially abundant proteins interacting with the cell growth pathway

Differentially expressed protein connected to cell growth with number of paths, mean distance and ranks			
Start Node	Number of path	Mean distance	Rank
stat3	2	1.5	1
ras	1	2	2

Table 3.14: Result relation for computing number of paths, mean distance and rank for differentially abundant proteins connected to cell growth

Differentially expressed protein with corresponding rank	
Differentially expressed protein	Rank
stat3	1
ras	2

Table 3.15: Result relation of using (Table 3.3, Table 3.2 and Table 3.4) as the input parameters to *rank node*

studies of disease causing genes showed the degree (number of interactions) and average length of path are significant features in determining disease causing genes [110, 111]. Lexicographical ordering is used to break ties.

3.1.6 BuildGoSlim

BuildGoSlim is a macro that makes it possible to group genes and gene products into broad biological categories that give a higher-level view of their function when analyzing results of a high-throughput experiment. The broad biological categories are referred to as *GO Slim* terms. For example, using the portion of gene ontology database in Figure 3.2 as sample Gene Ontology database, and the terms *developmental process, meristem maintenance, death* and *muscle attachment* as sample *GO Slim* terms, *BuildGoSlim* would generate a new *association* relation that groups genes and gene products into these broad biological categories. Figure 3.6 shows a graphical representation of this new *association* relation.

Input relations

BuildGoSlim takes as input three relations: *GRAPH_PATH*, *GOSLIM*, and *ASSOCIATION*. Schemas for these relations are shown in Tables 3.16, 3.17 and 3.18 respectively. These schemas contain sample data from Figure 3.6. *GRAPH_PATH* is a transitive closure relation that gives all the descendants of a given term. *GOSLIM* contains the broad biological categories that give a higher-level functional view of genes in an experiment. *ASSOCIATION* gives the gene and gene product annotations to the terms.

Operator Summary

The precedence chart below summarizes the main steps in *BuildGoSlim*. The first *Match Join* finds each descendant term for *GO Slim* terms (T_1: Line 1, Algorithm 3.6). The following *Group* and *Match Join* are used to map the descendant terms to the closest *GO Slim* term (Lines 2 and 3, Algorithm 3.6). Because not every *GO Slim* terms has a descendant term mapping to it, the *Minus* operator is used to identify *GO*

GRAPH_PATH	
TERM1_ID	TERM2_ID
developmental process	death
death	tissue death

Table 3.16: Schema for GRAPH_PATH

GOSLIM
TERM1_ID
developmental process
muscle attachment

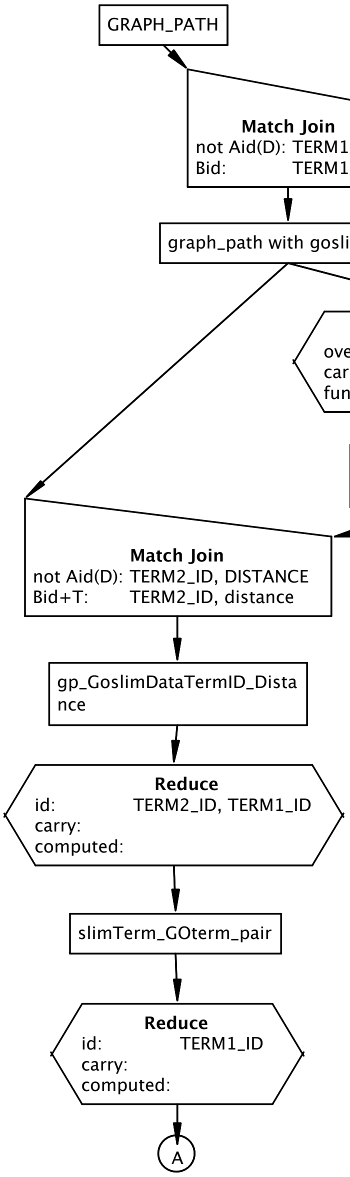
Table 3.17: Schema for GOSLIM

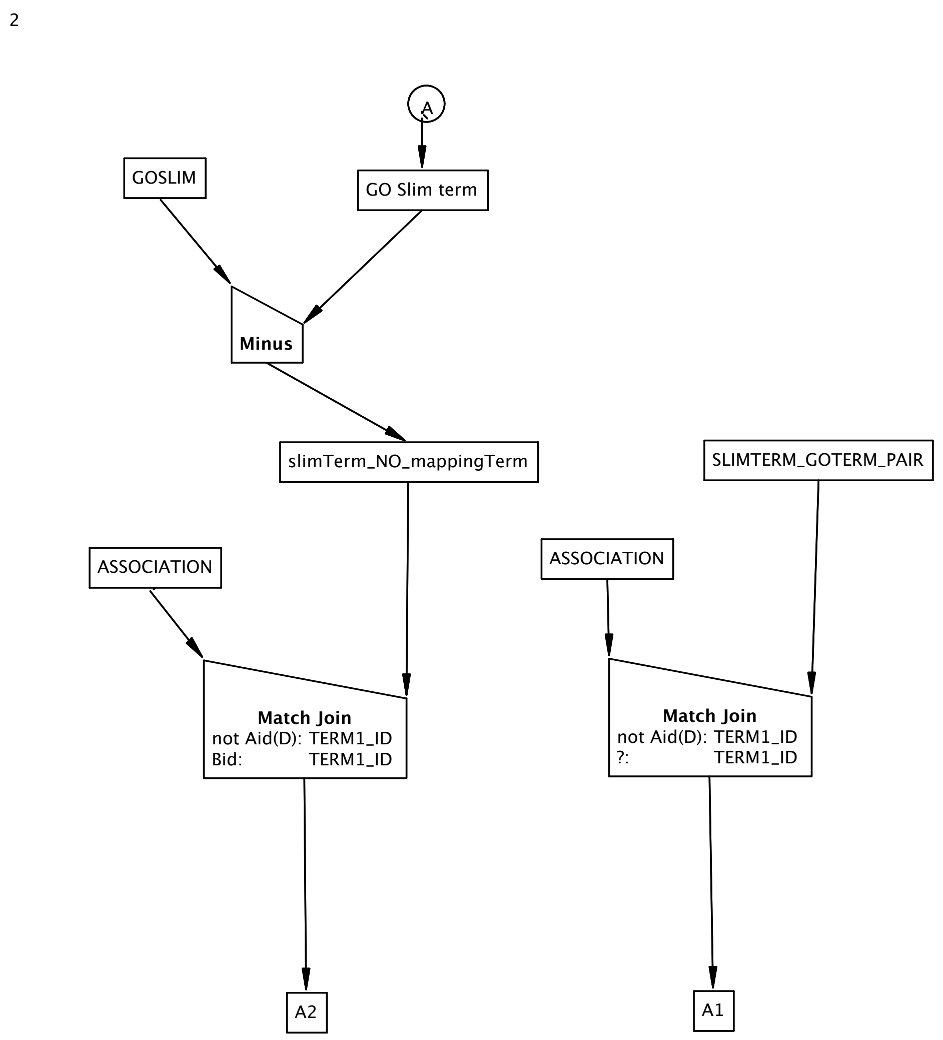
ASSOCIATION		
ASSOCIATION_ID	TERM1_ID	GENE_PRODUCT
1	developmental process	Ak1
2	meristem maintenance	Ap2
2	meristem maintenance	FEY

Table 3.18: Schema for ASSOCIATION

Slim terms with no mapping terms (T_6: Line 6, Algorithm 3.6). To avoid joining two large tables, these terms are joined differently to obtain the result relations (A1 and A2) [Lines 7 and 8, Algorithm 3.6].

Because GO terms can have multiple parents, a term can be mapped to more than one parent term leading to redundancies. It is therefore necessary to identify terms with gene products from descendant terms in order to ensure terms are not mapped to multiple parents. The *Reduce* operators remove these duplicates (Lines 4 and 5, Algorithm 3.6).


GRAPH_PATH
GOSLIM
Match Join
not Aid(D): TERM1_ID
Bid: TERM1_ID
graph_path with goslim data
Group
over: TERM2_ID
carry:
func: distance = min(distance)
termWithDistanceToGOSlim_term
Match Join
not Aid(D): TERM2_ID, DISTANCE
Bid+T: TERM2_ID, distance
gp_GoslimDataTermID_Distance
Reduce
id: TERM2_ID, TERM1_ID
carry:
computed:
slimTerm_GOterm_pair
Reduce
id: TERM1_ID
carry:
computed:
A


2
A
GOSLIM
GO Slim term
Minus
slimTerm_NO_mappingTerm
SLIMTERM_GOTERM_PAIR
ASSOCIATION
ASSOCIATION
Match Join
not Aid(D): TERM1_ID
Bid: TERM1_ID
Match Join
not Aid(D): TERM1_ID
?: TERM1_ID
A2
A1

Algorithm 3.6 BuildGoSlim (GRAPH_PATH A, GOSLIM B : operands; TN : result)

```
$T_1 := JOIN(A, B[A.TERM1\_ID = B.TERM1\_ID])$
$T_2 := GROUP(T_1, [TERM2\_ID, distance = min(distance)])$
$T_3 := JOIN(T_1, T_2[T_1.TERM2\_ID = T_2.TERM2\_ID \wedge T_1.dist = T_2.dist]])$
$T_4 := REDUCE(T_3, [TERM1\_ID, TERM2\_ID])$
$T_5 := REDUCE(T_4, [TERM1\_ID])$
$T_6 := MINUS(B, T_4)$
$A_1 := JOIN(ASSOCIATION\ A, T_6\ B[A.TERM1\_ID = B.TERM1\_ID])$
$A_2 := JOIN(ASSOCIATION\ A, T_4\ B[A.TERM1\_ID = B.TERM1\_ID])$
{Below are relation names for relations above}
$T_1 := graph\_path\ with\ goslim\ data$
$T_2 := term\ with\ distance\ to\ GO\ Slimterm$
$T_3 := graph\_path\ with\ goslim\ and\ distance\ data$
$T_4 := Slim\ term\ GO\ term\ pair$
$T_5 := GOSlim\ term$
$T_6 := Slim\ term\ with\ no\ mapping\ GO\ term$
$A_1 := ASSOCIATION\ for\ goslim\ term\ without\ mapping\ term$
$A_2 := ASSOCIATION\ for\ goslim\ term\ with\ mapping\ term$
```

Result relation

Figure 3.6 shows a graphical representation of the result relation of using Figure 3.2 as sample Gene Ontology database, and the terms *developmental process, meristem maintenance, death* and *muscle attachment* as sample *GO Slim* terms. No terms map to *developmental process* and it contains the same gene products it did in Figure 3.2. *developmental process* is an example of a term in the relation *goslim term without mapping term.* The gene products "*Sr*" and "*Grip*" are mapped from the term "*determination of muscle attachment site*" to the term "*muscle attachment*". *muscle attachment* is an example of a term in the relation *goslim term with mapping term.*

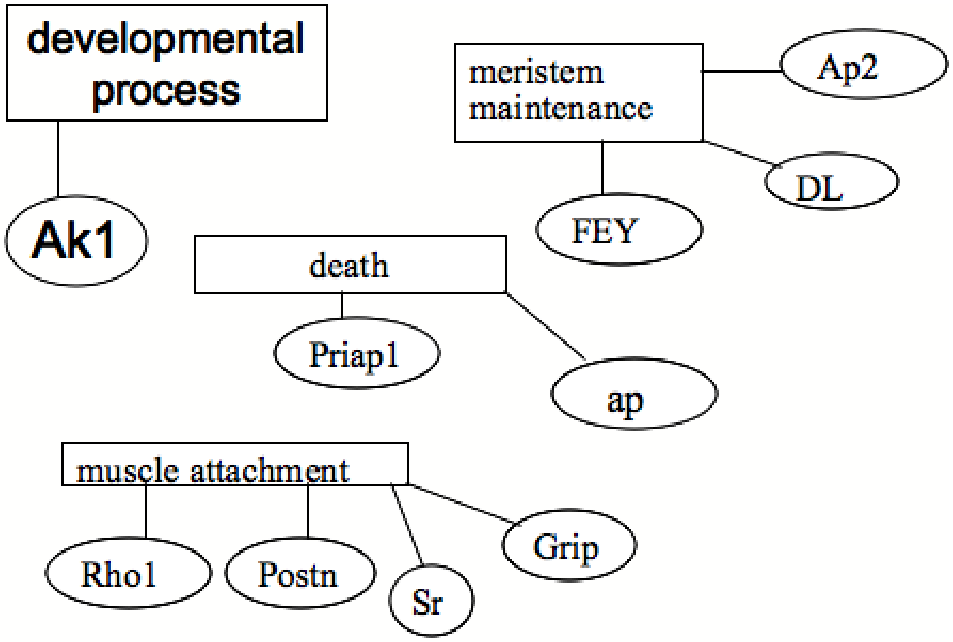


Figure 3.6: Results for *BuildGoSlim* when using Figure 3.2 as sample Gene Ontology database, and the terms *developmental process, meristem maintenance, death* and *muscle attachment* as sample *GO Slim* terms.

3.2 Operators for Biological Reaction Data

In Section 3.1, we presented operators for analyzing biological data expressed by sets of binary relationships. Binary relations, however, cannot capture the complexity of biological reactions. This section presents a datamodel for biological reactions that captures their complexity and new operators that use this datamodel to analyze high-throughput proteomics data.

3.2.1 Data Model

Instead of representing pathways using the *edge* and *Node* relational model shown in Figure 3.7 that tranforms a reaction to a binary protein-protein interactions, we use the datamodel in Figure 3.9 that does not transform a reaction to a protein-protein interaction. This datamodel, shown in Figure 3.9, captures the complexity of biological reactions and retains information lost when using the *Edge* and *Node* datamodel in igure 3.7. Tables 3.21(a), 3.21(b) and 3.21(c) show use of the data model in Figure 3.9 to store STAT3 activation reaction.

As seen in Table 3.21(a) this new data model can capture additional information about biological reactions. The ability to store and query this information will significantly expand the scope of queries for analyzing high-throughput data. For example, users will be able to determine phosphorylation events by querying stored data to determine if ATP and ADP are involved in a reaction.

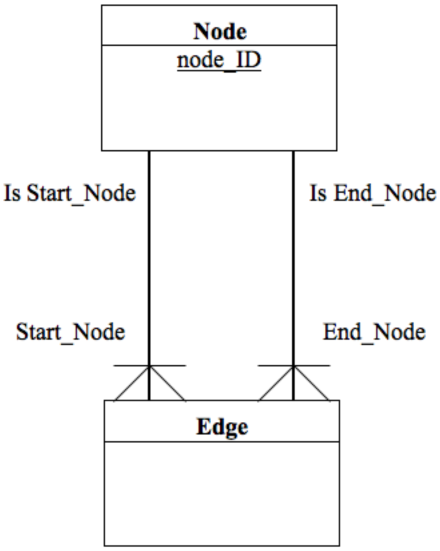


Figure 3.7: Edge-Node Relational Model

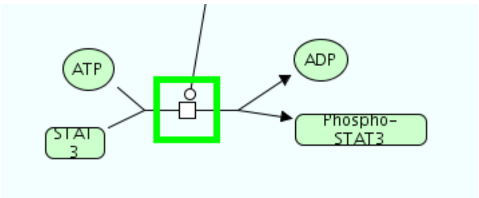


Figure 3.8: STAT3 Activation

StartNode	EndNode
STAT3	STAT3

Table 3.19: *Edge*

Node
STAT3

Table 3.20: *Node*

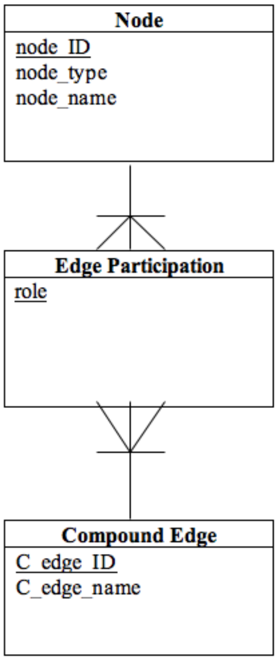


Figure 3.9: Compound Edge Relational Model

Table 3.21: Relations for Compound Edge Relational Model in Figure 3.9

(a) *Node*

node_ID	node_type	node_name
1	protein	STAT3
2	simple molecule	ATP
3	simple molecule	ADP
4	protein	phospho-STAT3

(b) *Edge Participation*

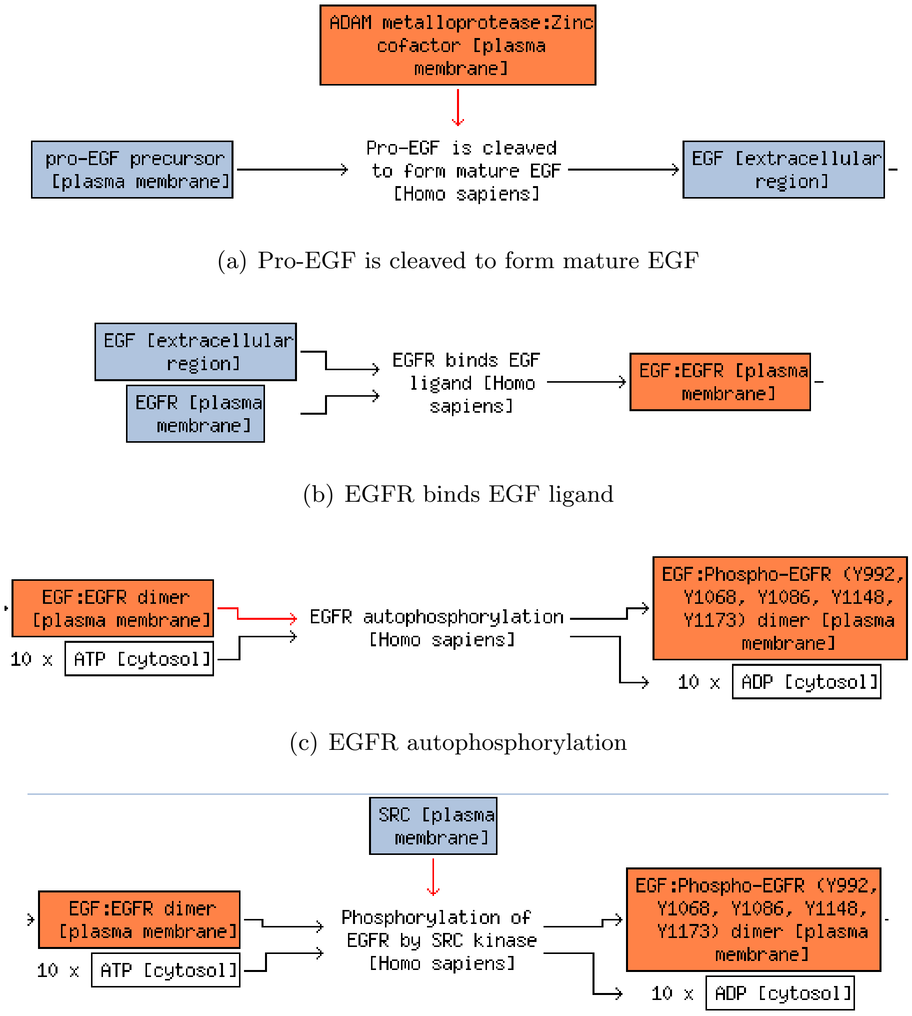
node_ID	role	C_edge_ID
1	substrate	1
2	co-substrate	1
3	co-product	1
4	product	1

(c) *Compound Edge*

C_edge_ID	C_edge_name
1	STAT3 activation

3.2.2 Illustration Data

In Section 3.1.1 we used part of EGF pathway as sample data to describe the operators presented. Again, we use part of EGF pathway shown in Figure 3.10 as sample data. However unlike in section 3.1 where only protein information was presented, additional data such as co-substrates, co-products and sub-cellular localization are presented.


ADAM metalloprotease:Zinc cofactor [plasma membrane]
pro-EGF precursor [plasma membrane]
Pro-EGF is cleaved to form mature EGF [Homo sapiens]
EGF [extracellular region]
EGF [extracellular region]
EGFR [plasma membrane]
EGFR binds EGF ligand [Homo sapiens]
EGF:EGFR [plasma membrane]
EGF:EGFR dimer [plasma membrane]
10 x
ATP [cytosol]
EGFR autophosphorylation [Homo sapiens]
EGF:Phospho-EGFR (Y992, Y1068, Y1086, Y1148, Y1173) dimer [plasma membrane]
10 x
ADP [cytosol]
SRC [plasma membrane]
EGF:EGFR dimer [plasma membrane]
10 x
ATP [cytosol]
Phosphorylation of EGFR by SRC kinase [Homo sapiens]
EGF:Phospho-EGFR (Y992, Y1068, Y1086, Y1148, Y1173) dimer [plasma membrane]
10 x
ADP [cytosol]

(a) Pro-EGF is cleaved to form mature EGF

(b) EGFR binds EGF ligand

(c) EGFR autophosphorylation

(d) Phosphorylation of EGFR by SRC kinase

Figure 3.10: Part of EGF signaling pathway

Note, because databases are content-neutral, content-neutral terms such as *Edge*,

Table 3.22: Relations for Compound Edge Relational Model in Figure 3.9 using Reactions (a) and (d) in Figure 3.10 as sample data.

(a) *Node*

node_ID	node_type	node_name
1	protein	pro-EGF
2	simple molecule	Zn
3	protein	EGF
4	protein	EGFR
5	simple molecule	ATP
6	protein	SRC
7	protein	Phospho-EGFR(Y992,Y1068,Y1086,Y1148,Y1173)
8	simple molecule	ADP

(b) *Edge Participation*

node_ID	role	C_edge_ID
1	substrate	(c)
2	enzyme	(c)
3	product	(c)
3	substrate	(d)
4	substrate	(d)
5	co-substrate	(d)
6	enzyme	(d)
3	product	(d)
7	product	(d)
8	co-product	(d)

(c) *Compound Edge*

C_edge_ID	C_edge_name
(a)	Pro-EGF is cleaved to form mature EGF
(d)	Phosphorylation of EGFR by SRC kinase

Node and *Co-edge* are used for relation and attribute names. In the context of oral cancer research *Edge* and *Co-node* correspond to the terms reaction and enzyme respectively.

Table 3.22 shows relations for *Compound Edge Relational Model* in Figure 3.9. Only two of the reactions in Figure 3.10 (reactions (a) and (d))are shown. In contrast to relations in the *Edge* and *Node* model used in Section 3.1, relations in this new datamodel contain information about enzymes participating in a reaction.

3.2.3 Operator Overview

Next we describe two new operators that use this new datamodel to analyze high-throughput proteomics data. These new operators enable executiong of useful biological queries not possible using relations in Section 3.1 such as " *for each phosphorylated protein, find each downstream protein that is either an enzyme or product in a Reactome reaction* ".

3.2.4 Compute Transitive Start Node - Node Pair

Section 3.1.3 presented an operator that retrieves pairs of proteins connected by protein-protein interactions. This operator worked on a relational model that defined a reaction using a single *start node* and single *end node* represented by a *simple edge*. The operator expects each node to be a protein and to function as either a *start node* or *end node* to an *edge*. This operator performs a similar function, retrieves pairs of nodes connected by a set of reactions. However, it does not make any assumption about the nature of the nodes. A node can be a protein, gene, small molecule, complex or any other compound that is part of a reaction. Also, in addition to being either a *start node* or *end node*, a protein can be a *co-edge*. As previously describe, a *co-edge* is a content-neutral terms used to describe compounds or elements not changed by a reaction e.g., enzymes.

Operator Summary

Compute Transitive Start Node - Node Pair takes as input an *Edge Participation* relation defining reactions in pathway database e.g., Table 3.22(b) and a *Start Node* specifying a protein or gene a user is interested in. It then finds each molecule connected

<table>
<tr><th colspan="4">Circumstance</th><th colspan="3">Result Relation Name</th><th colspan="3">Result Relation Structure</th></tr>
<tr><th>Operator</th><th>Type</th><th>Relation inputs</th><th>Non-relation inputs</th><th>Base</th><th>Row modifier</th><th>Column modifier</th><th>Identifier</th><th>Width</th><th>Height</th></tr>
<tr><td rowspan="2">Compute Transitive Start Node - Node pair</td><td>multi</td><td>- Edge Participation
- Start Node</td><td>result_rel_name</td><td>transitive</td><td>- Start Node</td><td>- Start Node
- Node
- Node Type</td><td>- Start Node
- Node
- Node Type</td><td>identifier set size</td><td>≥ edge h</td></tr>
<tr><td colspan="9">“ find each protein interacting with beta-catenin and has amylase as an enzyme in one of its interactions ”</td></tr>
<tr><td rowspan="2">Compute Start &amp; End Node Restricted Reaction Path</td><td>multi</td><td>- compound edge node
- Start Node
- End Node</td><td>result_rel_name</td><td>- path edge</td><td>- path edge</td><td>- path edge
- position</td><td>- path edge
- position</td><td>2</td><td>≥ h</td></tr>
<tr><td colspan="9"></td></tr>
</table>

Table 3.23: Operators for analyzing high-throughput data using reaction data.

Transitive *start_node* - *node* pair with *node* as *end_node* or *co_edge*		
Start_Node	Node	Node_Type
pro-EGF	Zn	enzyme
pro-EGF	EGF	product
pro-EGF	SRC	enzyme
pro-EGF	Phospho-EGFR(Y992,Y1068,Y1086,Y1148,Y1173)	product
pro-EGF	ADP	co-product

Table 3.24: Result relation for *Compute Transitive Start Node - Node Pair* when using *Edge Participation* and *Start Node* as inputs.

to the *Start Node* by one or more reactions as either an enzyme, product or co-product. This operator is implemented using a combination of relational database operators and PL/SQL programs. For implementation details, see Appendix B.

Table 3.24 shows the result relation of *Compute Transitive Start Node - Node Pair* when using the relation in Table 3.22(b) as input relation (*Edge Participation*) and *pro-EGF* as *Start Node*. This result relation can be used to determine genes or gene products interacting with *pro-EGF* as enzymes, products or co-products of a reaction.

3.2.5 Compute Start & End Node Restricted Reaction Path

Section 3.1.4 presented operators that generate relations used to retrieve each path edge in a path of protein-protein interactions connecting two proteins. These operators worked on a relational model that defined a reaction using a single *start node* and single *end node* (protein-protein interaction) represented by a *simple edge*. *Compute Start & End Node Restricted Reaction Path* performs a similar function but instead of retrieving protein-protein interactions, it retrieves reactions connecting a pair of molecules.

Operator Summary

It takes as input an *Edge Participation* relation defining reactions in pathway database e.g., Table 3.22(b), *Start Node* and *End Node*. It finds each reaction in a path connecting the two molecules (*Start Node* and *End Node*). *Compute Start & End Node*

Restricted Reaction Path is implemented using a combination of relational database operators and PL/SQL programs. For implementation details, see Appendix B.

Table 3.25 shows the result relation of *Compute Start & End Node Restricted Reaction Path* when using the relation in Table 3.22(b) as input relation (*Edge Participation*) and two different sets of *Start Node* and *End Node.* Table 3.25(a) shows the result relation of using *pro-EGF* and *EGF* as *Start Node* and *End Node* respectively. Table 3.25(b) shows the result relation of using *pro-EGF* and *phospho-EGFR* as *Start Node* and *End Node* respectively To obtain more information about these reactions, one can JOIN these result relations with the *Compound Edge* relation to obtain additional information about these reactions.

Table 3.25: Start & End Node Restricted Reaction Path

(a) Result relation for *Compute Start & End Node Restricted Reaction Path* when using *Edge Participation*, *pro-EGF* and *phospho-EGF* as inputs.

Path_ID	C_edge_ID	C_edge_position
1	(a)	1
1	(b)	2
1	(c)	3
1	(d)	3

(b) Result relation for *Compute Start & End Node Restricted Reaction Path* when using *Edge Participation*, *pro-EGF* and *SRC* as inputs.

Path_ID	C_edge_ID	C_edge_position
1	(a)	1
1	(b)	2
1	(d)	3

Chapter 4

Analysis of High-throughput Proteomics Data

In this chapter, we demonstrate use of operators developed in Chapter 3 to analyze high-throughput proteomics data. Using these operators, researchers have the ability to repeatedly perform complex analyses that cannot be performed when using commercial software such as Ingenuity Pathways Analysis (Ingenuity ® Systems,www.ingenuity.com) and, other analysis software [112] where users have no access to the underlying database and database management system extensions. Section 4.1 demonstrates use of *BuildGoSlim* to group genes and gene products into broad biological categories that give a higher-level view of their function. Section 4.2 presents an analysis of high-throughput data in an oral cancer experiment to annotate and identify promising candidate biomarkers worth of follow-up validation studies.

4.1 Analysis of proteomics data using *BuildGoSlim*

The discovery nature of biological science naturally leads to scientists naming what they find. However, giving different names to what turns out to be the same concept and giving different concepts the same name impedes science, making it effectively impossible for humans and computers alike to analyze biological concepts within and especially across different organisms [71]. The Gene Ontology Consortium was formed to help reduce this babel, specifically to " *produce a dynamic, controlled vocabulary that can be* *applied to all eukaryotes even as knowledge of gene and protein roles in cells is accumulating and changing*" [113]. The consortium has successfully encouraged the disciplined use of a common language by establishing, by consensus, a restricted vocabulary, making it publicly available in the Gene Ontology database, GO, and providing mechanisms for its periodic update. Now it is common for GO terms to be used in the research literature and public databases [114, 115].

Tools to produce variants of GO called GO Slims were developed because, for some tasks, such as analyzing the results of an experiment, two characteristics of GO make it less than ideal. First, users may be interested in only a small portion of the entire database and masses of irrelevant terms slow down or interfere with their work. For example, a researcher analyzing whole saliva samples from oral cancer patients may want to restrict analysis to genes or gene products annotated to *epithelial cells.* Second, since each gene or gene product is annotated to the most specific (lowest) sensible GO term, the form of GO makes it inconvenient for users to see or programs to process the annotations associated with the progeny of a term. They would prefer to see the associations of the progeny of each specified term rolled into its associations. SGD [116] gives an example of such a use: " *if you wanted to find all the genes in an expression cluster that were localized to the nucleus, it would be useful to be able to map the granular annotations, such as* ***perinuclear space****, to general terms, such as* ***nucleus***". GO Slims are intended to overcome these troublesome characteristics; indeed, the gene ontology consortium claims, "*GO slims are particularly useful for giving a summary of the results of GO annotation of a genome, microarray, or cDNA collection when broad classification of gene product function is required*" [117].

4.1.1 Related Work

Several tools exist that create GO Slims: OBO-Edit [114], AgBase [115], CGD Gene Ontology Slim Mapper [118], SGD Gene Ontology Slim Mapper [116], and map2slim.pl [117]. Each has notable capabilities and limitations.

OBO-Edit [114] is designed to help users create new ontologies starting from the terms in GO. Because it does not contain term annotations, OBO-Edit alone cannot be used for analyzing. AgBase [115] provides several species-specific, pre-generated GO Slims such as ChickGo and CowGO. AgBase users cannot create a user-specific custom

GO Slim.

CGD Gene Ontology Slim Mapper is a web based tool used to generate a GO Slim, where CGD curators have chosen the GO Slim terms used in Slim Mapper, based "*on annotation statistics and biological significance*" [118]. Slim Mapper allows users to customize, but with two notable limitations. First, starting from that curated set of terms, Slim Mapper allows a user to further restrict the terms and generate a custom GO Slim, but does not give the user the option of choosing terms from the original GO. Second, a user can chose GO Slim terms from only one of the three GO ontologies to include in the custom GO Slim.

The GO consortium provides a Perl script, called *map2slim.pl*, that can be used to generate a customized GO Slim [117]. *map2slim.pl* cannot be seamlessly integrated into applications not written in Perl and users of the script have to update the GO Slim whenever the database is updated. These tools are limited because they support the creation of a static GO Slim, are language dependent, or exist only as part of other applications. *BuildGoSlim* overcomes limitations of existing GO Slim tools.

4.1.2 Application

BuildGoSlim has been used in a number of high-throughput proteomics studies. In a proteomics analysis of cells in whole saliva from oral cancer patients via value-added three-dimensional peptide fractionation and tandem mass spectrometry, over 1900 proteins were identified in whole saliva samples containing mostly exfoliated epithelial cells [24]. One of the initial goals was to identified proteins associated with *epithelium* or any of the *epithelial* processes. We first searched GO for terms with *epithelial* or *epithelium* as part of the term name or description. Both terms were of interest as they describe functions, processes or cellular components associated with *epithelial cells*. We then used the *association* relation that gives the relationship between terms and genes or gene products (proteins) to find all the proteins in our dataset directly annotated to these terms. To find indirect relationships, we took the terms that matched our search for *epithelial* or *epithelium* and used them to generate a GO Slim. We then used the GO Slim to find proteins in our subset indirectly annotated to terms containing *epithelial* or *epithelium* as part of the term name or term description. The use of *BuildGoSlim* returned 20% more proteins relative to the initial search using just the *association* relation which only gave direct annotations.

In another experiment, *BuildGoSlim* was used to demonstrate the diagnotic potential of whole human saliva when using hexapeptide libraries for dynamic range compression [21]. As discussed in the introduction chapter saliva has several features that make it an ideal fluid for biomarker discovery studies. It is easily collected in a noninvasive manner, its molecular constituents significantly overlap with that of plasma (the current gold standard in biomarker studies)[119, 120], and its dynamic range is not as severe as that of plasma. Using hexapeptide libraries for dynamic range compression substantially increases number of identified proteins across across physiochemical and functional categories. To demonstrate use of hexapeptide libraries for dynamic range compression does not affect the diagnostic potential of whole saliva, this study compared saliva proteins identified using these libraries (*post-DRC saliva*) to proteins identified without use of hexapeptide libraries (*untreated saliva*). This analysis would determine if hexapeptide libraries introduced any potential biases i.e., if specific classes or types of proteins might have been enriched/identified post-DRC. Results show proteins in both *post-DRC saliva* and *untreated saliva* exhibit comparable functional diversity and disease linkage demonstrating use of hexapeptide libraries for dynamic range compression does not affect the diagnostic potential of whole saliva.

BuildGoSlim was used to determine distribution of proteins in select Gene Ontology categories (SLIM). Figure 4.1.2, included in a recent publication presenting these results [21], shows the output from *BuildGoSlim*. A survey of these categories demonstrate that *post-DRC saliva* was enriched across all categories. The notable exception was '*plasma membrane*' proteins (within Gene Ontology cellular component) wherein analysis of Untreated Saliva yielded 84 proteins versus 76 proteins identified *post-DRC*. Although this category was not the most extensively populated, it might suggest that hydrophobic proteins (such as those associated with plasma membrane) might not be enriched to the same extent as soluble proteins. In spite of the slightly lower total numbers, *post-DRC* analyses identified 22 new plasma membrane proteins not seen in *Untreated Saliva*, resulting in a total of 106 salivary proteins in this category.

Consistent with saliva being a '*secreted*' fluid, the major GO cellular component categories in saliva are cytoplasmic, organelle and extracellular proteins. With regard to GO biological processes, the largest set of proteins grouped into protein metabolic

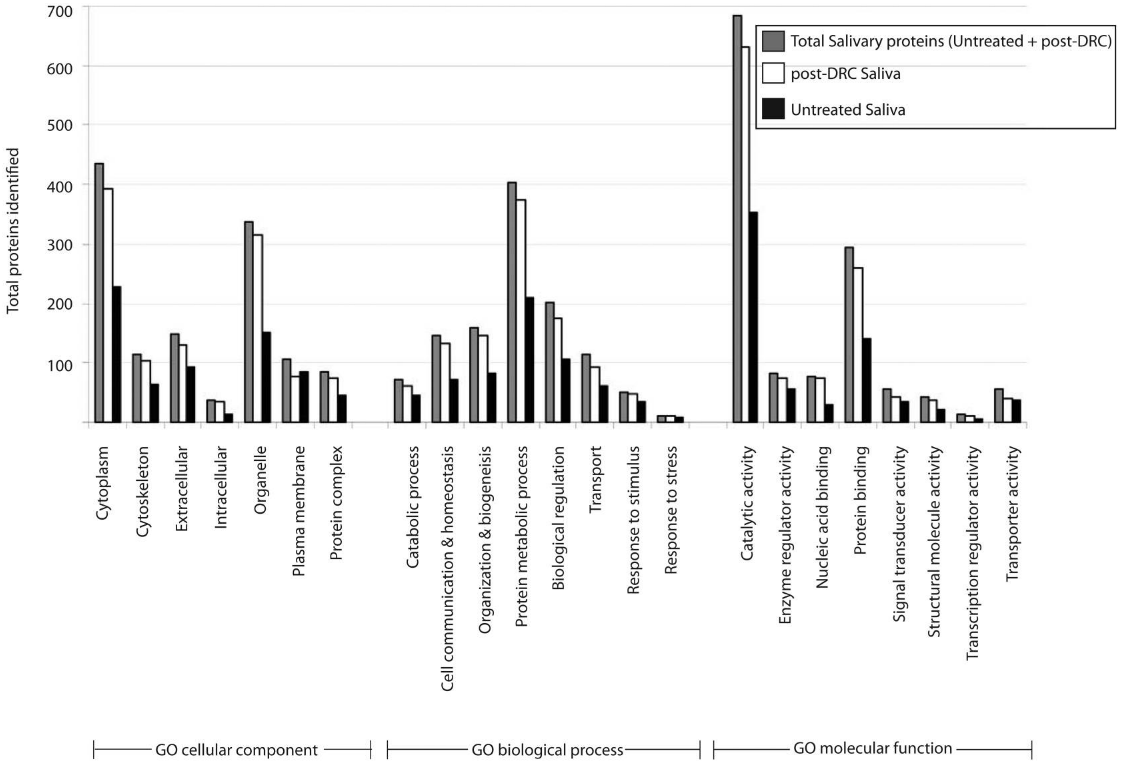


Figure 4.1: Comparison of *post-DRC saliva* and *untreated saliva* using *BuildGoSlim*. *post-DRC saliva* denotes whole saliva proteins identified using hexapeptide beads for dynamic range compression. *untreated saliva* denotes whole saliva proteins identified without use of hexapeptide beads.

categories. In summary, *BuildGoSlim* helped demonstrate dynamic range compression using hexapeptide libraries increases proteins identified without any significant biases toward specific physiochemical or biological functional categories.

In yet another cancer related study, *BuildGoSlim* was used to evaluate the diagnostic potential of a novel technique for collecting tumor interstitial fluid [35]. Tumors lack normal drainage of secreted fluids which leads to build up of tumor interstitial fluid (TIF). TIF likely contains a high proportion of tumor-specific proteins with potential as biomarkers. This novel collection technique uses an ultrafiltration catheter that could damage cells leading to sample contamination.

Identified head and neck squamous cell carcinoma (HNSCC) TIF proteins were classified according to the Gene Ontology term "*cellular components*" to determine cellular localizations. Proteins associated with the cytoskeleton were grouped in "*cytoplasmic*," and proteins associated with the endoplasmic reticulum, Golgi complex, mitochondrion, lysosome, peroxisome, or nucleus were grouped as "*organellar*". In order to assess potential cell lysis caused by this in situ collection technique, the cellular localization of the HNSCC TIF proteome (525 proteins) was compared to a similar sized proteome (524 proteins) identified from lysed cells gathered from brushing the cheek buccal epithelium of a healthy volunteer (Figure 4.1.2). If this novel collection technique caused a significant amount of cell lysis, we expected that the HNSCC TIF proteome would show a similar proportion of cytoplasmic and organellar proteins as compared to the epithelial cell proteome. As expected, the cellular lysate contained a large amount of cytoplasmic (421) and organellar (374) proteins, whereas only 61 proteins were classified as extracellular. On the other hand, HNSCC TIF proteome had lower amounts of proteins classified as cytoplasmic and organellar, 339 and 269, respectively, but a comparatively higher proportion of proteins grouped as extracellular (203 total). The overall percentage of proteins grouped as extracellular for HNSCC TIF was high even compared to known extracellular fluids like saliva. In summary, this analysis using *BuildGoSlim* showed a low level cell lysis associated with the HNSCC TIF sample.

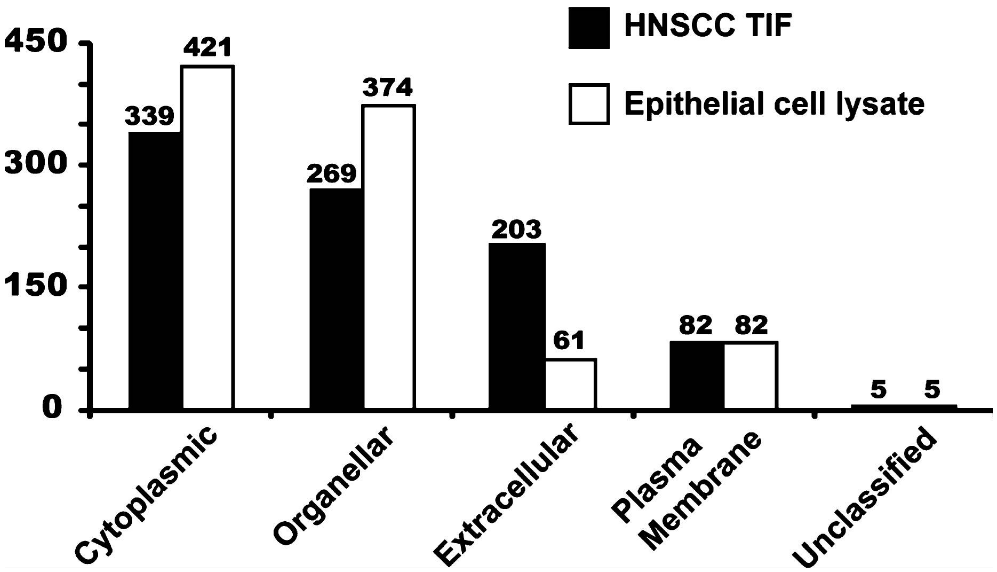


Figure 4.2: Comparison of HNSCC TIF and healthy buccal epithelia cell lysate from brush biopsy using the cellular components Gene Ontology terms. Identified proteins of HNSCC TIF (*solid bars*) and epithelial cell lysate (*open bars*) were categorized into different cellular localizations. Proteins with names associated with cytosol or cytoskeleton were grouped as cytoplasmic. Proteins with localizations for nucleus, mitochondria, the endoplasmic reticulum, the Golgi complex, peroxisomes, or lysosomes were grouped as organellar. Several proteins were grouped into more than one bin. Proteins with no Gene Ontology information or no clear localization were grouped in unclassified.

4.2 Identifying promising candidate biomarkers

A major limitation of existing computational techniques when using high-throughput techniques is results that are too broad to be practically useful [34]. A lot of the ‘potential’ disease-specific biomarkers discovered so far have been found not to be specific to the disease being studied [36]. They either belong to biological categories that change in response to infection or tissue injury, or are proteins whose changes are induced by other stresses such as medication and diet and may have absolutely no relationship to the disease of interest. Needed are computational approaches that limit analyses to the disease condition being studied thus avoiding results that are too broad to be practically useful.

What is a *biological pathway*? It is a molecular network that represents a grouping of functionally related molecules [121, 122]. It has been shown to drive variations in physiological states associated with diseases [123]. Recently Chen et al provided direct experimental support linking complex traits such as obesity to molecular networks [124]. Molecular networks provide a means for limiting analyses to specific diseases or pathways of interest.

The use of molecular networks in high-throughput studies has been demonstrated in a different but related problem, candidate disease gene identification. Inclusion of protein-protein interaction data in candidate disease gene identification analysis has been shown to result in an approximately ten-fold improvement in disease gene identification [125]. Coupled with other types of data such as expression and functional annotation data, protein-protein interaction networks can provide a more targeted approach for identifying candidate disease biomarkers [33].

4.2.1 Background

Several pathways have been identified as playing key roles in the development of complex diseases and have previously been studied as therapeutic targets [62, 63, 65, 64]. Based on its role in regulating cells in human tumors the MAPK signaling pathway has long been viewed as an attractive pathway for anticancer therapies [126]. Metastatic tumors share perturbations in *cell adhesion*, *cytoskeleton remodeling* and *oxidative phosphorylation* pathways regardless of the tissue of origin [127]. Activation of small G protein RAS leads to downstream activation of a number of growth factors, cytokines, and proto-oncogenes and has been associated with cancer [126]. Proteins interacting with pathways known to have a role in disease development are thus good starting point when searching for promising candidate biomarkers.

Because of size and complexity of each biological pathway, pathway databases are increasingly being stored in relational databases [128, 104]. As a result, use of database operators as computational tools to analyze promising candidate disease biomarkers offers several advantages. First, they shift the burden of analysis to the database management system resulting in improved productivity and performance [69]. Second, database operators enable execution of complex queries useful for prioritizing candidate biomarkers. Third, database operators make it possible to repeatedly perform complex analysis enabling refinement of results based on specific experimental needs. Finally, with database operators, developed techniques can be easily integrated with application software.

Database operators give users the ability to examine interactions between candidate biomarkers. The ability to examine these interactions gives additional information on the interplay between candidate biomarkers, useful information in understanding diseases [101]. As a result, these operators will make it easier for translational researchers to manually validate their results [33].

4.2.2 Related Work

Genomic techniques such as microarrays precede high-throughput proteomics techniques and most of the work done on developing computational techniques to analyze high-throughput data has been on genomic technologies. Unfortunately most of these techniques do not translate to high-throughput proteomics technologies. For example, the differential biclustering algorithm for gene expression analysis developed by Odibat et al [129] assumes samples are analyzed independently where a different microarray chip is used for each patient sample. For each gene, it expects a different expression value for each sample with the total number of expression values corresponding to the total number of samples in a given patient group. If a patient group contains 10 different samples this algorithm expects 10 different expression values. In contrast, the novel proteomics technique developed by Griffin et al [24] pools biological samples for each patient group. Instead of obtaining expression values corresponding to each sample, a single value is produced for each identified protein irrespective of the total number of samples in a patient group. This differential biclustering algorithm cannot be applied to data generated using the proteomics technique developed by Griffin et al [24].

A number of techniques have been developed to identify and prioritize candidate disease genes using protein-protein interaction (PPI) data [110, 125, 130, 131, 132] but none for prioritizing candidate disease biomarkers. These techniques use features common to disease genes to identify potential targets. Use of PPI data in prioritizing desease genes motivates, in part, use of pathways to analyze and identify promising candidate disease biomarkers. In both cases, a list of differentially abundant proteins is analyzed to identify potential targets. So why not use techniques developed to identifying and prioritizing disease genes to identify and prioritize candidate biomarkers?

The problem of identifying candidate biomarkers is more complex. With disease genes, knowing whether or not a gene is associated with a disease is sufficient. For disease biomarkers, a gene has to be able to distinguish between different disease groups. Therefore, while features common to disease genes are useful in identifying disease genes they are an undesired characteristic when trying to identify disease specific biomarkers [33]. Consequently, techniques used to identify candidate disease genes cannot be used to identify candidate disease biomarkers, especially for non-mendelian diseases such as cancer likely to share phenotypes. Systematic inflammation, for example, is a phenotype shared by both obesity and type-2 diabetes. Genes associated with systematic inflammation will likely be associated with both diseases and cannot be used as biomarkers.

Similar to techniques developed to identify and prioritize disease genes, database extensions have been developed to analyze high-throughput biological data [133, 134, 135, 136, 137] but none to identify and prioritize disease biomarkers. Most of these database extensions address the problem of extending relational databases to support connectivity queries in the context of analyzing reachability queries [138, 139].

Systems Biology Graph Extender (SBGE) is a research prototype that extends IBM RDBMS DB2 database to support queries over biological networks and graph structures [133]. Biopathways Graph Data Manager (BGDM) is a general purpose graph data management system that can be adapted to support biopathways and protein interaction network databases for microbial organisms [135]. BDBMS is an extensible prototype database management system for supporting biological data whose emphasis is on annotation and provenance management, local dependency tracking, update authorization, and non-traditional access methods such as indexing techniques on multidimensional datasets and compressed data [134]. Periscope/SQ is a declarative system on top of relational database management system and provides a method for querying biological sequences [137]. It is based on PiQA, algebra for querying protein datasets [140].

Other database systems developed for analyzing biological data include PathCase, PQL, PathFinder and PathGen [122, 141, 142, 143]. PathCase is a system with a set of software tools for modeling, storing, analyzing, visualizing, and querying biological data at different levels of detail [122]. It provides a querying tool capable of searching for paths between two nodes. Pathway Query Language (PQL) was developed for querying large protein interaction or pathway databases [141]. It is designed to extract subgraphs with given properties from a graph. PathFinder is a framework for identifying signaling pathways [142]. Given a pair of proteins, PathFinder finds candidate pathway segments between the starting protein and ending protein. Domain specific information is used to include possible missing links thus identifying a potential signaling pathway. PathGen incorporates data from several sources to create transitive connections that span multiple gene interaction databases [143].

To develop automated database tools that use pathways to analyze and identify promising candidate disease biomarkers, the ability to execute connectivity queries is needed. A few of the aforementioned database extensions provide some functionality that can be used to extract paths between molecules [133, 142, 143] or answer some biological related connectivity queries [122, 133, 135, 141]. None, however, provide the necessary functionality needed to analyze candidate biomarkers using user specified pathways. For example, they do not provide a means to select pathways to be used in the analysis. Furthermore, they all attempt to develop generic systems to analyze biological graph data. Consequently, a direct comparison to previous work similar to that presented in chapter 2 (Table 2.1) is not possible.

4.2.3 Application

Next we illustrate use of operators developed in chapter 3 to analyze proteomics data in an oral cancer experiment. First, we use of operators developed for binary relation biological data. Binary relation biological data in an abstraction that simplifies biological reactions. Binding interactions and activation reactions are both abstracted to interactions. We then use operators developed for pathway reaction data to analyze high-throughput proteomics data. Operators developed for pathway reaction data have additional functionalities and can execute queries not possible when using operators developed for binary relation biological data.

Using binary relation biological data

As previously stated, tumors share perturbations in a number of pathways such as cell adhesion, cytoskeleton remodeling and oxidative phosphorylation, regardless of the tissue of origin [127]. To select candidate biomarkers, we identified differentially abundant proteins interacting with pathways associated with cancer development. We reasoned that proteins interacting with known cancer genes are more likely to be involved in the transition to malignant oral cancer. Figure 4.2.3 shows a graphical illustration of this process. Pathways in Reactome known to be associated with cancer were identified and protein-protein interactions used to identify differentially abundant proteins interacting with these pathways.

The ideal of studying upstream or downstream interacting partners as potential biomarkers is supported by a number of studies in the literature [144, 55, 145, 146, 147]. Qu et al study *stat3* downstream genes by persistently activating *stat3* and showing its downstream genes serve as biomarkers in human lung carcinomas [144]. In his insight overview Sawyers [55] proposes a pathway-specific biomarkers approach given the activation state of many pathways can be assessed by examining downstream substrates in the pathway, further supporting our approach.

Experimental Setup

Saliva samples were collected from three patient groups; healthy, pre-malignant, malignant and post-treatment [24]. The healthy patient group exhibited signs of healthy individuals with no oral lesions. Those in pre-malignant group had pre-malignant dysplastic lesions but the lesions had not progressed to malignancy. Patients in the malignant group had malignant oral lesions while the post-treatment group consisted of patients who had undergone treatment. The saliva samples were divided into cellular and soluble portions and analyzed using advanced mass spectrometry-based quantitative proteomics techniques to measure protein abundance as described in section 2.4. The goal was to find proteins differentially abundant between different patient groups and identify candidate biomarkers. Of particular interest was the transition between pre-malignant and malignant oral lesions (oral cancer).

In the soluble portion of saliva, 145 proteins were identified as differentially abundant between pre-malignant and malignant patient groups. Each of the 145 differentially abundant proteins is a candidate biomarker. To identify the most promising biomarkers, follow up validation studies with techniques such as western blotting are needed. These techniques are expensive and time consuming and cannot be employed on a list containing 145 proteins in a timely and cost efficient manner necessitating a means to

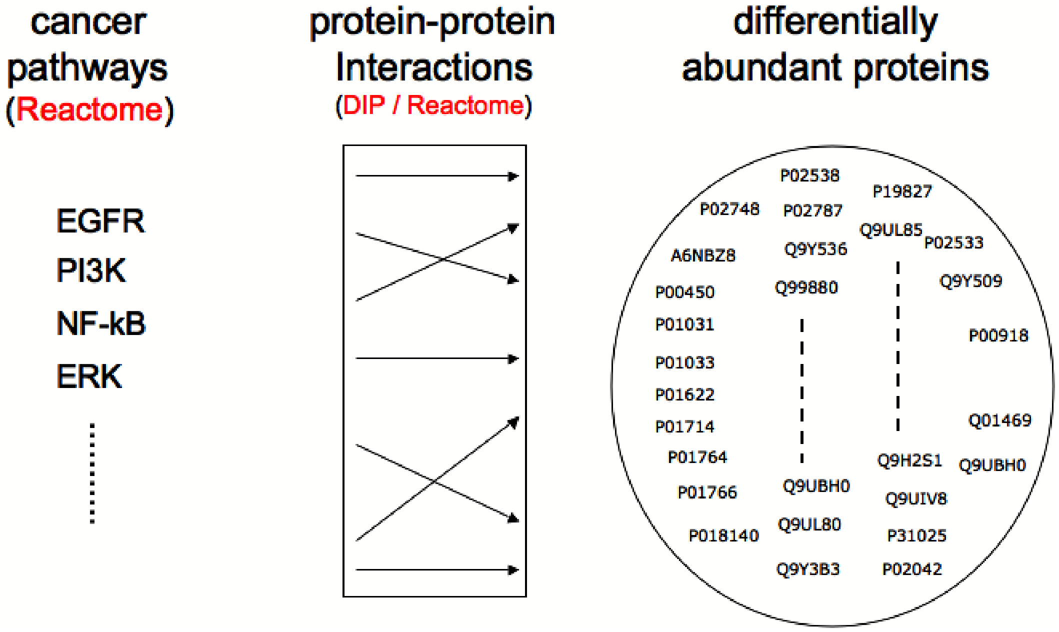


Figure 4.3: Illustration showing use of pathway and PPI data to determine proteins associated with known cancer pathways.

prioritize the list. We demonstrate use of operators that use binary relation biological data to analyze this list to identify promising candidate biomarkers.

Analysis

The 145 differentially expressed proteins (Appendix D: Table D.3) were stored in a relation named *node_interest*. We then downloaded Reactome [104] and DIP [148] databases and combined them into one database. Next, we identified pathways in Reactome known be involved in cancer (Appendix D: Table D.1). Proteins belonging to these pathways were stored in a relation named *pathway_of_interest*. Unlike our example in the illustration data, the *pathway_of_interest* relation contained more than one pathway.

The operator *Compute ShortestDist Transitive Edge* was used to find differentially abundant proteins (node_interest) connected to pathways associated with cancer development (pathway_of_interest) by five or fewer interactions. Because each protein is part of the same complex biological system, at a global level all proteins interact with each other. To ensure our results were not too broad to be practically useful, we needed to restrict the number of interactions permitted between the differentially abundant proteins and pathways associated with cancer development. Additionally, given the small percentage of experimentally verified annotations in pathway databases, limiting the number of interactions between molecules reduces likelihood of finding pairs of molecules connected by incorrectly annotated interactions. Using a distance parameter of four or less in *Compute ShortestDist Transitive Edge* identified less than three proteins as candidate biomarkers. Distances of six to ten did not result in an increase in the number of differentially abundant proteins identified as candidate biomarkers. We thus settled on using five as the distance parameter to *Compute ShortestDist Transitive Edge*.

Table D.2 in Appendix D lists the output of *Compute ShortestDist Transitive Edge*: differentially abundant proteins interacting with pathways associated with cancer development pathways by five or fewer interactions. Next, we ranked these proteins to prioritize them for follow up validation studies. Using the operator *Rank Node*, proteins in Table D.2 were ranked based on number of interactions with pathways associated with cancer development (pathway_of_interest). Note, the 145 proteins could have been directly ranked using *Rank Node*. However, to reduce computational complexity, we used *Compute ShortestDist Transitive Edge* to reduce our search space by first finding differentially abundant proteins linked to pathways associated with cancer. Table 4.1

<table>
<tr><th colspan="4">Differentially abundant soluble protein in oral cancer study identified as one of the promising candidate biomarker with rank signifying priority status of the protein</th></tr>
<tr><th>Node of Interest</th><th>Gene Symbol</th><th>Gene Name</th><th>Rank</th></tr>
<tr><td>P61978</td><td>HNRNPK</td><td>Heterogeneous nuclear ribonucleoprotein K</td><td>1</td></tr>
<tr><td>P31946</td><td>YWHAB</td><td>14-3-3 protein beta/alpha</td><td>2</td></tr>
<tr><td>P35222</td><td>CTNNB1</td><td>Catenin beta-1</td><td>3</td></tr>
<tr><td>P02679</td><td>FGG</td><td>Fibrinogen gamma chain</td><td>4</td></tr>
<tr><td></td><td>UBE2N</td><td>Ubiquitin-conjugating enzyme E2 N</td><td>5</td></tr>
<tr><td>P63104</td><td>YWHAZ</td><td>14-3-3 protein zeta/delta</td><td>6</td></tr>
</table>

Table 4.1: Result relation of using Rank Node to prioritize differentially abundant soluble proteins in oral cancer study

shows the result relation of *Rank Node* appended with annotation information.

The ability to examine interactions between candidate biomarkers and pathways associated with cancer development aids the understanding of disease mechanism [101]. The operator *Compute Path Edge* provides this functionality by retrieving each path edge for a path connecting two proteins. Table 4.2 shows the result relation of using the protein P02679 as the *inNode* input parameter to *Compute inNode Restricted Path Edge* and filtering the result relation to only include paths between the proteins P02679 and P31946.

To demonstrate these operators identified potentially interesting candidate biomarkers, we searched the literature to find out if any of the proteins in our prioritized list

Path edge for path connecting P02679 and P31946 with distance less than or equal to 2			
P ID	Start Node	End Node	Position
11	P02679	P62834	1
11	P62834	P31946	2
14	P02679	P46108	1
14	P46108	P31946	2

Table 4.2: Result relation of using *compute path edge* to find each path with P02679 as Start Node and protein P31946 as End Node and distance less than or equal to 2

(Table 4.1) has known associations to cancer. A summary of the results of this literature search are shown in Table 4.3. Of the six candidate biomarkers **beta catenin** has published literature linking it to pleomorphic adenoma. Pleomorphic adenoma is one of the most common types of salivary gland tumor. Even though it has not been directly linked to oral cancer progression, ubiquitin-conjugating enzyme E2 N plays a role in the error-free DNA repair pathways and contributes to the survival of cells after DNA damage. Given uncontrolled growth is one of the defining characteristics of cancer; expression of a protein contributing to the survival of cells could indicate the outset of cancer.

Rank Node uses a simple criterion of considering the number of connections and average length of paths to rank differentially abundant proteins. Other factors such as post-translational modifications (PTMs) and microRNAs have been shown to contribute to disease development [66, 67]. In the next section we demonstrate use of operators that use pathway reaction data to analyze high-throughput proteomics data. In addition to using PPI data, these operators enable use of information such as post-translational modifications and microRNA data to analyze high-throughput proteomics data.

Protein	**Gene**	**Comments**
P61978	HNRNPK	Likely to play a role in the nuclear metabolism of hnRNAs, particularly for pre-mRNAs that contain cytidine-rich sequences
P31946	YWHAB	Negative regulator of osteogenesis
P35222	CTNNB1	A chromosomal aberration involving CTNNB1 may be a cause of **salivary gland pleiomorphic adenomas**
P02679	FGG	Defects in FGG are a cause of thrombophilia
P61088	UBE2N	Plays a role in the error-free DNA repair pathway and contributes to the survival of cells after DNA damage
P63104	YWHAZ	Adapter protein implicated in the regulation of a large spectrum of both general and specialized signaling pathway

Table 4.3: Annotation information of prioritized list of candidate biomarkers obtained from published literature (www.uniprot.org)

Using pathway reaction data

In chapter 3, we presented operators that use protein-protein interactions in pathway databases to analyze high-throughput proteomics data. Pathway databases, however, contain more than protein-protein interaction data. For example, in addition to pathway names and PPI data Reactome contains additional information such as post-translational modifications.

The ability to query this additional information will significantly expand the scope of queries will help analyze high-throughput proteomics data. For example, post-translational modification events have been shown to contribute to development of diseases. Kruck et al show activation of $mTOR$ in renal cell carcinoma is due to phosphorylation and not protein overexpression as would be expected [66]. A demonstration that post-translational modification events lead to disease development motivates use of this additional information to analyze high-throughput proteomics data. Below, we present an analysis done that uses pathway reaction data and the developed relational database operators to annotate and identify promising candidate biomarkers.

Experimental Setup

In a first-of-its-kind study designed to identify proteins that easily and readily distinguish pre-malignant oral lesions from those already transitioned to malignancy, advanced mass spectrometry-based quantitative proteomics analysis was done on pooled soluble fraction of whole saliva from four subjects with pre-malignant lesions and four with malignant lesions [29]. Proteins differentially abundant between the two groups were identified. The following analysis is of these proteins differentially abundant between the two groups. Details the experimental are presented in [24].

Determing differentially abundant proteins

Relative protein abundance ratios between the two subject groups were calculated from iTRAQ reagent reporter ion intensities. Only proteins identified from two or more MS/MS spectra matched to peptides were considered for quantitative analysis. To account for any systematic errors biasing relative protein abundance ratios, each protein ratio was normalized by calculating the median ratio across the entire set of proteins and dividing each protein ratio by this mean value. Normalization was done

by dividing each ratio by the median ratio in log_2 scale:
normalized ratio = $2^{log_2(proteinratio)-log_2(medianratio)}$.

To determine proteins showing significant abundance differences between the different groups, mean and standard deviation for each ratio was calculated across all matched MS/MS spectra, and used to determine proteins showing significant differences in abundance between the two groups. Proteins with abundance differences greater than one standard deviation from the mean value were deemed to be significantly changing.

Identifying proteins in Reactome

Using Ingenuity Pathway Analysis (IPA) software (Ingenuity Systems, Inc), the differentially abundant proteins were mapped to their respective genes. The proteins together with their gene names, abundance information and annotation data from IPA were stored in the same relational database containing Reactome data. A JOIN operator using gene names was used to identify differentially abundant proteins also in Reactome.

Unfortunately, very few of these differentially abundant proteins were in Reactome. Incompleteness of genomics and proteomics databases is a widely known problem and it has been suggested by some to be "*one of the great challenges for post-genomic biology* "[149]. It is therefore not surprisingly that only five of the proteins identified as significantly changing between the two groups were in reactome. Table 4.4 lists these proteins. Compounding the problem of incomplete annotation is the gene synonym problem [150]. It is therefore likely some of these proteins did not match any the proteins in Reactome because of the gene synonym problem. Nevertheless the five proteins present in Reactome were analyzed using the developed relational database operators.

Establishing plausible biological mechanism

Despite acknowledgment that establishing plausible biological mechanism is essential to complement sophisticated mathematical and statistical techniques designed to control for uncertainity and effects of biases [57], computational biomarker research is still dominated by development of these techniques [151, 152, 153, 154]. Mathematical and statistical techniques can't compensate for data limitations. To establish plausible biological mechanism, it is necessary to know how these differentially abundant proteins

<u>**Protein**</u>	**Gene**	**Description**	$\frac{cancer}{normal}$
IPI00032825	TMED7	transmembrane emp24 protein	1.50
IPI00062037	DYNLL2	dynein, light chain, LC8-type 2	0.58
IPI00103419	SF4	splicing factor 4	1.81
IPI00298961	XPO1	exportin 1 (CRM1 homolog, yeast)	1.60
IPI00017672	NP	nucleoside phosphorylase	0.25

Table 4.4: Differentially abundant proteins present in Reactome database

interact with each other. We next used the operator *Compute Transitive Start Node & Node pair* to determine how the five differentially abundant proteins present in Reactome interact with each other. These five proteins were stored in a relation named *Start Node* and used as one of the input relations (*node*). Table 4.5 shows select attributes and rows from the result relation of *Compute Transitive Start Node & Node pair.* The column *Start Node* has been renamed *Protein.* The column *Gene* is the corresponding gene name for the proteins in *Protein. Node* lists each gene product interact with the protein in the column *Protein.* The column *Node_Type* specifies the role of the gene product in the column *Node* in the reaction interacting with the protein.

Looking at rows 18 and 24 in Table 4.5, there are a series of reactions in Reactome linking the protein *nucleoside phosphorylase* to *Exportin-1* and *Splicing factor 4* both of which are also in the list of differentially abundant proteins. *Exportin-1* is a product of the reaction linking the two proteins while *Splicing factor 4* is an catalyst (enzyme) in the reaction connecting it to *nucleoside phosphorylase.* In addition to interacting with these two proteins, from the result relation we can tell *nucleoside phosphorylase* also interacts with different types of RNAs.

From a cancer biomarker perspective, *nucleoside phosphorylase* and in turn *Exportin-1* and *Splicing factor 4* interact with *Cell division cycle and apoptosis regular protein 1. Apoptosis* is the process of programmed cell death and malfunction of this process can lead to cancer [155]. Three of the five proteins in Reactome interact with each other and one of them, *nucleoside phosphorylase*, interacts with a protein with known association to cancer development. This association between the candidate biomarkers and a protein associated with a cancer development establishes a plausible biological

Row Number	**Protein**	**Gene**	$\frac{cancer}{normal}$	**Node**	**Node_Type**
1	IPI00017672	NP	0.25	RNA-directed RNA polymerase subunit P3	output
2	IPI00017672	NP	0.25	RNA-directed RNA polymerase subunit P1	output
3	IPI00017672	NP	0.25	RNA-directed RNA polymerase subunit P2	output
4	IPI00017672	NP	0.25	H1N1 Genomic RNA Segment 1	output
11	IPI00017672	NP	0.25	H1N1 Genomic RNA Segment 8	output
12	IPI00017672	NP	0.25	Hemagglutinin precursor	output
13	IPI00017672	NP	0.25	Neuraminidase	output
14	IPI00017672	NP	0.25	Matrix protein M1	output
15	IPI00017672	NP	0.25	NEP/NS2	output
16	IPI00017672	NP	0.25	Matrix protein 2	output
17	IPI00017672	NP	0.25	NS2 mRNA	output
18	**IPI00017672**	**NP**	**0.25**	**Exportin-1**	**output**
19	IPI00017672	NP	0.25	NUP210 protein	output
20	IPI00017672	NP	0.25	EMBL	output
21	IPI00017672	NP	0.25	AF033819	output
22	IPI00017672	NP	0.25	Heterogeneous nuclear ribonucleoprotein U-like protein 1	catalyst
23	IPI00017672	NP	0.25	Cell division cycle and apoptosis regulator protein 1	catalyst
24	**IPI00017672**	**NP**	**0.25**	**Splicing factor 4**	**catalyst**
25	IPI00017672	NP	0.25	U1A snRNA	catalyst

Table 4.5: Select attributes and rows from the result relation of *Compute Transitive Start Node & Node pair*

mechanism.

Having only five proteins in the list of differentially abundant proteins be present in Reactome significantly limited the ability to utilize these operators for analysis. For example, with more proteins, one can determine reactions and interactions between the proteins and in turn determine the *degree* of each protein. *Degree* is the number of interactions with other differentially abundant proteins. The *Degree* distribution could then be considered in analysis the differentially abundant proteins to determine promising candidate biomarkers. Protein *degree* is positively correlated with pleiotropic effects on cellular functions [156, 157].

Chapter 5

Conclusion and Future Work

This chapter discusses specific contributions of this dissertation, provides a general summary of their overall impact, and presents proposal for future work.

Cancer, a complex and heterogeneous disease caused largely by abnormalities of the epigenome is the second leading cause of death after heart diseases in the United States (US) [1]. If detected early, while still localized, survival rates for cancer improve drastically. Unfortunately, for oral cancer 5-year survival rate has not significantly improved in the past 30 years [19] due, in part, to lack of reliable biomarkers for early detection [13].

The combination of novel high-throughput mass spectromerty based proteomics techniques on LTQ line of instruments and saliva as a source of candidate biomarkers provides a unique opportunity for identifying reliable biomarkers for oral cancer progression. Preventing adoption of this combination in biomarker discovery studies is lack of software for automated protein quantification and computational techniques needed to identify and prioritize promising candidate biomarkers for follow up validation studies[30].

Below, I summarize work done in this dissertation to overcome this limitation. First, I summarize work done to support protein quantification on LTQ type instruments. Second, I briefly discuss *BuildGoSlim*, a database operator that summarizes results of a high-throughput study. Third, I present a summary of relational database operatorsfor analysis of high-throughput proteomics data using biological graph data. Fourth, I summarize a datamodel and operators for analysis of high-throughput proteomics data using reaction data. Finally, I present a summary of proposed future work.

For accurate quantification from isobaric peptide tagging data, regardless of the mass spectrometer used for analysis, reporter ion intensities must be considered. Furthermore, this quantification technique needs to be made available as software to the wider proteomics community. Ideally such software would, at the very least, meet the following criteria: 1) be compatible with centroided LTQ MS/MS data; 2) employ a technique accounting for errors introduced by low reporter ion intensities, critical for accurate protein quantification [23, 31, 22, 32] ; 3) compatible with different isobaric tagging; and 4) be packed in a freely-available and flexible software pipeline that makes it amenable to individual instruments and possibly even emerging instrumental operation methods (e.g., Orbitrap HCD [38, 39] or ETD [40, 41]). No available software currently meets these criteria.

LTQ-iQuant implements a new technique compatible with centroided LTQ tandem mass spectrometry data that accurately reports protein abundance ratios.

Using an iTRAQ-labeled standard mixture, this new technique was compared to the commercial software Mascot, the only available option for quantifying isobaric peptide data on the LTQ line of instruments. LTQ-iQuant was also compared to Protein Pilot on the 4800 MALDI TOF/TOF, the *defacto* instrument and software standard for iTRAQ-based proteomics, by analyzing an iTRAQ-labeled stem cell lysate. Results of this comparison illuminated two points.

1. For proteins quantified by both LTQ-Orbitrap and 4800 MALDI TOF/TOF, results obtained with the 4800 MALDI TOF/TOF and Protein Pilot were comparable to results with the LTQ-Orbitrap and LTQ-iQuant, validating the accuracy of this new technique.

2. LTQ-Orbitrap and LTQ-iQuant identifies and quantifies significantly more proteins than comparable analysis on the 4800 MALDI TOF/TOF.

These findings are especially significant since the 4800 instrument is currently considered the best option for large-scale iTRAQ-based quantitative analysis [37].

To accurately report protein abundances training data was used to generate a default weight matrix based on collective reporter ion intensities. The training data was obtained by analyzing a standard mixture of tryptic peptides from yeast whole cell lysate labeled with iTRAQ reagents and mixed to known ratios of 10:5:2:1 for, respectively, reporter ion masses of 114, 115, 116 and 117. This weight matrix was then used to estimate error in reported peptide abundances.

In our experience, however, the absolute intensities of reporter ions can vary between different LTQ instruments, and even on the same instruments under different tuning parameters. We would also expect that different isobaric tagging methods (e.g., iTRAQ *versus* TMT) would produce different absolute reporter ion intensities. Given these variations, the relationship of the collective reporter ion intensity on accuracy may differ from the relationship derived from the analysis of our standard iTRAQ-labeled yeast mixture, which was used to generate the default weight matrix.

A key advantage of LTQ-iQuant is the capability of users to input their own training data.

If different tuning parameters, tagging method or LTQ instrument are used, users can analyze a standard mixture, similar to our yeast whole cell lysate mixture, and obtain new training data that more accurately reflects the relationship between reporter ion intensity and accuracy of reported peptide ratio for their experimental setup. This training data can then be used to generate a new weight matrix thus overcoming the limitation of using our default weight matrix leading to the following unique benefits.

1. Users can generate a weight matrix customized to individual instrument performance thus optimizing LTQ-iQuant for different experimental setups.

2. LTQ-iQuant can be used with different isobaric tagging methods (e.g., Tandem Mass Tags(TMT)).

3. LTQ-iQuant will adapt to emerging instrument operations methods (e.g., Orbitrap HCD or ETD).

The ability to create customized weights from user-generated training data should help ensure the most accurate quantitative measures for users of LTQ-iQuant even with different isobaric tagging methods, emerging instrumental operation methods and different tuning parameters.

Besides automating the accurate quantification technique to make it amenable to large-scale quantitative proteomics studies, LTQ-iQuant has several other characteristics that make it attractive.

Instead of distributing just binary code files, the source code together with documentation have been made freely available. LTQ-iQuant is mzXML compatible, accepts isotope purity correction factors and can be used with different iTRAQ tags in experiments where not all labels are used. These characteristics make LTQ-iQuant attractive because of the following reasons.

1. Users have the ability to modify the source code to meet their specific needs.
2. Because it is mzXML compatible it gives users the flexibility to use different database search algorithms.
3. It can be used for both 4-plex and 8-plex iTRAQ reagent-labeling methods and in cases where not all iTRAQ labels are used (e.g., a binary sample comparison where only the 114 and 117 labels are used, etc.).

In high-throughput experiments, it is useful for researchers to be able to group genes or gene products into broad biological categories that give a higher-level view of their function [71].

*This thesis presents **BuildGoSlim** which groups genes and gene products into broad biological categories.*

Plasma is currently the gold standard for biomarker studies [119, 120]. However, there is increasing evidence that other biological fluids such as saliva and tumor interstitial fluid (TIF) have great potential as sources for reliable biomarkers [28, 29, 35]. Previously, characterization of proteins in saliva was limited to high abundance proteins due to the large *dynamic range* problem. Use of TIF in head and neck squamous cell carcinoma (HNSCC) was limited due to lack of satisfactory collection techniques. Bandhakavi et al showed hexapeptide libraries for dynamic range compression substantially increases number of identified proteins in saliva [21]. Stone et al developed a novel technique that uses an ultra-filtration catheter to collect tumor interstitial fluid [35]. To demonstrate use of hexapeptide libraries and ultra-filtration catheter do not affect the diagnostic potential of saliva and TIF, we used *BuildGoSlim* to analyze proteins identified using these techniques and showed:

1. Proteins in both saliva proteins identified using hexapeptide libraries (*post-DRC saliva*) and proteins identified without use of hexapeptide libraries (*untreated saliva*) exhibit comparable functional diversity demonstrating hexapeptide libraries do not affect the diagnostic potential of whole saliva.

2. HNSCC TIF proteins obtained using the ultra-filtration catheter showed a low level cell lysis compared to proteins identified from lysed cells gathered from brushing the cheek buccal epithelium of a healthy volunteer demonstrating this novel collection technique does not cause a significant amount of cell lysis.

When using high-throughput proteomics techniques such as those based on the hybrid LTQ-Orbitrap in discovery-based biomarker studies, it is necessary to use computational techniques to analyze results. These techniques produce lists of hundreds or a few thousand differentially abundant proteins as candidate biomarkers. Existing computational techniques have one major drawback. They produce results that are too broad to be practically useful [34].

This thesis extends the relational database engine to enable use of protein-protein interaction (PPI) data and biological pathway data to analyze high-throughput proteomics data.

Several pathways have been identified as playing key roles in development of complex diseases such as cancer [62, 63, 64, 65] and have previously been studied as therapeutic targets for diseases supporting their use in prioritizing candidate biomarkers. Based on this observation that genes or proteins responsible for development of cancer are expected to interact with disease causing pathways, computational tools capable of elucidating interaction between candidate biomarkers and pathways can be used to identify and prioritize biomarkers [33]. This dissertation presents several operators: *Compute Transitive Edge* operators; *Compute Path Edge* operators; *Rank Node* operator and

BuildGoSlim operator. They use biological pathway information and biological graph data to analyze high-throughput proteomics data leading to the following capabilities.

1. Enable use of biological pathway information and biological graph data to prioritize candidate biomarkers.

2. Makes it possible for users to examine interactions between candidate biomarkers and biological pathways, a useful functionality that aids understanding of disease mechanism.

3. Shifts the burden of analysis to the database management system resulting in improved productivity and performance.

4. Makes it possible to repeatedly perform complex analysis enabling refinement of results based on specific experimental needs.

Using these relational database extensions, we analyzed a dataset of salivary proteins differentially expressed between pre-malignant and malignant oral lesions. Six proteins were identified as candidate biomarkers worth of validation studies. A literature search reveals these high priority candidate biomarkers interact with proteins implicated in cancer development highlighting their potential utility as biomarkers demonstrating the effectiveness of these operators [33].

A graph is an abstract representation of a set of objects where some pairs of the objects are connected by links [158]. Graphs are among the most ubiquitous models of both natural and human-made structures. They have been extensively studied as a mathematical concepts and work has been done in database research to support storage and analysis of graph data [138, 139, 133, 135]. Results from graph theory have been applied to different disciplines such as sociology [159, 160]. More recently, graph and network properties have been used for gene function prediction and disease gene prediction [131].

For biological data that can be represented as graph data, a plethora of techniques and tools exists from relational database research and graph theory that can be adopted for their storage and analysis. The Gene Ontology consortium provides a transitive closure relation "*graph_path*" for answering connectivity queries on the Gene Ontology database [128]. Prior to the use of relational databases to store biological graph data, transitive closure was being used to answer connectivity queries on graph data [161].

Nabieva et al formulate the protein annotation problem as a minimum multiway cut problem [162]. Similar to transitive closure, the minimum cut problem had been extensively studied prior to the existence of high-throughput proteomics data [163].

As pointed out in chapter 4, using graphs and networks to represent biological pathways simplifies their structure. This simplification results in loss of information limiting users queries and analysis. For example, if a protein-protein interaction (PPI) network is used to represent a signal transduction pathway information about participation of small molecules such ATP is lost. ATP is involved in protein phosphorylation, a post-translational modification event associated with disease development. Consequently, use of PPI networks to represent signal transduction pathways does not permit full analysis of high-throughput genomics and proteomics data.

A datamodel is presented that does not reduce biological pathways to PPI networks.

The operators presented above can only be used with datamodels that reduce biological pathways to PPI networks. This new datamodel results in the following additional capabilities.

1. Enables storage of additional reaction information such as enzymes, co-substrates and co-products involved in a biological pathway reaction.

2. Enables execution of queries based on small molecules and non-protein gene products such as microRNAs.

To take advantage of the additional information stored using this new datamodel, new operators are presented.

Compute Transitive Start Node - Node Pair operator and Compute Start & End Node Restricted Reaction Path operator use post-translational modification information and enzymatic information in reactions to analyze proteomics data

These operators enable execution of queries currently not possible when using PPI networks to represent biological pathways. For example, because PPI networks do not contain small molecule information, one cannot retrieve each reaction in a pathway with ATP as one of the participating molecules.

Even though these operators were developed for identification of salivary oral cancer biomarkers, they are content neutral and can be used in different diseases studies. Collectively, these relational database extensions will help overcome one of the main challenges of high-throughput computational techniques; provide a systematic way of bridging the gap between unbiased data driven approach, and hypothesis driven approach to prioritize candidate biomarkers worth of more expensive and time consuming validation studies.

Future Work

As with any research, this work has generated new research questions. Below we list several of these questions and propose future work to address them.

Dual SILAC-iTRAQ multitagging labeling is a new technique that makes it possible to estimate protein turn-over rates without steady state assumption [164]. Extending LTQ-iQuant to support analysis of multitagged data will enable estimation of protein turn-over rates on LTQ type instruments. This extension will include modifying the parser that extracts reporter ions to differentiante heavy label fraction reporter ions from light label fraction reporter ions.

"*Spike-in*" controls are an alternative to "*training data*" for estimating errors in an experiment and was recently used by Hill et al in an iTRAQ experiment [165]. Extending LTQ-iQuant to support accurate quantification of isobaric tagged data with spike-in controls will make it accessible to a wider proteomics community. In future we plan to extend LTQ-iQuant to support analysis of data with "*Spike-in*" controls.

In addition to enabling storage of reactions represented using multiple inputs and multiple outputs, the new datamodel enables storage of sub-cellular information in biological pathways. Plans exist to transform Reactome into an appropriate format that will enable users to execute queries about sub-cellular information.

Better diagnostic biomarkers for the transition between pre-malignant oral lesions and oral squamous cell carcinoma are urgently needed to improve survival rates for oral cancer. Advances in methods and technology now enable construction of a comprehensive biomarker pipeline [18]. Given the heterogeneity and complex nature of cancer, no one individual has sufficient breath of knowledge to effectively perform tasks in the entire pipeline. Unforunately, lack of partnerships with knowledge at different stages of the pipeline is often a key reason for failure of successful translational research [166]. In fact, [167] reviewed cancer literature of studies relating gene expression to patient outcome and found 50% of the publications had at least one flaw serious enough to raise questions about the validity of the conclusion that could have been avoided by collaboration. In collaboration with researchers from Department of Biochemistry, Molecular Biology and Biophysics, Department of Oral Medicine, Diagnosis and Radiology, Department of Biomedical Informatics and Computational Biology, Department of Biostatistics, Department of Pediatrics, and Minnesota Supercomputing Institute I developed computational tools for accurate protein quantification and analysis to identify candidate salivary oral cancer biomarkers.

The best pattern recognizers, especially for complex tasks, are still humans [168]. For example, a well trained pathologist is certain when a tissue is positive for an antibody marker under a light microscope, and there is minimum quantitation needed in order to reach a conclusion. To analyze differentially abundant proteins, I extended the relational database to aid analysis of high-throughput data by cancer researchers. These database extensions supplement statistical techniques developed to control for uncertainity, effects of biases and artifacts. In conclusion, the work I present in this thesis will aid identification of the elusive biomarkers for diagnosis of oral cancer progression.

References

[1] M. Lechner, C. Boshoff, and S. Beck. Cancer epigenome. *Advances in genetics*, 70:247, 2010.

[2] LAG Ries, MP Eisner, CL Kosary, BF Hankey, BA Miller, L. Clegg, A. Mariotto, MP Fay, EJ Feuer, and BK Edwards. SEER cancer statistics review, 1975–2000. *Bethesda, MD: National Cancer Institute*, pages 1975–2000, 2003.

[3] M.A. Haynes and B.D. Smedley. *The unequal burden of cancer: an assessment of NIH research and programs for ethnic minorities and the medically underserved.* National Academies Press, 1999.

[4] WHO. World health organization: Cancer statistics (last accessed date january 7 2011). http://www.who.int/cancer/en/, 2011.

[5] V.A. Triolo and I.L. Riegel. The American Association for Cancer Research, 1907–1940. *Cancer Research*, 21(2):137, 1961.

[6] AN ACT. The National Cancer Act of 1971, 1971.

[7] R. Dulbecco. A turning point in cancer research: sequencing the human genome. *Science(Washington)*, 231(4742):1055–1055, 1986.

[8] M. McGeary and M. Burstein. Sources of cancer research funding in the United States. *Bethesda: National Cancer Institute. Available: http:// www. iom. edu/Object. File/Master/12/783/Fund. pdf. Accessed*, 27, 2006.

[9] MDAnderson. Md anderson cancer center: Quick facts 2010 (last accessed date january 7 2011). http://www.mdanderson.org/about-us/facts-and-history/fact-sheet/index.html, 2011.

[10] J.C. Bailar and H.L. Gornik. Cancer undefeated. *New England Journal of Medicine*, 336(22):1569, 1997.

[11] G. Lewison, A. Purushotham, M. Mason, G. McVie, and R. Sullivan. Understanding the impact of public policy on cancer research: A bibliometric approach. *European Journal of Cancer*, 46(5):912–919, 2010.

[12] P. Anand, A.B. Kunnumakara, C. Sundaram, K.B. Harikumar, S.T. Tharakan, O.S. Lai, B. Sung, and B.B. Aggarwal. Cancer is a preventable disease that requires major lifestyle changes. *Pharmaceutical research*, 25(9):2097–2116, 2008.

[13] S.M. Gapstur and M.J. Thun. Progress in the War on Cancer. *JAMA*, 303(11):1084, 2010.

[14] E. Weiderpass. Lifestyle and Cancer Risk. *Journal of Preventive Medicine and Public Health*, 43(6):459–471, 2010.

[15] American Cancer Society. Cancer Facts and Figures 2009, 2009.

[16] K.J. Martin, M.V. Fournier, G. Reddy, and A.B. Pardee. A Need for Basic Research on Fluid-Based Early Detection Biomarkers. *Cancer research*, 70(13):5203, 2010.

[17] P.D. WAGNER, M. VERMA, and S. SRIVASTAVA. Challenges for biomarkers in cancer detection. *Annals of the New York Academy of Sciences*, 1022(1):9–16, 2004.

[18] N. Rifai, M.A. Gillette, and S.A. Carr. Protein biomarker discovery and validation: the long and uncertain path to clinical utility. *Nature biotechnology*, 24(8):971–984, 2006.

[19] N. L. Rhodus. Oral cancer: leukoplakia and squamous cell carcinoma. *Dent. Clin. North Am.*, 49:143–165, Jan 2005.

[20] I. Messana, R. Inzitari, C. Fanali, T. Cabras, and M. Castagnola. Facts and artifacts in proteomics of body fluids. What proteomics of saliva is telling us? *Journal of separation science*, 31(11):1948–1963, 2008.

[21] S. Bandhakavi, M.D. Stone, G. Onsongo, S.K. Van Riper, and T.J. Griffin. A dynamic range compression and three-dimensional peptide fractionation analysis platform expands proteome coverage and the diagnostic potential of whole saliva. *Journal of Proteome Research*, 8(12):5590–5600, 2009.

[22] M. Bantscheff, M. Boesche, D. Eberhard, T. Matthieson, G. Sweetman, and B. Kuster. Robust and sensitive iTRAQ quantification on an LTQ Orbitrap mass spectrometer. *Molecular & Cellular Proteomics*, 7(9):1702, 2008.

[23] T.J. Griffin, H. Xie, S. Bandhakavi, J. Popko, A. Mohan, J.V. Carlis, and L.A. Higgins. iTRAQ reagent-based quantitative proteomic analysis on a linear ion trap mass spectrometer. *J. Proteome Res*, 6(11):4200–4209, 2007.

[24] Hongwei Xie, Getiria Onsongo, Jonathan Popko, Ebbing P de Jong, Jing Cao, John V Carlis, Robert J Griffin, Nelson L Rhodus, and Timothy J Griffin. Proteomics analysis of cells in whole saliva from oral cancer patients via value-added three-dimensional peptide fractionation and tandem mass spectrometry. *Mol Cell Proteomics*, 7(3):486–98, 2008.

[25] L.F. Hofman. Human saliva as a diagnostic specimen. *Journal of Nutrition*, 131(5):1621S, 2001.

[26] H.P. Lawrence. Salivary markers of systemic disease: noninvasive diagnosis of disease and monitoring of general health. *JOURNAL-CANADIAN DENTAL ASSOCIATION*, 68(3):170–175, 2002.

[27] T.J. Griffin. R01 DE17734 Grant Application, 2007.

[28] Y. Li, M.A.R. St John, X. Zhou, Y. Kim, U. Sinha, R.C.K. Jordan, D. Eisele, E. Abemayor, D. Elashoff, N.H. Park, et al. Salivary transcriptome diagnostics for oral cancer detection. *Clinical Cancer Research*, 10(24):8442, 2004.

[29] E.P. De Jong, H. Xie, G. Onsongo, M.D. Stone, X.B. Chen, J.A. Kooren, E.W. Refsland, R.J. Griffin, F.G. Ondrey, B. Wu, et al. Quantitative Proteomics Reveals Myosin and Actin as Promising Saliva Biomarkers for Distinguishing Pre-Malignant and Malignant Oral Lesions. *PloS one*, 5(6):e11148, 2010.

[30] G. Onsongo, M.D. Stone, S.K. Van Riper, J. Chilton, B. Wu, L.A. Higgins, T.C. Lund, J.V. Carlis, and T.J. Griffin. LTQ-iQuant: A freely-available software pipeline for automated and accurate protein quantification of isobaric tagged peptide data from LTQ instruments. *Proteomics*, 2010.

[31] D.L. Meany, H. Xie, L.D.V. Thompson, E.A. Arriaga, and T.J. Griffin. Identification of carbonylated proteins from enriched rat skeletal muscle mitochondria using affinity chromatography-stable isotope labeling and tandem mass spectrometry. *Proteomics*, 7(7):1150–1163, 2007.

[32] M. Bantscheff, D. Eberhard, Y. Abraham, S. Bastuck, M. Boesche, S. Hobson, T. Mathieson, J. Perrin, M. Raida, C. Rau, et al. Quantitative chemical proteomics reveals mechanisms of action of clinical ABL kinase inhibitors. *Nature biotechnology*, 25(9):1035–1044, 2007.

[33] Getiria Onsongo, Hongwei Xie, Timothy J. Griffin, and John V. Carlis. Relational operators for prioritizing candidate biomarkers in high-throughput differential expression data. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, BCB '10, pages 25–34, New York, NY, USA, 2010. ACM.

[34] M Oti and H G Brunner. The modular nature of genetic diseases. *Clin Genet*, 71(1):1–11, 2007.

[35] M.D. Stone, R.M. Odland, T. McGowan, G. Onsongo, C. Tang, N.L. Rhodus, P. Jagtap, S. Bandhakavi, and T.J. Griffin. Novel In Situ Collection of Tumor Interstitial Fluid from a Head and Neck Squamous Carcinoma Reveals a Unique Proteome with Diagnostic Potential. *Clinical Proteomics*, pages 1–8, 2010.

[36] X. Ye, J. Blonder, and T.D. Veenstra. Targeted proteomics for validation of biomarkers in clinical samples. *Briefings in Functional Genomics*, 8(2):126, 2009.

[37] M.A. Kuzyk, L.B. Ohlund, M.H. Elliott, D. Smith, H. Qian, A. Delaney, C.L. Hunter, and C.H. Borchers. A comparison of MS/MS-based, stable-isotope-labeled, quantitation performance on ESI-quadrupole TOF and MALDI-TOF/TOF mass spectrometers. *Proteomics*, 9(12):3328–3340, 2009.

[38] Y. Zhang, M. Askenazi, J. Jiang, C.J. Luckey, J.D. Griffin, and J.A. Marto. A robust error model for iTRAQ quantification reveals divergent signaling between oncogenic FLT3 mutants in acute myeloid leukemia. *Molecular & Cellular Proteomics*, 9(5):780, 2010.

[39] Y. Zhang, S.B. Ficarro, S. Li, and J.A. Marto. Optimized Orbitrap HCD for quantitative analysis of phosphopeptides. *Journal of the American Society for Mass Spectrometry*, 20(8):1425–1434, 2009.

[40] H. Han, D.J. Pappin, P.L. Ross, and S.A. McLuckey. Electron transfer dissociation of iTRAQ labeled peptide ions. *J. Proteome Res*, 7(9):3643–3648, 2008.

[41] D. Phanstiel, Y. Zhang, J.A. Marto, and J.J. Coon. Peptide and protein quantification using iTRAQ with electron transfer dissociation. *Journal of the American Society for Mass Spectrometry*, 19(9):1255–1262, 2008.

[42] D.F. Ransohoff. Rules of evidence for cancer molecular-marker discovery and validation. *Nature Reviews Cancer*, 4(4):309–314, 2004.

[43] R.L. Stears, T. Martinsky, M. Schena, et al. Trends in microarray analysis. *Nature medicine*, 9(1):140–145, 2003.

[44] D.F. Ransohoff. Cancer: developing molecular biomarkers for cancer. *Science*, 299(5613):1679, 2003.

[45] N.E. Savin. Multiple hypothesis testing. *Handbook of Econometrics*, 2:827–879, 1984.

[46] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[47] M.H. Katz. Multivariable analysis: a primer for readers of medical research. *Annals of internal medicine*, 138(8):644, 2003.

[48] C. Wei, J. Li, and R.E. Bumgarner. Sample size for detecting differentially expressed genes in microarray experiments. *BMC genomics*, 5(1):87, 2004.

[49] R. Simon, M.D. Radmacher, and K. Dobbin. Design of studies using DNA microarrays. *Genetic Epidemiology*, 23(1):21–36, 2002.

[50] C. Ambroise and G.J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10):6562, 2002.

[51] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[52] R.G. Miller. Simultaneous statistical inference. *SPRINGER-VERLAG INC., 175 FIFTH AVE., NEW YORK, NY, 1981, 300*, 1981.

[53] J.D. Storey and R. Tibshirani. *Estimating the positive false discovery rate under dependence, with applications to DNA microarrays.* Dept. of Statistics, Stanford University, 2001.

[54] J.A. Ludwig and J.N. Weinstein. Biomarkers in cancer staging, prognosis and treatment selection. *Nature Reviews Cancer*, 5(11):845–856, 2005.

[55] Charles L Sawyers. The cancer biomarker problem. *Nature*, 452(7187):548–52, 2008.

[56] Wikipedia. Wikipedia (last accessed date feb 24 2011). http://en.wikipedia.org/wiki/Epidemiology, 2011.

[57] G. Taubes and C.C. Mann. Epidemiology faces its limits. *Science(Washington)*, 269(5221):164–164, 1995.

[58] DMP Thomson, J. Krupey, SO Freedman, and P. Gold. The radioimmunoassay of circulating carcinoembryonic antigen of the human digestive system. *Proceedings of the National Academy of Sciences of the United States of America*, 64(1):161, 1969.

[59] P. Lo Gerfo, J. Krupey, and H.J. Hansen. Demonstration of an antigen common to several varieties of neoplasia. *New England Journal of Medicine*, 285(3):138–141, 1971.

[60] N. Zamcheck, TL Moore, P. Dhar, and H. Kupchik. Immunologic diagnosis and prognosis of human digestive-tract cancer: carcinoembryonic antigens. *New England Journal of Medicine*, 286(2):83–86, 1972.

[61] D.F. Ransohoff and A.R. Feinstein. Problems of spectrum and bias in evaluating the efficacy of diagnostic tests. *New England Journal of Medicine*, 299(17):926–930, 1978.

[62] E.D. Cohen, K. Ihida-Stansbury, M.M. Lu, R.A. Panettieri, P.L. Jones, and E.E. Morrisey. Wnt signaling regulates smooth muscle precursor development in the mouse lung via a tenascin C/PDGFR pathway. *J Clin Invest*, 119(9):2538–49, 2009.

[63] Jianqiang Ding, Dongmei Song, Xiaobing Ye, and Shu Fang Liu. A pivotal role of endothelial-specific NF-kappaB signaling in the pathogenesis of septic shock and septic vascular dysfunction. *J Immunol*, 183(6):4031–8, 2009.

[64] Hue H Luu, Ruiwen Zhang, Rex C Haydon, Elizabeth Rayburn, Quan Kang, Weike Si, Jong Kyung Park, Hui Wang, Ying Peng, Wei Jiang, and Tong-Chuan He. Wnt/beta-catenin signaling pathway as a novel cancer drug target. *Curr Cancer Drug Targets*, 4(8):653–71, 2004.

[65] Guoqiang Zhang, Kelly A Kernan, Alison Thomas, Sarah Collins, Yumei Song, Ling Li, Weizhong Zhu, Renee C Leboeuf, and Allison A Eddy. A novel signaling pathway: fibroblast nicotinic receptor alpha1 binds urokinase and promotes renal fibrosis. *J Biol Chem*, 284(42):29050–64, 2009.

[66] Stephan Kruck, Jens Bedke, Jorg Hennenlotter, Petra A Ohneseit, Ursula Kuehs, Erika Senger, Karl-Dietrich Sievert, and Arnulf Stenzl. Activation of mTOR in renal cell carcinoma is due to increased phosphorylation rather than protein overexpression. *Oncol Rep*, 23(1):159–63, 2010.

[67] R. Garzon, G. Marcucci, and C.M. Croce. Targeting microRNAs in cancer: rationale, strategies and challenges. *Nature Reviews Drug Discovery*, 9(10):775–789, 2010.

[68] Y. Yarden. The EGFR family and its ligands in human cancer:: signalling mechanisms and therapeutic opportunities. *European Journal of Cancer*, 37:3–8, 2001.

[69] M.V. Mannino and L.D. Shapiro. Extensions to query languages for graph traversal problems. *IEEE Transactions on Knowledge and Data Engineering*, 2(3):353–363, 1990.

[70] K.R. Popper. Conjectures and Refutations: The Growth of Scientific Knowledge, 1963.

[71] G. Onsongo, H. Xie, T.J. Griffin, and J. Carlis. Generating GO Slim Using Relational Database Management Systems to Support Proteomics Analysis. In *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on*, pages 215–217, 2008.

[72] P.L. Ross, Y.N. Huang, J.N. Marchese, B. Williamson, K. Parker, S. Hattan, N. Khainovski, S. Pillai, S. Dey, S. Daniels, et al. Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. *Molecular & Cellular Proteomics*, 3(12):1154, 2004.

[73] A. Thompson, J. Sch
"afer, K. Kuhn, S. Kienle, J. Schwarz, G. Schmidt, T. Neumann, and C. Hamon. Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. *Anal. Chem*, 75(8):1895–1904, 2003.

[74] K. Aggarwal, L.H. Choe, and K.H. Lee. Shotgun proteomics using the iTRAQ isobaric tags. *Briefings in Functional Genomics*, 5(2):112, 2006.

[75] S. Wiese, K.A. Reidegeld, H.E. Meyer, and B. Warscheid. Protein labeling by iTRAQ: A new tool for quantitative mass spectrometry in proteome research. *Proteomics*, 7(3):340–350, 2007.

[76] A.D. McNaught, A. Wilkinson, International Union of Pure, and Applied Chemistry. *Compendium of chemical terminology: IUPAC recommendations*. Blackwell Science, 1997.

[77] Scripps_Center. Ion trapś limitations: Precursor ion scanning, one-third rule and dynamic range. http://masspec.scripps.edu/mshistory/whatisms_details.php, 2010.

[78] M.M. Savitski, M. F
"alth, YM Fung, C.M. Adams, and R.A. Zubarev. Bifurcating Fragmentation Behavior of Gas-Phase Tryptic Peptide Dications in Collisional Activation. *Journal of the American Society for Mass Spectrometry*, 19(12):1755–1763, 2008.

[79] Matrix Science. Accuracy and resolution. http://www.matrixscience.com/help/mass_accuracy_help.html, 2010.

[80] E.J. Want, B.F. Cravatt, and G. Siuzdak. The expanding role of mass spectrometry in metabolite profiling and characterization. *Chembiochem*, 6(11):1941–1951, 2005.

[81] J.M. Armenta, I. Hoeschele, and I.M. Lazar. Differential protein expression analysis using stable isotope labeling and PQD linear ion trap MS technology. *Journal of the American Society for Mass Spectrometry*, 20(7):1287–1302, 2009.

[82] W.T. Lin, W.N. Hung, Y.H. Yian, K.P. Wu, C.L. Han, Y.R. Chen, Y.J. Chen, T.Y. Sung, and W.L. Hsu. Multi-Q: a fully automated tool for multiplexed protein quantitation. *J. Proteome Res*, 5(9):2328–2338, 2006.

[83] I.P. Shadforth, T.P.J. Dunkley, K.S. Lilley, and C. Bessant. i-Tracker: For quantitative proteomics using iTRAQ. *Bmc Genomics*, 6(1):145, 2005.

[84] M.J. MacCoss, C.C. Wu, H. Liu, R. Sadygov, and J.R. Yates III. A correlation algorithm for the automated quantitative analysis of shotgun proteomics data. *Anal. Chem*, 75(24):6912–6921, 2003.

[85] P. Pedrioli, A. Keller, and N. King. Libra (last accessed date nov 9 2010). http://sashimi.svn.sourceforge.net/viewvc/sashimi/trunk/trans_proteomic_pipeline/src/Quantitatio
2010.

[86] W.X. Schulze and M. Mann. A novel proteomic screen for peptide-protein interactions. *Journal of Biological Chemistry*, 279(11):10756, 2004.

[87] V.P. Andreev, L. Li, T. Rejtar, Q. Li, J.G. Ferry, and B.L. Karger. New Algorithm for 15N/14N Quantitation with LC- ESI- MS Using an LTQ-FT Mass Spectrometer. *J. Proteome Res*, 5(8):2039–2045, 2006.

[88] N.A. Karp, W. Huber, P.G. Sadowski, P.D. Charles, S.V. Hester, and K.S. Lilley. Addressing accuracy and precision issues in iTRAQ quantitation. *Molecular & Cellular Proteomics*, 9(9):1885, 2010.

[89] Applied Biosystems. Protein pilot version 3.0 online help manual, 2009.

[90] B. Carrillo, C. Yanofsky, S. Laboissiere, R. Nadon, and R.E. Kearney. Methods for combining peptide intensities to estimate relative protein abundance. *Bioinformatics*, 26(1):98, 2010.

[91] T.C. Lund, L.B. Anderson, V. McCullar, L.A. Higgins, G.H. Yun, B. Grzywacz, M.R. Verneris, and J.S. Miller. iTRAQ is a useful method to screen for membrane-bound proteins differentially expressed in human natural killer cell types. *J. Proteome Res*, 6(2):644–653, 2007.

[92] S.K. Akkina, Y. Zhang, G.L. Nelsestuen, W.S. Oetting, and H.N. Ibrahim. Temporal stability of the urinary proteome after kidney transplant: more sensitive than protein composition? *J. Proteome Res*, 8(1):94–103, 2009.

[93] I.V. Shilov, S.L. Seymour, A.A. Patel, A. Loboda, W.H. Tang, S.P. Keating, C.L. Hunter, L.M. Nuwaysir, and D.A. Schaeffer. The Paragon Algorithm, a next generation search engine that uses sequence temperature values and feature probabilities to identify peptides from tandem mass spectra. *Molecular & Cellular Proteomics*, 6(9):1638, 2007.

[94] J. Sui, J. Zhang, T.L. Tan, C.B. Ching, and W.N. Chen. Comparative proteomics analysis of vascular smooth muscle cells incubated with S-and R-enantiomers of atenolol using iTRAQ-coupled two-dimensional LC-MS/MS. *Molecular & Cellular Proteomics*, 7(6):1007, 2008.

[95] Y. Ishihama, J. Rappsilber, and M. Mann. Modular stop and go extraction tips with stacked disks for parallel and multidimensional peptide fractionation in proteomics. *J. Proteome Res*, 5(4):988–994, 2006.

[96] D.N. Perkins, D.J.C. Pappin, D.M. Creasy, and J.S. Cottrell. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis*, 20(18):3551–3567, 1999.

[97] Wikipedia. Weighted mean (last accessed date january 7 2011). http://en.wikipedia.org/wiki/Weighted_mean, 2011.

[98] MB Wilk and R. Gnanadesikan. Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1, 1968.

[99] R. Khattree and C.R. Rao. *Statistics in industry*. North-Holland, 2003.

[100] S.S. Shapiro and M.B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591, 1965.

[101] D. Soh, D. Dong, Y. Guo, and L. Wong. Enabling more sophisticated gene expression analysis for understanding diseases and optimizing treatments, 2007.

[102] J.V. Carlis and S. Krieger. *Mastering Relational Database Querying and Analysis*. DRAFT, 2008.

[103] TS Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, et al. Human Protein Reference Database–2009 update. *Nucleic acids research*, 2008.

[104] Imre Vastrik, Peter D'Eustachio, Esther Schmidt, Gopal Gopinath, David Croft, Bernard de Bono, Marc Gillespie, Bijay Jassal, Suzanna Lewis, Lisa Matthews, Guanming Wu, Ewan Birney, and Lincoln Stein. Reactome: a knowledge base of biologic pathways and processes. *Genome Biol*, 8(3):R39, 2007.

[105] S. Dar and R. Agrawal. Extending SQL with generalized transitive closure. *IEEE Transactions on Knowledge and Data Engineering*, 5(5):799–812, 1993.

[106] C. Ordonez. Optimizing recursive queries in SQL. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 834–839. ACM New York, NY, USA, 2005.

[107] P. Valduriez and H. Boral. Evaluation of Recursive Queries Using Join Indices. In *First International Conference on Expert Database Systems*, pages 271–293, 1986.

[108] Keydata. Sql rank. http://www.1keydata.com/sql/sql-rank.html, 2010.

[109] K. Bruso. The development of bucketing operators and a supporting operator framework for relational database management systems, 2009.

[110] Jianzhen Xu and Yongjin Li. Discovering disease-genes by topological features in human protein-protein interaction network. *Bioinformatics*, 22(22):2800–5, 2006.

[111] Z. Tu, L. Wang, M. Xu, X. Zhou, T. Chen, and F. Sun. Further understanding human disease genes by comparing with housekeeping genes and other genes. *BMC genomics*, 7(1):31, 2006.

[112] L.C. Tranchevent, R. Barriot, S. Yu, S. Van Vooren, P. Van Loo, B. Coessens, B. De Moor, S. Aerts, and Y. Moreau. ENDEAVOUR update: a web resource for gene prioritization in multiple species. *Nucleic Acids Research*, 36(Web Server issue):W377, 2008.

[113] M Ashburner, CA Ball, JA Blake, D Botstein, H Butler, JM Cherry, AP Davis, K Dolinski, SS Dwight, JT Eppig, MA Harris, DP Hill, L Issel-Tarver, A Kasarskis, S Lewis, JC Matese, JE Richardson, M Ringwald, GM Rubin, and G Sherlock. Gene ontology: tool for the unification of biology. the Gene Ontology Consortium. *Nat Genet*, 25(1):25–29, 2000.

[114] Berkeley Bioinformatics and Ontologies Project. Obo-edit, 2007.
http://oboedit.org/author.html.

[115] F.M. McCarthy, N. Wang, G.B. Magee, B. Nanduri, M.L. Lawrence, E.B. Camon, D.G. Barrell, D.P. Hill, M.E. Dolan, W.P. Williams, et al. AgBase: a functional genomics resource for agriculture. *BMC Genomics*, 7(1):229, 2006.

[116] SGD . Sgd gene ontology slim mapper, 2007.
http://db.yeastgenome.org/cgi-bin/GO/goTermMapper.pl.

[117] GO Consortium. Go slim and subset guide, 2007.
http://www.geneontology.org/GO.slims.shtml.

[118] Candida Genome Database. Cgd gene ontology slim mapper, 2007. http://www.candidagenome.org/cgi-bin/GO/goTermMapper.

[119] N.L. Anderson and N.G. Anderson. The human plasma proteome: history, character, and diagnostic prospects. *Molecular & cellular proteomics: MCP*, 1(11):845, 2002.

[120] N.L. Anderson, M. Polanski, R. Pieper, T. Gatlin, R.S. Tirumalai, T.P. Conrads, T.D. Veenstra, J.N. Adkins, J.G. Pounds, R. Fagan, et al. The human plasma proteome: a nonredundant list developed by combination of four separate sources. *Molecular & Cellular Proteomics*, 3(4):311, 2004.

[121] Kinemed. Kinemed (last accessed date feb 03 2011). http://www.kinemed.com/glossary, 2011.

[122] L. Krishnamurthy, J. Nadeau, G. Ozsoyoglu, M. Ozsoyoglu, G. Schaeffer, M. Tasan, and W. Xu. Pathways database system: an integrated system for biological pathways. *Bioinformatics*, 19(8):930, 2003.

[123] E.E. Schadt. Molecular networks as sensors and drivers of common human diseases. *Nature*, 461(7261):218–223, 2009.

[124] Y. Chen, J. Zhu, P.Y. Lum, X. Yang, S. Pinto, D.J. MacNeil, C. Zhang, J. Lamb, S. Edwards, S.K. Sieberts, et al. Variations in DNA elucidate molecular networks that cause disease. *Nature*, 452(7186):429–435, 2008.

[125] Maricel G Kann. Protein interactions and disease: computational approaches to uncover the etiology of diseases. *Brief Bioinform*, 8(5):333–46, 2007.

[126] Judith S Sebolt-Leopold and Roman Herrera. Targeting the mitogen-activated protein kinase cascade to treat cancer. *Nat Rev Cancer*, 4(12):937–47, 2004.

[127] Andrey A Ptitsyn, Michael M Weil, and Douglas H Thamm. Systems biology approach to identification of biomarkers for metastatic progression in cancer. *BMC Bioinformatics*, 9 Suppl 9(NIL):S8, 2008.

[128] The Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. *Genome Res*, 11(8):1425–33, 2001.

[129] O. Odibat, C.K. Reddy, and C.N. Giroux. Differential biclustering for gene expression analysis. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 275–284. ACM, 2010.

[130] Xuebing Wu, Rui Jiang, Michael Q Zhang, and Shao Li. Network-based global inference of human disease genes. *Mol Syst Biol*, 4(NIL):189, 2008.

[131] Richard A George, Jason Y Liu, Lina L Feng, Robert J Bryson-Richardson, Diane Fatkin, and Merridee A Wouters. Analysis of protein sequence and interaction data for candidate disease gene prediction. *Nucleic Acids Res*, 34(19):e130, 2006.

[132] Trey Ideker and Roded Sharan. Protein networks in disease. *Genome Res*, 18(4):644–52, 2008.

[133] BA Eckman and PG Brown. Graph data management for molecular and cell biology. *IBM Journal of Research and Development*, 50(6):560, 2006.

[134] M.Y. Eltabakh, M. Ouzzani, W.G. Aref, A.K. Elmagarmid, Y. Laura-Silva, M.U. Arshad, D. Salt, and I. Baxter. Managing biological data using bdbms. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 1600–1603. Citeseer, 2008.

[135] F. Olken. Graph data management for molecular biology. *OMICS A Journal of Integrative Biology*, 7(1):75–78, 2003.

[136] Getiria Onsongo, Hongwei Xie, Timothy J. Griffin, and John Carlis. Generating go slim using relational database management systems to support proteomics analysis. In *Proceedings of the 2008 21st IEEE International Symposium on Computer-Based Medical Systems*, pages 215–217, Washington, DC, USA, 2008. IEEE Computer Society.

[137] S. Tata, W. Lang, and J.M. Patel. Periscope/SQ: interactive exploration of biological sequence databases. In *Proceedings of the 33rd international conference on Very large data bases*, pages 1406–1409. VLDB Endowment, 2007.

[138] P. Bouros, S. Skiadopoulos, T. Dalamagas, D. Sacharidis, and T. Sellis. Evaluating reachability queries over path collections. In *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*, page 416. Springer, 2009.

[139] T. Andreasen, H. Bulskov, and R. Knappe. On ontology-based querying. In *Flexible query answering systems: recent advances: proceedings of the Fourth International Conference on Flexible Query Answering Systems, FQAS'2000, October 25-28, 2000, Warsaw, Poland*, page 15. Physica Verlag, 2000.

[140] S. Tata and J.M. Patel. PiQA: An algebra for querying protein data sets. In *Proc. of 15th SSDBM Conf.* Citeseer, 2003.

[141] U. Leser. A query language for biological networks. *BIOINFORMATICS-OXFORD-*, 21(2), 2005.

[142] G. Bebek and J. Yang. PathFinder: mining signal transduction pathway segments from protein-protein interaction networks. *BMC bioinformatics*, 8(1):335, 2007.

[143] K. Clement, N. Gustafson, A. Berbert, H. Carroll, C. Merris, A. Olsen, M. Clement, Q. Snell, J. Allen, and R.J. Roper. PathGen: A Transitive Gene Pathway Generator. *Bioinformatics*, 2009.

[144] Peng Qu, Jennifer Roberts, Yuan Li, Marjorie Albrecht, Oscar W Cummings, John N Eble, Hong Du, and Cong Yan. Stat3 downstream genes serve as biomarkers in human lung carcinomas and chronic obstructive pulmonary disease. *Lung Cancer*, 63(3):341–7, 2009.

[145] R Mehrian-Shai, C D Chen, T Shi, S Horvath, S F Nelson, J K V Reichardt, and C L Sawyers. Insulin growth factor-binding protein 2 is a candidate biomarker for PTEN status and PI3K/Akt pathway activation in glioblastoma and prostate cancer. *Proc Natl Acad Sci U S A*, 104(13):5563–8, 2007.

[146] Lao H Saal, Peter Johansson, Karolina Holm, Sofia K Gruvberger-Saal, Qing-Bai She, Matthew Maurer, Susan Koujak, Adolfo A Ferrando, Per Malmstrom, Lorenzo Memeo, Jorma Isola, Par-Ola Bendahl, Neal Rosen, Hanina Hibshoosh,

Markus Ringner, Ake Borg, and Ramon Parsons. Poor prognosis in carcinoma is associated with a gene expression signature of aberrant PTEN tumor suppressor pathway activity. *Proc Natl Acad Sci U S A*, 104(18):7564–9, 2007.

[147] Andrea H Bild, Guang Yao, Jeffrey T Chang, Quanli Wang, Anil Potti, Dawn Chasse, Mary-Beth Joshi, David Harpole, Johnathan M Lancaster, Andrew Berchuck, John A Jr Olson, Jeffrey R Marks, Holly K Dressman, Mike West, and Joseph R Nevins. Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*, 439(7074):353–7, 2006.

[148] I Xenarios, D W Rice, L Salwinski, M K Baron, E M Marcotte, and D Eisenberg. DIP: the database of interacting proteins. *Nucleic Acids Res*, 28(1):289–91, 2000.

[149] A.D. Hanson, A. Pribat, J.C. Waller, and V. de Crécy-lagard. Unknownproteins and orphanenzymes: the missing half of the engineering parts list–and how to find it. *The Biochemical journal*, 425(1):1, 2010.

[150] C. Roney and V. Dana. Disclosing ambiguous gene aliases by automatic literature profiling. *BMC Genomics*, 11, 2010.

[151] F.M. Selaru, Y. Xu, J. Yin, T. Zou, T.C. Liu, Y. Mori, J.M. Abraham, F. Sato, S. Wang, C. Twigg, et al. Artificial neural networks distinguish among subtypes of neoplastic colorectal lesions**. *Gastroenterology*, 122(3):606–613, 2002.

[152] E.F. Petricoin III, A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, et al. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359(9306):572–577, 2002.

[153] Y. Qu, B.L. Adam, Y. Yasui, M.D. Ward, L.H. Cazares, P.F. Schellhammer, Z. Feng, O.J. Semmes, and G.L. Wright Jr. Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients. *Clinical Chemistry*, 48(10):1835, 2002.

[154] E. Huang, S.H. Cheng, H. Dressman, J. Pittman, M.H. Tsou, C.F. Horng, A. Bild, E.S. Iversen, M. Liao, C.M. Chen, et al. Gene expression predictors of breast cancer outcomes. *The Lancet*, 361(9369):1590–1596, 2003.

[155] I. Fabregat. Dysregulation of apoptosis in hepatocellular carcinoma cells. *World journal of gastroenterology: WJG*, 15(5):513, 2009.

[156] T. Yamada and P. Bork. Evolution of biomolecular networks - lessons from metabolic and protein interactions. *Nature Reviews Molecular Cell Biology*, 10(11):791–803, 2009.

[157] N.H. Barton and P.D. Keightley. Understanding quantitative genetic variation. *Nature Reviews Genetics*, 3(1):11–21, 2002.

[158] Wikipedia. Wikipedia (last accessed date april 09 2011). http://en.wikipedia.org/wiki/Graph*_mathematics*, 2011.

[159] RM Christley, GL Pinchbeck, RG Bowers, D. Clancy, NP French, R. Bennett, and J. Turner. Infection in social networks: using network analysis to identify high-risk individuals. *American journal of epidemiology*, 162(10):1024, 2005.

[160] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 217–224. ACM, 2002.

[161] HV Jagadish, R. Agrawal, and L. Ness. A study of transitive closure as a recursion mechanism. *ACM SIGMOD Record*, 16(3):331–344, 1987.

[162] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21(suppl 1):i302, 2005.

[163] R.E. Gomory and T.C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[164] K.P. Jayapal, S. Sui, R.J. Philp, Y.J. Kok, M.G.S. Yap, T.J. Griffin, and W.S. Hu. Multitagging Proteomic Strategy to Estimate Protein Turnover Rates in Dynamic Systems. *Journal of Proteome Research*, 9(5):2087–2097, 2010.

[165] E.G. Hill, J.H. Schwacke, S. Comte-Walters, E.H. Slate, A.L. Oberg, J.E. Eckel-Passow, T.M. Therneau, and K.L. Schey. A statistical model for iTRAQ data analysis. *Journal of proteome research*, 7(8):3091–3101, 2008.

[166] R. Simon. Lost in translation: problems and pitfalls in translating laboratory observations to clinical utility. *European Journal of Cancer*, 44(18):2707–2713, 2008.

[167] A. Dupuy and R.M. Simon. Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. *Journal of the National Cancer Institute*, 99(2):147, 2007.

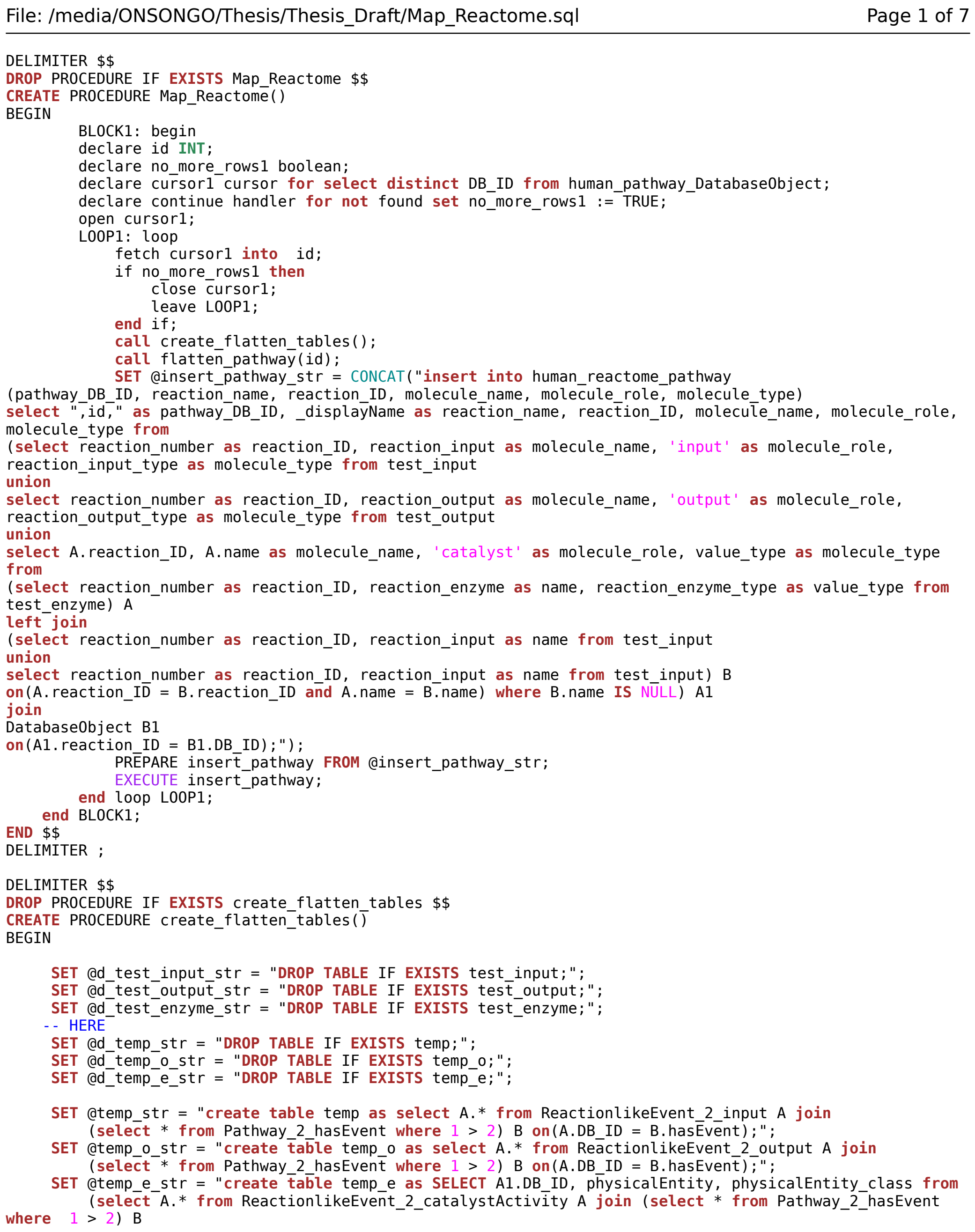
[168] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000.

Appendix A

Mapping Reactome

Algorithm A.1 Map Reactome()

```
STEP 1: Get ID for pathways from the relation DatabaseObject
STEP 2: Get reactions belonging to each pathway
STEP 3: Decompose complexes in reaction to constituent gene products
STEP 4: Update reactions to consist of gene products
```

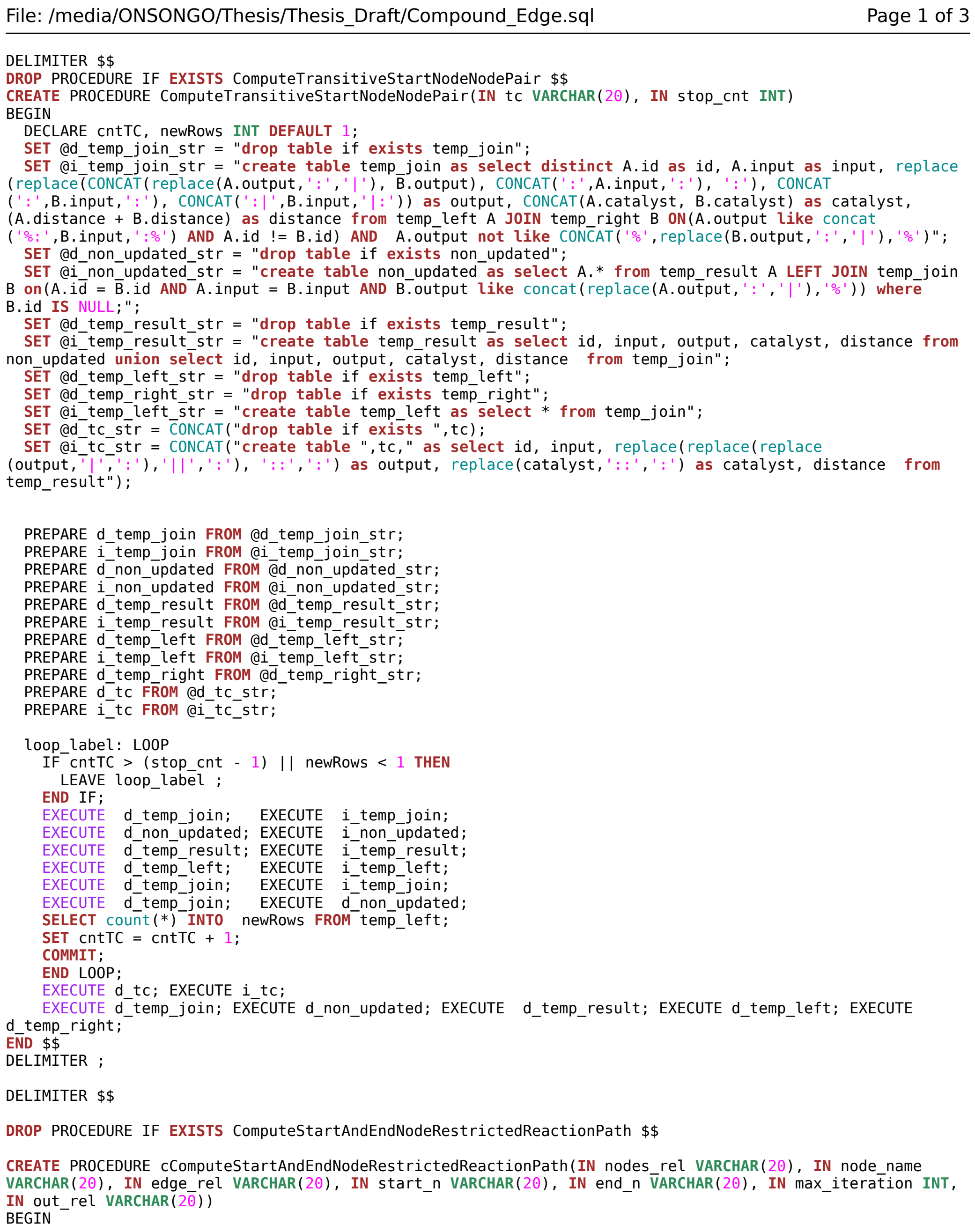


```
DELIMITER $$
DROP PROCEDURE IF EXISTS Map_Reactome $$
CREATE PROCEDURE Map_Reactome()
BEGIN
        BLOCK1: begin
        declare id INT;
        declare no_more_rows1 boolean;
        declare cursor1 cursor for select distinct DB_ID from human_pathway_DatabaseObject;
        declare continue handler for not found set no_more_rows1 := TRUE;
        open cursor1;
        LOOP1: loop
            fetch cursor1 into  id;
            if no_more_rows1 then
                close cursor1;
                leave LOOP1;
            end if;
            call create_flatten_tables();
            call flatten_pathway(id);
            SET @insert_pathway_str = CONCAT("insert into human_reactome_pathway
(pathway_DB_ID, reaction_name, reaction_ID, molecule_name, molecule_role, molecule_type)
select ",id," as pathway_DB_ID, _displayName as reaction_name, reaction_ID, molecule_name, molecule_role,
molecule_type from
(select reaction_number as reaction_ID, reaction_input as molecule_name, 'input' as molecule_role,
reaction_input_type as molecule_type from test_input
union
select reaction_number as reaction_ID, reaction_output as molecule_name, 'output' as molecule_role,
reaction_output_type as molecule_type from test_output
union
select A.reaction_ID, A.name as molecule_name, 'catalyst' as molecule_role, value_type as molecule_type
from
(select reaction_number as reaction_ID, reaction_enzyme as name, reaction_enzyme_type as value_type from
test_enzyme) A
left join
(select reaction_number as reaction_ID, reaction_input as name from test_input
union
select reaction_number as reaction_ID, reaction_input as name from test_input) B
on(A.reaction_ID = B.reaction_ID and A.name = B.name) where B.name IS NULL) A1
join
DatabaseObject B1
on(A1.reaction_ID = B1.DB_ID);");
            PREPARE insert_pathway FROM @insert_pathway_str;
            EXECUTE insert_pathway;
        end loop LOOP1;
    end BLOCK1;
END $$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS create_flatten_tables $$
CREATE PROCEDURE create_flatten_tables()
BEGIN

     SET @d_test_input_str = "DROP TABLE IF EXISTS test_input;";
     SET @d_test_output_str = "DROP TABLE IF EXISTS test_output;";
     SET @d_test_enzyme_str = "DROP TABLE IF EXISTS test_enzyme;";
    -- HERE
     SET @d_temp_str = "DROP TABLE IF EXISTS temp;";
     SET @d_temp_o_str = "DROP TABLE IF EXISTS temp_o;";
     SET @d_temp_e_str = "DROP TABLE IF EXISTS temp_e;";

     SET @temp_str = "create table temp as select A.* from ReactionlikeEvent_2_input A join
         (select * from Pathway_2_hasEvent where 1 > 2) B on(A.DB_ID = B.hasEvent);";
     SET @temp_o_str = "create table temp_o as select A.* from ReactionlikeEvent_2_output A join
         (select * from Pathway_2_hasEvent where 1 > 2) B on(A.DB_ID = B.hasEvent);";
     SET @temp_e_str = "create table temp_e as SELECT A1.DB_ID, physicalEntity, physicalEntity_class from
         (select A.* from ReactionlikeEvent_2_catalystActivity A join (select * from Pathway_2_hasEvent
where  1 > 2) B
```

```
         on(A.DB_ID = B.hasEvent)) A1 join CatalystActivity B1 on(A1.catalystActivity = B1.DB_ID);";

     PREPARE d_test_input FROM @d_test_input_str;
     PREPARE d_test_output FROM @d_test_output_str;
     PREPARE d_test_enzyme FROM @d_test_enzyme_str;
     PREPARE d_temp FROM @d_temp_str;
     PREPARE d_temp_o FROM @d_temp_o_str;
     PREPARE d_temp_e FROM @d_temp_e_str;

     PREPARE temp FROM @temp_str;
     PREPARE temp_o FROM @temp_o_str;
     PREPARE temp_e FROM @temp_e_str;

     EXECUTE d_test_input;
     EXECUTE d_test_output;
     EXECUTE d_test_enzyme;
     EXECUTE d_temp;
     EXECUTE d_temp_o;
     EXECUTE d_temp_e;
     EXECUTE temp;
     EXECUTE temp_o;
     EXECUTE temp_e;

END $$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS flatten_pathway $$
CREATE PROCEDURE flatten_pathway(IN pathway_ID INT)
BEGIN
    SET @d_temp_str = "drop table if exists temp;";
    SET @temp_str = CONCAT("create table temp as select A.* from ReactionlikeEvent_2_input A join
                     (select * from Pathway_2_hasEvent where DB_ID = ",pathway_ID," and hasEvent_class =
'Reaction') B
                     on(A.DB_ID = B.hasEvent);");
    SET @reaction_input_str = "create table reaction_input as select DISTINCT DB_ID, input, input_class
from temp where
                     input_class = 'EntityWithAccessionedSequence';";
    SET @reaction_input_simpleEntity_str = "create table reaction_input_simpleEntity as select DISTINCT
DB_ID, input,
                     input_class from temp where input_class = 'SimpleEntity';";
    SET @temp1_str = "create table temp1 as select DISTINCT DB_ID, input, input_class from temp where 1 >
2;";
    SET @temp2_str = "create table temp2 as select DISTINCT DB_ID, input, input_class from temp where 1 >
2;";
    SET @c_test_input_str = "create table test_input(reaction_number INT, reaction_input TEXT,
reaction_input_type VARCHAR(32));";

    SET @d_temp_o_str = "drop table if exists temp_o;";
    SET @temp_o_str = CONCAT("create table temp_o as select A.* from ReactionlikeEvent_2_output A join
                       (select * from Pathway_2_hasEvent where DB_ID = ",pathway_ID," and hasEvent_class
= 'Reaction') B
                       on(A.DB_ID = B.hasEvent);");
    SET @reaction_output_str = "create table reaction_output as select DISTINCT DB_ID, output,
output_class from temp_o where
                       output_class = 'EntityWithAccessionedSequence';";
    SET @reaction_output_simpleEntity_str = "create table reaction_output_simpleEntity as select DISTINCT
DB_ID, output,
                     output_class from temp_o where output_class = 'SimpleEntity';";
    SET @temp1_o_str = "create table temp1_o as select DISTINCT DB_ID, output, output_class from temp_o
where 1 > 2;";
    SET @temp2_o_str = "create table temp2_o as select DISTINCT DB_ID, output, output_class from temp_o
where 1 > 2;";
    SET @c_test_output_str = "create table test_output(reaction_number INT, reaction_output TEXT,
reaction_output_type VARCHAR(32));";

    SET @d_temp_e_str = "drop table if exists temp_e;";
    SET @temp_e_str = CONCAT("create table temp_e as SELECT A1.DB_ID, physicalEntity,
physicalEntity_class from
                      (select A.* from ReactionlikeEvent_2_catalystActivity A join
                      (select * from Pathway_2_hasEvent where DB_ID = ",pathway_ID," and hasEvent_class =
'Reaction') B
                      on(A.DB_ID = B.hasEvent)) A1 join CatalystActivity B1 on(A1.catalystActivity =
B1.DB_ID);");
    SET @reaction_enzyme_str = "create table reaction_enzyme as select DISTINCT DB_ID, physicalEntity,
physicalEntity_class from temp_e where
                      physicalEntity_class = 'EntityWithAccessionedSequence';";
    SET @reaction_physicalEntity_simpleEntity_str = "create table reaction_physicalEntity_simpleEntity as
select DISTINCT DB_ID, physicalEntity,
                      physicalEntity_class from temp_e where physicalEntity_class = 'SimpleEntity';";
    SET @temp1_e_str = "create table temp1_e as select DISTINCT DB_ID, physicalEntity,
physicalEntity_class from temp_e where 1 > 2;";
    SET @temp2_e_str = "create table temp2_e as select DISTINCT DB_ID, physicalEntity,
physicalEntity_class from temp_e where 1 > 2;";
    SET @c_test_enzyme_str = "create table test_enzyme(reaction_number INT, reaction_enzyme TEXT,
reaction_enzyme_type VARCHAR(32));";

     PREPARE d_temp FROM @d_temp_str;
     PREPARE temp FROM @temp_str;
     PREPARE reaction_input FROM @reaction_input_str;
     PREPARE reaction_input_simpleEntity FROM @reaction_input_simpleEntity_str;
     PREPARE temp1 FROM @temp1_str;
     PREPARE temp2 FROM @temp2_str;
     PREPARE c_test_input FROM @c_test_input_str;

     PREPARE d_temp_o FROM @d_temp_o_str;
     PREPARE temp_o FROM @temp_o_str;
     PREPARE reaction_output FROM @reaction_output_str;
     PREPARE reaction_output_simpleEntity FROM @reaction_output_simpleEntity_str;
     PREPARE temp1_o FROM @temp1_o_str;
     PREPARE temp2_o FROM @temp2_o_str;
     PREPARE c_test_output FROM @c_test_output_str;

     PREPARE d_temp_e FROM @d_temp_e_str;
     PREPARE temp_e FROM @temp_e_str;
     PREPARE reaction_enzyme FROM @reaction_enzyme_str;
     PREPARE reaction_physicalEntity_simpleEntity FROM @reaction_physicalEntity_simpleEntity_str;
     PREPARE temp1_e FROM @temp1_e_str;
     PREPARE temp2_e FROM @temp2_e_str;
     PREPARE c_test_enzyme FROM @c_test_enzyme_str;
-- INPUT
     EXECUTE d_temp;
     EXECUTE temp;
     EXECUTE reaction_input;
     EXECUTE reaction_input_simpleEntity;
     EXECUTE temp1;
     EXECUTE temp2;
     EXECUTE c_test_input;
     CALL flatten_input('test_input');
-- OUTPUT
     EXECUTE d_temp_o;
     EXECUTE temp_o;
     EXECUTE reaction_output;
     EXECUTE reaction_output_simpleEntity;
     EXECUTE temp1_o;
     EXECUTE temp2_o;
     EXECUTE c_test_output;
     CALL flatten_output('test_output');
-- ENZYME
     EXECUTE d_temp_e;
     EXECUTE temp_e;
     EXECUTE reaction_enzyme;
     EXECUTE  reaction_physicalEntity_simpleEntity;
     EXECUTE temp1_e;
     EXECUTE temp2_e;
     EXECUTE c_test_enzyme;
     CALL flatten_enzyme('test_enzyme');

END $$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS flatten_enzyme $$
CREATE PROCEDURE flatten_enzyme(IN result_rel_name VARCHAR(32))
BEGIN
     DECLARE cntTC INT DEFAULT 2;
     SET @d_temp_str = "DROP TABLE temp_e;";
     SET @d_temp1_str = "drop table temp1_e;";
     SET @temp1_str = "create table temp1_e as select DISTINCT A.DB_ID as DB_ID, hasComponent as
physicalEntity, hasComponent_class as physicalEntity_class from (select DB_ID, physicalEntity,
physicalEntity_class from temp_e where physicalEntity_class = 'Complex') A join Complex_2_hasComponent B
ON(A.physicalEntity = B.DB_ID);";
     SET @d_temp2_str = "drop table temp2_e;";
     SET @temp2_str = "create table temp2_e as select DISTINCT A.DB_ID as DB_ID, hasCandidate as
physicalEntity, hasCandidate_class as physicalEntity_class from (select DB_ID, physicalEntity,
physicalEntity_class from temp_e where physicalEntity_class = 'CandidateSet') A join
CandidateSet_2_hasCandidate B ON(A.physicalEntity = B.DB_ID);";
     SET @in_temp_str = "create table temp_e as select * from temp1_e union select * from temp2_e;";
     SET @in_out_str_1 = "insert into reaction_enzyme (DB_ID, physicalEntity, physicalEntity_class)
select DISTINCT DB_ID, physicalEntity, physicalEntity_class from temp_e where physicalEntity_class =
'EntityWithAccessionedSequence';";
     SET @in_out_str_2 = "insert into reaction_physicalEntity_simpleEntity (DB_ID, physicalEntity,
physicalEntity_class) select DISTINCT DB_ID, physicalEntity, physicalEntity_class from temp_e where
physicalEntity_class = 'SimpleEntity';";
     SET @d_reaction_enzyme_str = "drop table reaction_enzyme;";
     SET @result_str_1 = CONCAT("insert into ",result_rel_name," (reaction_number, reaction_enzyme,
reaction_enzyme_type) SELECT A1.DB_ID as reaction_number, name as reaction_enzyme, 'gene' as
reaction_enzyme_type FROM (select A.DB_ID, referenceEntity  from reaction_enzyme A join
EntityWithAccessionedSequence B ON(A.physicalEntity = B.DB_ID)) A1 JOIN  (select * from
ReferenceEntity_2_name where name_rank = 0) B1 ON(A1.referenceEntity = B1.DB_ID);");
     SET @d_reaction_physicalEntity_simpleEntity_str = "drop table reaction_physicalEntity_simpleEntity;";
     SET @result_str_2 = CONCAT("insert into ",result_rel_name," (reaction_number, reaction_enzyme,
reaction_enzyme_type) SELECT reaction_number, name as reaction_enzyme, 'simpleMolecule' as
reaction_enzyme_type from (SELECT A.DB_ID as reaction_number, referenceEntity FROM
reaction_physicalEntity_simpleEntity A JOIN SimpleEntity_2_referenceEntity B ON(A.physicalEntity =
B.DB_ID)) A1 JOIN (select * from ReferenceEntity_2_name where name_rank = 0) B1 ON(A1.referenceEntity =
B1.DB_ID);");

     PREPARE d_temp FROM @d_temp_str;
     PREPARE d_temp1 FROM @d_temp1_str;
     PREPARE temp1 FROM @temp1_str;
     PREPARE d_temp2 FROM @d_temp2_str;
     PREPARE temp2 FROM @temp2_str;
     PREPARE in_temp FROM @in_temp_str;
     PREPARE in_out_1 FROM @in_out_str_1;
     PREPARE in_out_2 FROM @in_out_str_2;
     PREPARE d_reaction_enzyme FROM @d_reaction_enzyme_str;
     PREPARE result_1 FROM @result_str_1;
     PREPARE d_reaction_physicalEntity_simpleEntity FROM @d_reaction_physicalEntity_simpleEntity_str;
     PREPARE result_2 FROM @result_str_2;

    loop_label: LOOP
       IF cntTC < 1  THEN
         LEAVE loop_label;
       END IF;
       EXECUTE d_temp1;
       EXECUTE temp1;
       EXECUTE d_temp2;
       EXECUTE temp2;
       EXECUTE d_temp;
       EXECUTE in_temp;
       EXECUTE in_out_1;
       EXECUTE in_out_2;
       SELECT count(*) INTO  cntTC FROM temp_e;
   END LOOP;
   EXECUTE result_1;
   EXECUTE result_2;
   EXECUTE d_temp; EXECUTE d_temp1; EXECUTE d_temp2; EXECUTE d_reaction_enzyme; EXECUTE
d_reaction_physicalEntity_simpleEntity;
END $$
DELIMITER ;

DELIMITER $$
DROP PROCEDURE IF EXISTS flatten_input $$
CREATE PROCEDURE flatten_input(IN result_rel_name VARCHAR(32))
BEGIN
     DECLARE cntTC INT DEFAULT 2;
     SET @d_temp_str = "DROP TABLE temp;";
     SET @d_temp1_str = "drop table temp1;";
     SET @temp1_str = "create table temp1 as select DISTINCT A.DB_ID as DB_ID, hasComponent as input,
hasComponent_class as input_class from (select DB_ID, input, input_class from temp where input_class =
'Complex') A join Complex_2_hasComponent B ON(A.input = B.DB_ID);";
     SET @d_temp2_str = "drop table temp2;";
     SET @temp2_str = "create table temp2 as select DISTINCT A.DB_ID as DB_ID, hasCandidate as input,
hasCandidate_class as input_class from (select DB_ID, input, input_class from temp where input_class =
'CandidateSet') A join CandidateSet_2_hasCandidate B ON(A.input = B.DB_ID);";
     SET @in_temp_str = "create table temp as select * from temp1 union select * from temp2;";
     SET @in_out_str_1 = "insert into reaction_input (DB_ID, input, input_class) select DISTINCT DB_ID,
input, input_class from temp where input_class = 'EntityWithAccessionedSequence';";
     SET @in_out_str_2 = "insert into reaction_input_simpleEntity (DB_ID, input, input_class) select
DISTINCT DB_ID, input, input_class from temp where input_class = 'SimpleEntity';";
     SET @d_reaction_input_str = "drop table reaction_input;";
     SET @result_str_1 = CONCAT("insert into ",result_rel_name," (reaction_number, reaction_input,
reaction_input_type) SELECT A1.DB_ID as reaction_number, name as reaction_input, 'gene' as
reaction_input_type FROM (select A.DB_ID, referenceEntity  from reaction_input A join
EntityWithAccessionedSequence B ON(A.input = B.DB_ID)) A1 JOIN  (select * from ReferenceEntity_2_name
where name_rank = 0) B1 ON(A1.referenceEntity = B1.DB_ID);");
     SET @d_reaction_input_simpleEntity_str = "drop table reaction_input_simpleEntity;";
     SET @result_str_2 = CONCAT("insert into ",result_rel_name," (reaction_number, reaction_input,
reaction_input_type) SELECT reaction_number, name as reaction_input, 'simpleMolecule' as
reaction_input_type from (SELECT A.DB_ID as reaction_number, referenceEntity FROM
reaction_input_simpleEntity A JOIN SimpleEntity_2_referenceEntity B ON(A.input = B.DB_ID)) A1 JOIN
(select * from ReferenceEntity_2_name where name_rank = 0) B1 ON(A1.referenceEntity = B1.DB_ID);");

     PREPARE d_temp FROM @d_temp_str;
     PREPARE d_temp1 FROM @d_temp1_str;
     PREPARE temp1 FROM @temp1_str;
     PREPARE d_temp2 FROM @d_temp2_str;
     PREPARE temp2 FROM @temp2_str;
     PREPARE in_temp FROM @in_temp_str;
     PREPARE in_out_1 FROM @in_out_str_1;
     PREPARE in_out_2 FROM @in_out_str_2;
     PREPARE d_reaction_input FROM @d_reaction_input_str;
     PREPARE result_1 FROM @result_str_1;
     PREPARE result_2 FROM @result_str_2;
     PREPARE d_reaction_input_simpleEntity FROM @d_reaction_input_simpleEntity_str;

    loop_label: LOOP
       IF cntTC < 1  THEN
         LEAVE loop_label;
       END IF;
       EXECUTE d_temp1;
       EXECUTE temp1;
       EXECUTE d_temp2;
       EXECUTE temp2;
       EXECUTE d_temp;
       EXECUTE in_temp;
       EXECUTE in_out_1;
       EXECUTE in_out_2;
       SELECT count(*) INTO  cntTC FROM temp;
   END LOOP;
   EXECUTE result_1;
   EXECUTE result_2;
   EXECUTE d_temp; EXECUTE d_temp1; EXECUTE d_temp2; EXECUTE d_reaction_input; EXECUTE
d_reaction_input_simpleEntity;
END $$
DELIMITER ;


DELIMITER $$
DROP PROCEDURE IF EXISTS flatten_output $$
CREATE PROCEDURE flatten_output(IN result_rel_name VARCHAR(32))
BEGIN
     DECLARE cntTC INT DEFAULT 2;
     SET @d_temp_str = "DROP TABLE temp_o;";
     SET @d_temp1_str = "drop table temp1_o;";
     SET @temp1_str = "create table temp1_o as select DISTINCT A.DB_ID as DB_ID, hasComponent as output,
hasComponent_class as output_class from (select DB_ID, output, output_class from temp_o where
output_class = 'Complex') A join Complex_2_hasComponent B ON(A.output = B.DB_ID);";
     SET @d_temp2_str = "drop table temp2_o;";
     SET @temp2_str = "create table temp2_o as select DISTINCT A.DB_ID as DB_ID, hasCandidate as output,
hasCandidate_class as output_class from (select DB_ID, output, output_class from temp_o where
output_class = 'CandidateSet') A join CandidateSet_2_hasCandidate B ON(A.output = B.DB_ID);";
     SET @in_temp_str = "create table temp_o as select * from temp1_o union select * from temp2_o;";
     SET @in_out_str_1 = "insert into reaction_output (DB_ID, output, output_class) select DISTINCT
DB_ID, output, output_class from temp_o where output_class = 'EntityWithAccessionedSequence';";
     SET @in_out_str_2 = "insert into reaction_output_simpleEntity (DB_ID, output, output_class) select
DISTINCT DB_ID, output, output_class from temp_o where output_class = 'SimpleEntity';";
     SET @d_reaction_output_str = "drop table reaction_output;";
     SET @result_str_1 = CONCAT("insert into ",result_rel_name," (reaction_number, reaction_output,
reaction_output_type) SELECT A1.DB_ID as reaction_number, name as reaction_output, 'gene' as
reaction_output_type FROM (select A.DB_ID, referenceEntity  from reaction_output A join
EntityWithAccessionedSequence B ON(A.output = B.DB_ID)) A1 JOIN  (select * from ReferenceEntity_2_name
where name_rank = 0) B1 ON(A1.referenceEntity = B1.DB_ID);");
     SET @d_reaction_output_simpleEntity_str = "drop table reaction_output_simpleEntity;";
     SET @result_str_2 = CONCAT("insert into ",result_rel_name," (reaction_number, reaction_output,
reaction_output_type) SELECT reaction_number, name as reaction_output, 'simpleMolecule' as
reaction_output_type from (SELECT A.DB_ID as reaction_number, referenceEntity FROM
reaction_output_simpleEntity A JOIN SimpleEntity_2_referenceEntity B ON(A.output = B.DB_ID)) A1 JOIN
(select * from ReferenceEntity_2_name where name_rank = 0) B1 ON(A1.referenceEntity = B1.DB_ID);");

     PREPARE d_temp FROM @d_temp_str;
     PREPARE d_temp1 FROM @d_temp1_str;
     PREPARE temp1 FROM @temp1_str;
     PREPARE d_temp2 FROM @d_temp2_str;
     PREPARE temp2 FROM @temp2_str;
     PREPARE in_temp FROM @in_temp_str;
     PREPARE in_out_1 FROM @in_out_str_1;
     PREPARE in_out_2 FROM @in_out_str_2;
     PREPARE d_reaction_output FROM @d_reaction_output_str;
     PREPARE result_1 FROM @result_str_1;
     PREPARE result_2 FROM @result_str_2;
     PREPARE d_reaction_output_simpleEntity FROM @d_reaction_output_simpleEntity_str;

    loop_label: LOOP
       IF cntTC < 1  THEN
         LEAVE loop_label;
       END IF;
       EXECUTE d_temp1;
       EXECUTE temp1;
       EXECUTE d_temp2;
       EXECUTE temp2;
       EXECUTE d_temp;
       EXECUTE in_temp;
       EXECUTE in_out_1;
       EXECUTE in_out_2;
       SELECT count(*) INTO  cntTC FROM temp_o;
   END LOOP;
   EXECUTE result_1;
   EXECUTE result_2;
   EXECUTE d_temp; EXECUTE d_temp1; EXECUTE d_temp2; EXECUTE d_reaction_output; EXECUTE
d_reaction_output_simpleEntity;
END $$
DELIMITER ;
```

Appendix B

Reaction Data Operators



File: /media/ONSONGO/Thesis/Thesis_Draft/Compound_Edge.sql Page 1 of 3

```
DELIMITER $$
DROP PROCEDURE IF EXISTS ComputeTransitiveStartNodeNodePair $$
CREATE PROCEDURE ComputeTransitiveStartNodeNodePair(IN tc VARCHAR(20), IN stop_cnt INT)
BEGIN
  DECLARE cntTC, newRows INT DEFAULT 1;
  SET @d_temp_join_str = "drop table if exists temp_join";
  SET @i_temp_join_str = "create table temp_join as select distinct A.id as id, A.input as input, replace
(replace(CONCAT(replace(A.output,':','|'), B.output), CONCAT(':',A.input,':'), ':'), CONCAT
(':',B.input,':'), CONCAT(':|',B.input,'|:')) as output, CONCAT(A.catalyst, B.catalyst) as catalyst,
(A.distance + B.distance) as distance from temp_left A JOIN temp_right B ON(A.output like concat
('%:',B.input,':%') AND A.id != B.id) AND  A.output not like CONCAT('%',replace(B.output,':','|'),'%')";
  SET @d_non_updated_str = "drop table if exists non_updated";
  SET @i_non_updated_str = "create table non_updated as select A.* from temp_result A LEFT JOIN temp_join
B on(A.id = B.id AND A.input = B.input AND B.output like concat(replace(A.output,':','|'),'%')) where
B.id IS NULL;";
  SET @d_temp_result_str = "drop table if exists temp_result";
  SET @i_temp_result_str = "create table temp_result as select id, input, output, catalyst, distance from
non_updated union select id, input, output, catalyst, distance  from temp_join";
  SET @d_temp_left_str = "drop table if exists temp_left";
  SET @d_temp_right_str = "drop table if exists temp_right";
  SET @i_temp_left_str = "create table temp_left as select * from temp_join";
  SET @d_tc_str = CONCAT("drop table if exists ",tc);
  SET @i_tc_str = CONCAT("create table ",tc," as select id, input, replace(replace(replace
(output,'|',':'),'||',':'), '::',':') as output, replace(catalyst,'::',':') as catalyst, distance  from
temp_result");


  PREPARE d_temp_join FROM @d_temp_join_str;
  PREPARE i_temp_join FROM @i_temp_join_str;
  PREPARE d_non_updated FROM @d_non_updated_str;
  PREPARE i_non_updated FROM @i_non_updated_str;
  PREPARE d_temp_result FROM @d_temp_result_str;
  PREPARE i_temp_result FROM @i_temp_result_str;
  PREPARE d_temp_left FROM @d_temp_left_str;
  PREPARE i_temp_left FROM @i_temp_left_str;
  PREPARE d_temp_right FROM @d_temp_right_str;
  PREPARE d_tc FROM @d_tc_str;
  PREPARE i_tc FROM @i_tc_str;

  loop_label: LOOP
    IF cntTC > (stop_cnt - 1) || newRows < 1 THEN
      LEAVE loop_label ;
    END IF;
    EXECUTE  d_temp_join;   EXECUTE  i_temp_join;
    EXECUTE  d_non_updated; EXECUTE  i_non_updated;
    EXECUTE  d_temp_result; EXECUTE  i_temp_result;
    EXECUTE  d_temp_left;   EXECUTE  i_temp_left;
    EXECUTE  d_temp_join;   EXECUTE  i_temp_join;
    EXECUTE  d_temp_join;   EXECUTE  d_non_updated;
    SELECT count(*) INTO  newRows FROM temp_left;
    SET cntTC = cntTC + 1;
    COMMIT;
    END LOOP;
    EXECUTE d_tc; EXECUTE i_tc;
    EXECUTE d_temp_join; EXECUTE d_non_updated; EXECUTE  d_temp_result; EXECUTE d_temp_left; EXECUTE
d_temp_right;
END $$
DELIMITER ;

DELIMITER $$

DROP PROCEDURE IF EXISTS ComputeStartAndEndNodeRestrictedReactionPath $$

CREATE PROCEDURE cComputeStartAndEndNodeRestrictedReactionPath(IN nodes_rel VARCHAR(20), IN node_name
VARCHAR(20), IN edge_rel VARCHAR(20), IN start_n VARCHAR(20), IN end_n VARCHAR(20), IN max_iteration INT,
IN out_rel VARCHAR(20))
BEGIN
```

```
        DECLARE cntTC INT DEFAULT 20; DECLARE cnt INT DEFAULT 2;

        SET @i_path_str = CONCAT("INSERT INTO i_path(input, output, pos) SELECT DISTINCT
",start_n,",",end_n,", 1 AS pos FROM ",edge_rel," A join ",nodes_rel,
        " B on(A.",start_n,"=B.",node_name,");");
        SET @p_edges_str = CONCAT("CREATE TABLE p_edges AS SELECT DISTINCT ",start_n," as input,",end_n,"
as output FROM ",edge_rel);
        SET @p_left_rel_str = "CREATE TABLE p_left_rel AS SELECT path_id, input, output, pos from
i_path;";
        SET @p_path_str = "CREATE TABLE p_path AS SELECT path_id, input, output, pos from i_path;";
        SET @p_temp_join_str = "CREATE TABLE p_temp_join AS select temp1.path_id, temp1.A_input,
temp1.A_output, temp1.B_output, temp1.pos from
        (select A.path_id, A.input as A_input, A.output as A_output, B.output as B_output, A.pos + 1 as
pos from p_left_rel A join p_edges B on
        (A.input != B.output and A.output = B.input)) temp1 left join p_path B1 on
        (temp1.path_id = B1.path_id and temp1.A_output = B1.input and temp1.B_output = B1.output) where
B1.input IS NULL;";
        SET @p_next_left_rel_str = "CREATE TABLE p_next_left_rel AS SELECT distinct path_id, A_input AS
input, B_output AS output, pos from p_temp_join;";
        SET @p_temp_path_str = "CREATE TABLE p_temp_path AS SELECT path_id, input, output, pos FROM
p_path UNION SELECT path_id, A_output AS input,
        B_output AS output, pos FROM p_temp_join;";
        SET @u_left_rel_str = "CREATE TABLE p_left_rel AS SELECT distinct path_id, input, output, pos
FROM p_next_left_rel;";
        SET @u_path_str = "CREATE TABLE p_path AS SELECT distinct path_id, input, output, pos FROM
p_temp_path;";
        SET @out_rel_str = CONCAT("CREATE TABLE ",out_rel," AS SELECT * from p_path;");

        SET @d_edges_str = "DROP TABLE p_edges;";
        SET @d_left_rel_str = "DROP TABLE p_left_rel;";
        SET @d_path_str = "DROP TABLE p_path;";
        SET @d_temp_join_str = "DROP TABLE p_temp_join;";
        SET @d_next_left_rel_str = "DROP TABLE p_next_left_rel;";
        SET @d_temp_path_str = "DROP TABLE p_temp_path;";
        SET @d_i_path_str = "DROP TABLE i_path;";

        PREPARE i_path FROM @i_path_str;
        PREPARE p_edges FROM @p_edges_str;
        PREPARE p_left_rel FROM @p_left_rel_str;
        PREPARE p_path FROM @p_path_str;
        PREPARE p_temp_join FROM @p_temp_join_str;
        PREPARE p_next_left_rel FROM @p_next_left_rel_str;
        PREPARE p_temp_path FROM @p_temp_path_str;
        PREPARE u_left_rel FROM @u_left_rel_str;
        PREPARE u_path FROM @u_path_str;
        PREPARE s_out_rel FROM @out_rel_str;

        PREPARE d_edges FROM @d_edges_str;
        PREPARE d_left_rel FROM @d_left_rel_str;
        PREPARE d_path FROM @d_path_str;
        PREPARE d_temp_join FROM @d_temp_join_str;
        PREPARE d_next_left_rel FROM @d_next_left_rel_str;
        PREPARE d_temp_path FROM @d_temp_path_str;
        PREPARE d_i_path FROM @d_i_path_str;

                EXECUTE i_path;
                EXECUTE d_edges;
                EXECUTE p_edges;
                EXECUTE d_left_rel;
                EXECUTE p_left_rel;
                EXECUTE d_path;
                EXECUTE p_path;

        loop_label: LOOP
                IF cntTC < 1 OR cnt > max_iteration THEN
                        LEAVE loop_label ;
                END IF;
                EXECUTE d_temp_join;
                EXECUTE p_temp_join;
                EXECUTE d_next_left_rel;
                EXECUTE p_next_left_rel;
                EXECUTE d_temp_path;
                EXECUTE p_temp_path;
                EXECUTE d_left_rel;
                EXECUTE u_left_rel;
                EXECUTE d_path;
                EXECUTE u_path;

                SELECT count(*) INTO  cntTC FROM p_next_left_rel ;
                SET cnt = cnt + 1;
        END LOOP;
        EXECUTE s_out_rel;
        EXECUTE d_i_path;
        EXECUTE d_left_rel;
        EXECUTE d_path;
        EXECUTE d_temp_join;
        EXECUTE d_next_left_rel;
        EXECUTE d_temp_path;
        EXECUTE d_edges;
END $$
DELIMITER ;
```

Appendix C

ComputeMinDistTransitiveEdge

ComputeMinDistTransitiveEdge presents an alternative to transitive closure relation for answering a class of complex biological queries that requires connectivity information but not the full transitive closure. Instead of computing each transitive edge between two nodes as is the case with *transitive closure* only the shortest distance transitive edges are computed. It implements a join algorithm, *MinJoinLogarithmicTC*, for computing shortest transitive edge relation. *MinJoinLogarithmicTC* takes advantage of domain requirements (need to only compute shortest distance transitive edge) to more efficiently compute shortest transitive edge relation. It achieves this efficiency by avoiding join operations that would lead to transitive edges with distances greater than the shortest distance.

In query optimization, selection of rows is sometimes done as early as possible, especially when join operations are involved. The rationale behind this rule is that a join operation can operate on a smaller table reducing work [106]. Using the same principle *MinJoinLogarithmicTC* eliminates rows from input relations to join operations that would lead to transitive edges with distances greater than the shortest distance transitive edge. The smaller input relations will lead to reduced join operations and better performance. Before presenting the details of *MinJoinLogarithmicTC*, we will present *LogarithmicTC*. *LogarithmicTC* is a modified version of an algorithm previously discussed by Valduriez and Boral [107]. In Section C.1, *MinJoinLogarithmicTC* is compared with *LogarithmicTC* to demonstrate its efficiency in computing shortest distance transitive edge relation.

C.0.4 ComputeMinDistTransitiveEdge: Modified transitive closure algorithm

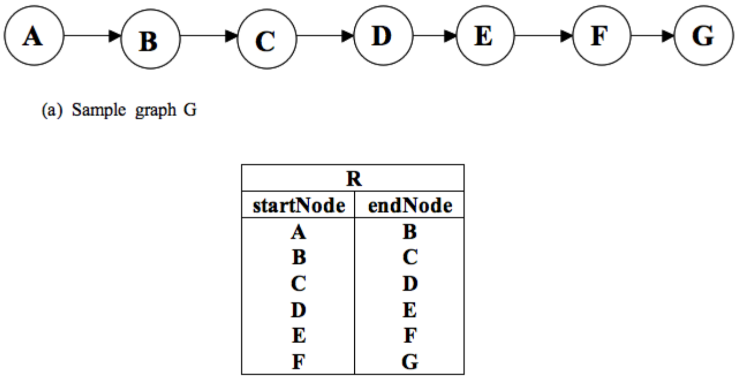
A classical approach for computing new transitive edges using joins involves use of self-join operations. This approach has one main limitation. Because new edges are combined with old edges before the self-join is performed, it results in regeneration of edges. To avoid regenerating edges, two separate joins operations can be used instead of one join operation (self-join). Valduriez and Boral [107] presented an algorithm for computing the transitive closure of a graph that uses two join operations instead of a self-join and called it *LogarithmicTC*.

To summarize how the algorithm presented by Valduriez and Boral [107] works, let R be a relation containing the initial edges of a graph, R^N contain new transitive edges generated at the N^{th} iteration and, T be a relation with all transitive edges generated so far. R^1 will thus contain new transitive edges generated after the first iteration. Instead of combining new transitive edges (R^N) with all the transitive edges generated so far (T) and performing a self-join on the resultant relation, two separate joins can be used avoiding regeneration of transitive edges. The first join will be a self-join on the new edges (R^N x R^N) and the second join will be between the new edges and all edges generated so far (R^N x T). This is the algorithm presented by Valduriez and Boral [107] to compute the transitive closure of a graph.

The algorithm by Valduriez et al can be modified to compute the shortest transitive edge relation of a graph. Each edge in a graph is given a distance of one. Along with computing new transitive edges, distance information for the transitive edges is computed. Before adding a new transitive edge, one checks to ensure a transitive edge with a shorter distance does not already exist in T.

Algorithm C.1 gives the pseudo code for the algorithm by Valduriez at al modified to compute the shortest transitive edge relation. We will still call this modified algorithm *LogarithmicTC*. It iteratively performs two join operations. At each iteration, new transitive edges are generated. The algorithm terminates when no new transitive edges are generated. A transitive edge is considered new if their does not already exist an edge connecting the two nodes in T. After each iteration, duplicate rows are eliminated. This optimization technique, eliminating duplicate rows, was also suggested by Ordonez [106].

It takes as input a graph relation(R) e.g.,Fig C.1 (b) and a result relation name.



(a) Sample graph G

R	
startNode	endNode
A	B
B	C
C	D
D	E
E	F
F	G

(b) Relation R for sample graph G

Figure C.1: Sample Graph

Figures C.2 illustrates the use of *LogarithmicTC* to compute shortest transitive edge relation for sample graph G. In the **Initialization step**, three copies of the input relation **R** are made: T, D_R and R_Δ (Lines 1 to 3, Algorithm C.1). T is the shortest transitve relation and will contain each transitive edge that has been generated so far. D_R and R_Δ will contain the newly generated transitive edges. D_R will contain new transitive edges generated by joining new edges with all transitve edges generated (RN x T). R_Δ will contain new transitive edges generated by doing a self-join on new edges (RN x RN). T, D_R and R_Δ will be used to determine these two sets of input relations. Looking at iteration 1 in Figures C.2, $A1 = A2 = B1 = B2$. The four relations contain the same set of rows because the relation R was used to initial T, D_R and R_Δ which in turn are used to set up $A1, A2, B1, B2$. *LogarithmicTC* can be summarized by the four steps below.

Step1: Update input relations to JOIN operations. This step uses T and R_Δ to update the four relations used to generate new transitive edges (Lines 5 to 8, Algorithm C.1). $A1 = R_\Delta$ and $B1 = T$. $A1$ and $B1$ are input relations to the first join. This is the join operation between new edges and all transitive edges (RN x T). $A2 = R_\Delta$ and $B2 = R_\Delta$ $A2$ and $B2$ are input relations to the second join. This is the join operation between new edges (RN x RN).

Generate new edges. Using the relations $A1, B1, A2$ and $B2$, this step performs the two JOIN operations (Lines 9 and 11, Algorithm C.1). The results of $A1\text{x}B1$ are stored in D_R while those of $A2\text{x}B2$ are stored in R_Δ. When performing the join, a distance column for the new transitive edge is computed (dist = A1.dist + B1.dist). Looking at D_R and R_Δ columns for iteration 1 in Figures C.2, we can see these rows were generated by joining $A1$ x $B1$ and $A2$ x $B2$ respectively.

Eliminate duplicates. In this step, duplicates are eliminated from the new edges generated in the previous step and stored in the relations D_R and R_Δ (Lines 10 and 12, Algorithm C.1). This duplicate elimination step was not present in the version of *LogarithmicTC* presented by Valduriez and Boral [107].

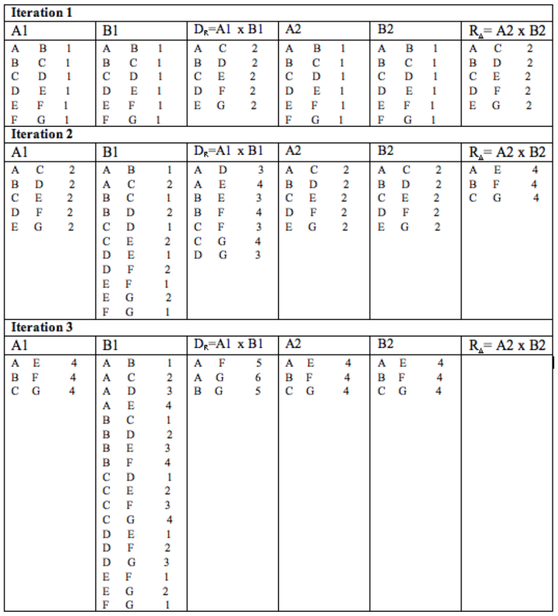
Update output relation. In this last step, the relation containing the shortest distance transitive edges (T)is updated using the duplicate free relations $D_R and R_\Delta$ (Line 13, Algorithm C.1). T is obtained by combining the contents of T, $D_R and R_\Delta$ using a union operation. Looking at $B1$ in iteration 2 of Figure C.2, we see its contents are the result of a union on T, $D_R and R_\Delta$ in iteration 1. If relation D_R is empty, the algorithm terminates. R_Δ in iteration 3 (Figures C.2) is empty. Because A1 $=R_\Delta$, in the next iteration (iteration 4), $D_R = A1\text{x}B1$ will be empty. We can therefore tell the algorithm will terminate after iteration 4. Furthermore, because R_Δ is used to update the contents of $A1$ and $A2$ (A1 $=R_\Delta$ and A2 $=R_\Delta$) which are empty after iteration 3, no new transitive edges will be generated in iteration 4.

C.0.5 Computing Shortest Transitive Edge Relation: *MinJoinLogarithmicTC*

Similar to *LogarithmicTC*, *MinJoinLogarithmicTC* iteratively performs join operations generating new transitive edges at each iteration. However, unlike *LogarithmicTC* which removes duplicates after joins, *MinJoinLogarithmicTC* removes rows in input relations that would produce duplicate entries in the result relation. *MinJoinLogarithmicTC* improves on *LogarithmicTC* by eliminating rows in $A2$ and $B2$ that are present in $A1$ and $B1$ in Algorithm C.1. To see how such an elimination can be possible, let us review some of the steps in Algorithm C.1. Looking at Lines 5 to 8 in Algorithm C.1 we know A1, A2 and B2 will always contain the same set of rows because R_Δ is used to update these relations. This observation can be verified by looking at A1, A2 and B2 in Iteration 1, 2 and 3 of Figure C.2. If A1 and A2 will always contain the same set of rows, removing

Algorithm C.1 LogarithmicTC(R:operand relation; T:result relation)

```
$T := R$
$D_R := R$
$R_\Delta := R$ {$R_\Delta$ will contain new tuples}
while $D_R \neq 0$ do
    $A1 := R_\Delta$
    $B1 := T$
    $A2 := R_\Delta$
    $B2 := R_\Delta$
    $D_{temp} := JOIN(A1, B1, A1.end = B1.start)$
    $D_R := REDUCE(D_{temp}, A1.start, B1.end, dist = A1.dist + B1.dist)$
    $R_{temp} := JOIN(A2, B2, A2.end = B2.start)$
    $R_\Delta := REDUCE(R_{temp}, A2.start, B2.end, dist = A2.dist + B2.dist)$
    $T := UNION(T, R_\Delta, D_R)$
end while
```



Iteration 1					
A1	B1	D_R=A1 x B1	A2	B2	R_Δ= A2 x B2
A B 1 B C 1 C D 1 D E 1 E F 1 F G 1	A B 1 B C 1 C D 1 D E 1 E F 1 F G 1	A C 2 B D 2 C E 2 D F 2 E G 2	A B 1 B C 1 C D 1 D E 1 E F 1 F G 1	A B 1 B C 1 C D 1 D E 1 E F 1 F G 1	A C 2 B D 2 C E 2 D F 2 E G 2
Iteration 2					
A1	B1	D_R=A1 x B1	A2	B2	R_Δ= A2 x B2
A C 2 B D 2 C E 2 D F 2 E G 2	A B 1 A C 2 B C 1 B D 2 C D 1 C E 2 D E 1 D F 2 E F 1 E G 2 F G 1	A D 3 A E 4 B E 3 B F 4 C F 3 C G 4 D G 3	A C 2 B D 2 C E 2 D F 2 E G 2	A C 2 B D 2 C E 2 D F 2 E G 2	A E 4 B F 4 C G 4
Iteration 3					
A1	B1	D_R=A1 x B1	A2	B2	R_Δ= A2 x B2
A E 4 B F 4 C G 4	A B 1 A C 2 A D 3 A E 4 B C 1 B D 2 B E 3 B F 4 C D 1 C E 2 C F 3 C G 4 D E 1 D F 2 D G 3 E F 1 E G 2 F G 1	A F 5 A G 6 B G 5	A E 4 B F 4 C G 4	A E 4 B F 4 C G 4	

Figure C.2: LogarithmicTC

rows in B2 that are also contained in B1 will avoid generating the same transitive edge twice. This observation is the motivation for one of the filtering techniques introduced in the new algorithm *MinJoinLogarithmicTC*, (Line 16, Algorithm C.2).

The second filtering technique introduced in the new algorithm *MinJoinLogarithmicTC* stems from the observation that more than one pair of nodes can be used to compute a transitive edge. For example, using the graph in Figure C.1(a) as an example, two pairs of nodes (A,D,2) - (D,F,3) and (A,E,4) - (E,F,1) can both be used to generate a path between A and F of length 5. The second filtering technique is motivated by trying to determine which sets of relations will avoid redundant pairs such (A,D,2) - (D,F,3) and (A,E,4) - (E,F,1) and is implemented in Line 16, Algorithm C.2 ($B1 := UNION(R^N_\Delta, R^{N-1}_{max\Delta})$).

Because *MinJoinLogarithmicTC* uses two join operations in each iteration similar to *LogarithmicTC*, we will show it correctly generates shortest distance transitive edges. With *LogarithmicTC*, after the first iteration, each transitive edge of length two is generated. After iteration two, each transitive edge of length four or less is generated. After iteration three, each transitive edge of length eight or less is generated. This observation can be generalized to after iteration N, each transitive edge of length 2^N or less is generated. Furthermore, we know at iteration N transitive edges of length x where $2^{N-1} \ < \ x \ \leq 2^N$ are generated.

Edges generated in iteration N - 1 and iteration N are sufficient to compute each transitive edge of length x where $2^{N-1} \ < \ x \ \leq 2^N$. Instead of using T which contains all transitive edges generated as *LogarithmicTC* does, *MinJoinLogarithmicTC* uses edges generated in iteration $N-1$ and iteration N to compute the next set of transitive edges avoiding redundant pairs. Using N=3 as an example, we will demonstate how *MinJoinLogarithmicTC* computes transitive edges.

The two joins performed by *MinJoinLogarithmicTC* are A1 x B1 and A2 x B2 where $A1 := R^N_{min\Delta}, B1 := UNION(R^N_\Delta, R^{N-1}_{max\Delta}), A2 := R^N_{max\Delta}, B2 := MINUS(R^N_\Delta, R^N_{min\Delta})$ (Lines 14 and 17, Algorithm C.2). For N=3, we know we need to generate transitive edges of length 9, 10, 11, 12, 13, 14, 15 and 16. The two relations used are R^{N-1}_Δ and R^N_Δ. Note, using R^{N-1}_Δ and R^N_Δ instead of T and R^N_Δ as *LogarithmicTC* leads to better performance since R^{N-1}_Δ is a much smaller relation compared to T. For N = $3R^{N-1}_\Delta$ contains edges of length 3 and 4 and R^N_Δ contains edges of length

5, 6, 7, 8. A join of R^N_Δ x R^N_Δ will produces new transitive edges of length 10, 11, 12, 13, 14, 15 and 16. Note, however,there are two possible ways of generating an edge of length 13. Using an edge of length six and an edge of length seven (6+7), or an edge of length five and an edge of length eight(5+8). Taking the edge with the shortest distance in R^N_Δ (5) and joining it with the edge with the longest distance in R^N_Δ (8) generates a transitive edge of length 13. We can therefore safely ignore the transitive edges generated by joining the other two edges (6+7). To eliminate this redundant pair (6+7), we proceed as follows. With N = 3, $R^N_{min\Delta}$ gives each transitive edge of length 5 and R^N_Δ gives each transitive edge of lengths 5, 6, 7, 8. A join of $R^N_{min\Delta}$ x R^N_Δ will produce edges of length (5+5, 5+6, 5+7, 5+8) = (10, 11, 12, 13). To produce edges of length 9, we will need to use the longest transitive edge produced in iteration N-1 (length 4). $R^{N-1}_{max\Delta}$ gives each edge of length 4 when N =3. So $R^N_{min\Delta}$ x R^{N-1}_Δ will generate each edge of length 9. The relations used in the first join A1 and B1 (Lines 13, Algorithm C.2) give this combination $(A1 := R^N_{min\Delta}, B1 := UNION(R^N_\Delta, R^{N-1}_{max\Delta}))$ So, with N=3, the first join A1 x B1 generates edges of length 9, 10, 11, 12, 13.

We now need to show the second join operation A2 x B2 generates the remaining edges of length 14, 15, and 16. $R^N_{max\Delta}$ gives each edge of length 8 when N =3. To generate transitive edges of length 14, 15 and 16, we use edges of length (8+6, 8+7, 8+8). Recall, R^N_Δ contains edges of length 5, 6, 7, 8. If we remove edges of length 6 ($R^N_{min\Delta}$), we can generate edges of length 14, 15, 16 without regenerating edges of length 12 (6+6). The second join operation in Algorithm C.2 (A2 x B2) accomplishes this task. $A2 := R^N_{max\Delta}$ and $B2 := MINUS(R^N_\Delta, R^N_{min\Delta})$ (Line 16, Algorithm C.2). We have shown with N = 3, the two join operation in Algorithm C.2 generate transitive edges of length 9, 10, 11, 12, 13, 14, 15 and 16.

Algorithm C.2 gives the pseudo code for *MinJoinLogarithmicTC*. The remaining commands in Algorithm C.2 are designed to update and maintain the four relations used in the joins: A1, B1, A2 and B2 and can be summarized as follows:

It takes as input a graph relation(R) and a result relation name (T). In the **Initialization step**, the relations $T, R^{N-1}_\Delta, R^{N-1}_{max\Delta}, R^{N-1}_\Delta,$ and R^N_Δ are initialized (Lines 1 to 5, Algorithm C.2). T contains the transitive edges being generated. R^{N-1}_Δ contains transitive edges newly generated at iteration N-1. $R^{N-1}_{max\Delta}$ contains longest transitive edges for each node generate in iteration N-1. For example, using B1 of iteration 2 in Figure C.3 as the relation R_{Δ}^{N-1}, $R_{max\Delta}^{N-1}$ will contain the edges [(A,C,2), (B,D,2), (C,E,2), (D,F,2), (E,G,2) and (F,G,1)]. Similary, with B1 of iteration 2 in Figure C.3 as the relation R_{Δ}^{N}, $R_{min\Delta}^{N}$ will have the edges [(A,B,1), (B,C,1), (C,D,1), (D,E,1), (E,F,1) and (F,G,1)]. R_{Δ}^{N} contains edges newly generated in iteration N. At the start of the algorithm, $N-1 = N = 1$. Relations $T1$ and $T2$ are temporary relations.

Determine duplicate generating input rows. In this step, the algorithm updates two relations, $R_{min\Delta}^{N}$ and $R_{max\Delta}^{N-1}$ (Lines 7 to 9 and Line 19, Algorithm C.2). These relations are used to determine the set of input relations that will produce non-redundant rows in the JOIN operation. The GROUP in Lines 3, Algorithm C.2 is used to determine the distance of the longest transitive edge starting at each node for nodes generated in iteration N-1 e.g., (A,1). The JOIN and PROJECT in Lines 4 and 5, Algorithm C.2 will generate the actual edge e.g., (A,B,1) in $R_{max\Delta}^{N-1}$. Similary, Lines 7 to 9, Algorithm C.2 and Lines 10 to 12, Algorithm C.2 produce $R_{min\Delta}^{N}$ and $R_{max\Delta}^{N}$ respectively. $R_{min\Delta}^{N}$, $R_{max\Delta}^{N}$, R_{Δ}^{N} and $R_{max\Delta}^{N-1}$ are used to produce A1, B1, A2 and B2.

Generate new edges. The four relations are then joined to generate new edges (Lines 14 and 17, Algorithm C.2). The new edges are stored in $D1$ and $D2$ (Lines 15 and 18, Algorithm C.2).

Update output and intermediate relations. In the final step, the shortest transitive edge relation (T) is updated. The new edges stored in D1 and D2 are added to T. Temporary relations: $R_{\Delta}^{N-1}, R_{max\Delta}^{N-1}$ and R_{Δ}^{N} are updated (Lines 19 and 20, Algorithm C.2). R_{Δ}^{N} becomes the new $R_{\Delta}^{N-1}, R_{max\Delta}^{N}$ becomes the new $R_{max\Delta}^{N-1}$ and R_{Δ}^{N} is updated to include new transitive edges in D1 and D2. If R_{Δ}^{N} is empty, the algorithm terminates. Otherwise it iterates again (Line 6, Algorithm C.2).

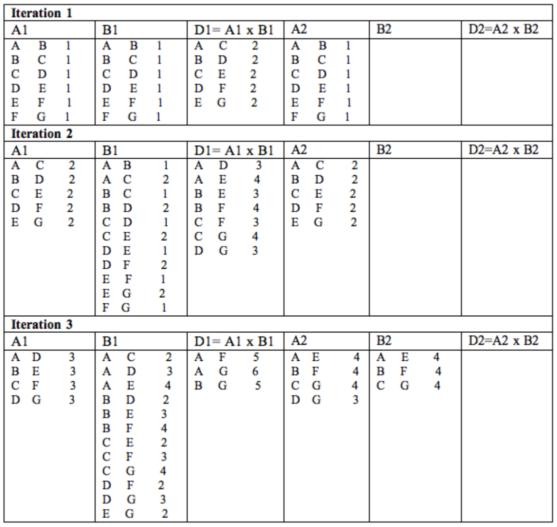
To demonstrate the robustness of our algorithm in real life applications, we experimented with a biological dataset commonly used to analyze microarray and proteomics data, the Gene Ontology database [128]. In the next section we present results of this experiment.

Algorithm C.2 MinJoinLogarithmicTC(R:operand relation; T:result relation)

```
$R := PROJECT(REDUCE(R, [start, end]), [start, end, dist = 1])$
$T := R, R_{\Delta}^{N-1} := R, R_{\Delta}^{N} := R$ $\{R_{\Delta}^{N}$ will contain new tuples$\}$
$T1 := GROUP(R_{\Delta}^{N-1}, [start, dist = MAX(distance)])$
$T2 := JOIN(R_{\Delta}^{N-1}, T1, [R_{\Delta}^{N-1}.start = T1.start \wedge R_{\Delta}^{N-1}.dist = T1.dist])$
$R_{max\Delta}^{N-1} := PROJECT(T2, [R_{\Delta}^{N-1}.start, R_{\Delta}^{N-1}.end, R_{\Delta}^{N-1}.dist])$
while $R_{\Delta}^{N} \neq 0$ do
  $T1 := GROUP(R_{\Delta}^{N}, [start, dist = MIN(dist)])$
  $T2 := JOIN(R_{\Delta}^{N}, T1, [R_{\Delta}^{N}.start = T1.start \wedge R_{\Delta}^{N}.dist = T1.dist])$
  $\mathbf{R_{min\Delta}^{N} := PROJECT(T2, [R_{\Delta}^{N}.start, R_{\Delta}^{N}.end, R_{\Delta}^{N}.dist])}$
  $T1 := GROUP(R_{\Delta}^{N}, [start, dist = MAX(dist)])$
  $T2 := JOIN(R_{\Delta}^{N}, T1, [R_{\Delta}^{N}.start = T1.start \wedge R_{\Delta}^{N}.dist = T1.dist])$
  $\mathbf{R_{max\Delta}^{N} := PROJECT(T2, [R_{\Delta}^{N}.start, R_{\Delta}^{N}.end, R_{\Delta}^{N}.dist])}$
  $A1 := R_{min\Delta}^{N}, B1 := UNION(R_{\Delta}^{N}, R_{max\Delta}^{N-1})$
  $D1_{temp} := JOIN(A1, B1, A1.end = B1.start)$
  $\mathbf{D1 := PROJECT(D1_{temp}, A1.start, B1.end, dist = A1.dist + B1.dist)}$
  $A2 := R_{max\Delta}^{N}, B2 := MINUS(R_{\Delta}^{N}, R_{min\Delta}^{N})$
  $D2_{temp} := JOIN(A2, B2, A2.end = B2.start)$
  $\mathbf{D2 := PROJECT(D2_{temp}, A2.start, B2.end, dist = A2.dist + B2.dist)}$
  $R_{\Delta}^{N-1} := R_{\Delta}^{N}, R_{max\Delta}^{N-1} := R_{max\Delta}^{N}$
  $R_{\Delta}^{N} := MINUS(UNION(D1, D2), T), T := UNION(T, R_{\Delta}^{N})$
end while
```

C.1 Experiment

To assess its performance, we will compare *MinJoinLogarithmicTC* with *LogarithmicTC*. Valduriez and Boral [107] showed that in its original form *LogarithmicTC* has a complexity that is logarithmic in the depth of a graph whose transitive closure is being evaluated. We will show that *MinJoinLogarithmicTC* not only has a logarithmic time complexity similar to *LogarithmicTC* but that by preemptively eliminating rows in the input tables, it results in lower computation costs for the shortest transitive edge relation.



Iteration 1					
A1	B1	D1= A1 x B1	A2	B2	D2=A2 x B2
A B 1 B C 1 C D 1 D E 1 E F 1 F G 1	A B 1 B C 1 C D 1 D E 1 E F 1 F G 1	A C 2 B D 2 C E 2 D F 2 E G 2	A B 1 B C 1 C D 1 D E 1 E F 1 F G 1		
Iteration 2					
A1	B1	D1= A1 x B1	A2	B2	D2=A2 x B2
A C 2 B D 2 C E 2 D F 2 E G 2	A B 1 A C 2 B C 1 B D 2 C D 1 C E 2 D E 1 D F 2 E F 1 E G 2 F G 1	A D 3 A E 4 B E 3 B F 4 C F 3 C G 4 D G 3	A C 2 B D 2 C E 2 D F 2 E G 2		
Iteration 3					
A1	B1	D1= A1 x B1	A2	B2	D2=A2 x B2
A D 3 B E 3 C F 3 D G 3	A C 2 A D 3 A E 4 B D 2 B E 3 B F 4 C E 2 C F 3 C G 4 D F 2 D G 3 E G 2	A F 5 A G 6 B G 5	A E 4 B F 4 C G 4 D G 3	A E 4 B F 4 C G 4	

Figure C.3: MinJoinLogarithmicTC

With different scientists working on the same problems, the discovery nature of biological science naturally leads to scientists naming what they find differently. Giving different names to what turn out to be the same concept, impedes science making it impossible for humans and computers alike to analyze biological data [71]. The Gene Ontology database (GO)was formed by the consortium to overcome this limitation. It consists of genes, gene products and terms associated with these genes and gene products. Terms are concepts used to organize the gene and gene products. GO organizes terms and the parent-child relationships between terms into three separate ontologies: biological processes, molecular functions and cellular components. Each ontology forms a directed acyclic graph (DAG).

We used the two operators on the same input graphs to generate the shortest transitive edge relation. We tested the performance of the two operators as the size of the graph increased by varying the number of nodes in the input graph. The total size of intermediate JOIN results, together with execution times were recorded. Figures C.4 and C.5 present the results of the experiment.

Figure C.4 gives the execution times of *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases. The x-axis represents the number of nodes in the input graph. The y-axis lists the execution times (seconds) for the two operators.

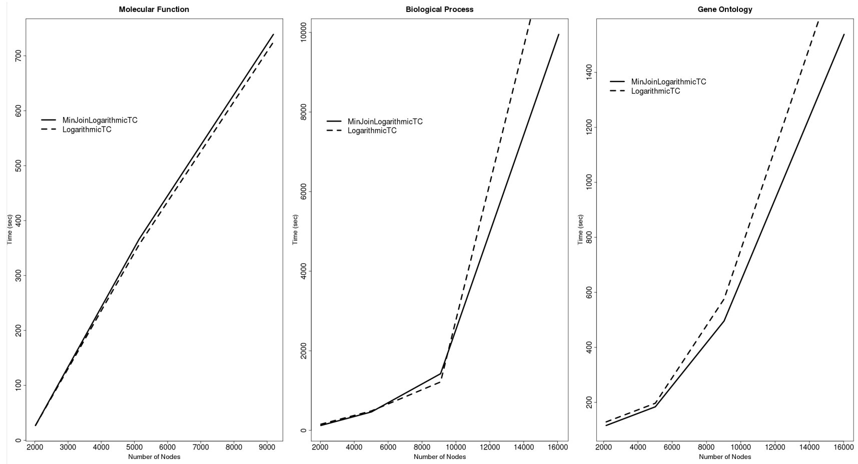


Figure C.4: Execution times of *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases

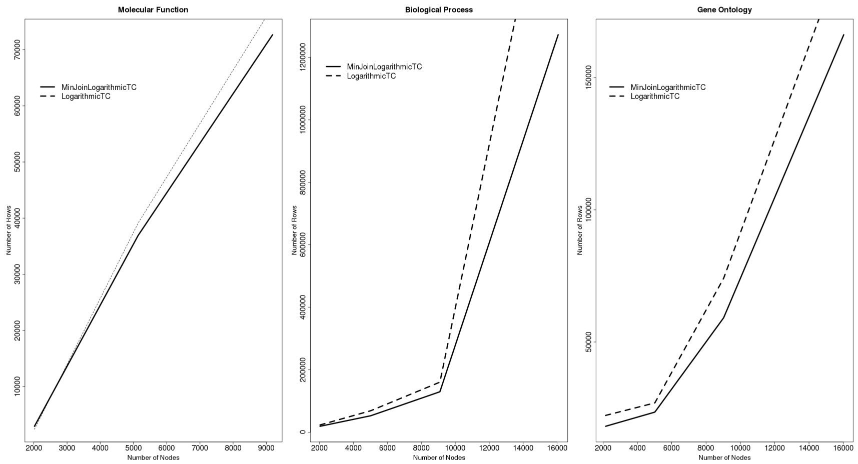


Figure C.5: Number of rows in intermediate JOIN relations for *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases

Figure C.5 gives the total size of intermediate JOIN relations for *MinJoinLogarithmicTC* and *LogarithmicTC* as the size of the input graph increases. The x-axis represents the number of nodes in the input graphs. The y-axis lists the size of intermediate result relations before duplicate entries are removed.

C.2 Discussion and Conclusion

Figures C.2 and C.3 illustrate the usage of *LogarithmicTC* and *MinJoinLogarithmicTC* in computing shortest transitive edge relation for sample graph G. This illustration demonstrates *MinJoinLogarithmicTC* filters out rows from input tables that would otherwise lead to duplicate entries. At each iteration, both algorithms perform two join operations. The relation $B2$ in *MinJoinLogarithmicTC* is generated by deleting each edge in the relation R_{Δ}^{N} that is also present in the relation $R_{max\Delta}^{N-1}$ (Line 16, Algorithm C.2). This MINUS operation preemptively eliminates rows from the input relation $B2$ that would produce rows that have already been generated using the first JOIN operation ($D1 = A1$ x $B1$). Note, the UNION used to generate the relation $B1$ in *MinJoinLogarithmicTC* removes duplicates (Line 13, Algorithm C.2), an additional filtering step that removes duplicates from the input relation $B1$.

Comparing **Iteration 1** in Figures C.2 and C.3, *MinJoinLogarithmicTC* is able to establish rows in $A2$ and $B2$ are already contained in $A1$ and $B1$. As a result, $B2$ is empty and hence *MinJoinLogarithmicTC* avoids performing the second JOIN operation ($D2 = A2$ x $B2$). The same effect can be observed in **Iteration 2**. Unlike *MinJoinLogarithmicTC, LogarithmicTC* performs the second join opeation ($D2 = A2$ x $B2$) resulting in transitive edges already computed by the first join operation ($D1 = A1$ x $B1$). This difference in size of intermediate relations can be seen in Figure C.5.

For small sized graphs (9000 nodes or less), there was no discernable difference in the execution times betweeen *MinJoinLogarithmicTC* and *LogarithmicTC*. In fact, *LogarithmicTC* appeared to have lower execution times. *MinJoinLogarithmicTC* uses more temporary relations when computing the shortest transitive edge relations. The lower execution times by *LogarithmicTC* for small input size graphs could be a result of the overhead cost of *MinJoinLogarithmicTC* incurred by the higher number of temporary relations maintained by the operator. For small graphs, the overhead cost of creating and maintaining these temporary relations is greater than the time saved by *MinJoinLogarithmicTC*. However, as the size of the graph increases, *MinJoinLogarithmicTC* consistently outperforms *LogarithmicTC*.

In conclusion, *MinJoinLogarithmicTC* not only generates the shortest transitive edge relation of a graph, a functionality needed to answer complex biological queries. It does so in a more efficient manner that does not generate duplicate entries in intermediate join relations.

Appendix D

Data Tables

Table D.1: Pathways in Reactome associated with cancer development.

Pathway Name
S-specific transcription in mitotic cell cycle
G2/M-specific transcription in mitotic cell cycle
Cell Cycle, Mitotic
Cell Cycle Checkpoints
Transcriptional activation of cell cycle inhibitor p21
G2-specific transcription in mitotic cell cycle
ERK1 activation
ERK activation
ERK2 activation
PKA-mediated activation of ERK2
Prolonged ERK activation events
APC/C-mediated degradation of cell cycle proteins
Signaling by EGFR
Grb2 events in EGFR signaling
Shc events in EGFR signaling

Continued on next page

Table D.1 – continued from previous page

Pathway Name
EGFR downregulation
Signalling to ERKs
Signaling by VEGF
Neurophilin interactions with VEGF and VEGFR
VEGF ligand-receptor interactions
Degradation of beta-catenin by the destruction complex
VEGF binds to VEGFR leading to receptor dimerization
Beta-catenin phosphorylation cascade
ERK/MAPK targets
Signalling to ERK5
ERKs are inactivated
EGFR interacts with phospholipase C-gamma

Table D.2: Transitive Edges for Differentially Expressed Soluble proteins interacting with pathways associated with cancer development.

StartNode	**EndNode**	**Distance**
P02679	P46108	1
P02679	Q07889	1
P02679	P62993	1
P02679	P31946	2
P02679	P04629	4
P02679	Q9ULH0	5
P31946	P01111	1
P31946	P04629	1
P31946	Q02750	1
P31946	Q9ULH0	1
P31946	P01112	1
P31946	P36507	1
P31946	P01138	1
P31946	P01116	1
P31946	P46108	1
P31946	P04049	1
P35222	P25054	1
P35222	O15169	1
P35222	P63208	1
P35222	P48729	1
P35222	Q13616	1
P35222	P49841	1
P35222	Q9Y297	1
P35222	P62988	1
P61088	P62988	1

Continued on next page

Table D.2 – continued from previous page

StartNode	EndNode	Distance
P61978	P53803	1
P61978	P52435	1
P61978	P13984	1
P61978	P62487	1
P61978	P19388	1
P61978	P36954	1
P61978	P19387	1
P61978	P61218	1
P61978	P30876	1
P61978	P52434	1
P61978	P24928	1
P61978	P35269	1
P61978	P62875	1
P61978	O15514	1
P63104	P04049	1

Table D.3: Soluble Saliva Proteins Differentially expressed between pre-malignant and malignant oral lesions.

Protein Name	**Ratio** $(\frac{malignant}{pre-malignant})$
SP:Q2M2I5	2.039473684
SP:P62820-1	1.565789474
SP:O95678	2.559210526
SP:P05164-1	1.480263158
SP:Q01546	2.25
SP:P05386	1.25
SP:P13645	1.506578947
SP:P13646-1	2.348684211
SP:P13647	3.743421053
SP:P15104	2.585526316
SP:Q5VZM2-1	1.453947368
SP:P15924-1	1.723684211
SP:Q9NWS1-2	1.276315789
SP:P35222-1	2.131578947
SP:Q9UBH0	1.434210526
SP:P00450	1.322368421
SP:Q86YJ6-4	1.605263158
SP:P00751-1	1.368421053
SP:P29034	1.769736842
SP:P35908	1.519736842
SP:P02679-1	1.5
SP:P04114	2.263157895
SP:Q6UX06	1.480263158
SP:P04196	1.203947368
SP:P02748	1.657894737
SP:P02765	2.243421053

Continued on next page

Table D.3 – continued from previous page

Protein Name	Ratio ($\frac{malignant}{pre-malignant}$)
TR:A6NBZ8	1.25
SP:P02787	1.440789474
SP:P02790	1.677631579
SP:P04217	1.355263158
SP:Q92876-1	2.171052632
SP:P06702	2.355263158
SP:Q7Z406-5	1.763157895
SP:P0C0L4	1.776315789
SP:P01031	1.394736842
SP:Q9Y3B3	2.440789474
SP:O15321	2.414473684
VE:OTTHUMP00000164639	1.796052632
EN:ENSP00000352479	1.907894737
SP:Q92576-1	1.861842105
TR:Q8IUK7	1.328947368
SP:P08779	3.802631579
VE:OTTHUMP00000178329	1.401315789
SP:P04083	1.901315789
SP:P31151	1.782894737
VE:OTTHUMP00000172823	1.361842105
SP:P19012	3.210526316
VE:OTTHUMP00000167597	1.776315789
SP:P12035	2.006578947
SP:P19827	1.598684211
SP:P04259	6.190789474
SP:P07858	1.611842105
TR:A8MQC9	1.453947368
SP:P02749	1.447368421

Continued on next page

Table D.3 – continued from previous page

Protein Name	**Ratio** ($\frac{malignant}{pre-malignant}$)
SP:P04004	2.230263158
SP:P02538	2.868421053
SP:Q9H2S1	5.618421053
SP:P19823	2
EN:ENSP00000262269	2.046052632
TR:Q96PQ8	1.467105263
SP:P02533	2.243421053
VE:OTTHUMP00000029269	1.578947368
TR:Q6NSB4	1.677631579
SP:Q04695	3.177631579
TR:Q8TC63	3.401315789
SP:P02774	1.881578947
SP:Q9Y536	1.440789474
VE:OTTHUMP00000021887	1.513157895
VE:OTTHUMP00000065239	2.618421053
TR:Q9Y509	1.282894737
SP:P28325	0.723684211
EN:ENSP00000374796	0.736842105
SP:P61088	0.802631579
SP:P28676	0.789473684
SP:P01833	0.730263158
SP:A0AV96-2	0.171052632
SP:O15144	0.736842105
SP:Q01082-1	0.828947368
SP:Q9UIV8-1	0.611842105
SP:P21128	0.631578947
VE:OTTHUMP00000025062	0.664473684
SP:P62736	0.743421053

Continued on next page

Table D.3 – continued from previous page

Protein Name	Ratio ($\frac{malignant}{pre-malignant}$)
SP:P31025	0.210526316
SP:P13987	0.828947368
SP:O43278-2	0.460526316
SP:P23528	0.756578947
SP:P07602-1	0.697368421
SP:P15814	0.644736842
SP:Q15080-1	0.677631579
RV:NP_079105	0.677631579
SP:Q99880	0.539473684
SP:P61626	0.355263158
SP:P78417	0.815789474
SP:P63104	0.697368421
SP:P19961	0.453947368
SP:P02647	0.598684211
SP:P02652	0.427631579
VE:OTTHUMP00000031679	0.592105263
VE:OTTHUMP00000066273	0.697368421
SP:P47756-1	0.802631579
SP:P14780	0.730263158
HI:HIT000321643	0.736842105
SP:P01033	0.513157895
TR:Q8N4G4	0.638157895
VE:OTTHUMP00000160357	0.782894737
SP:P61978-1	0.743421053
SP:P31946-1	0.625
SP:P16401	0.578947368
SP:P02008	0.203947368
SP:P00918	0.578947368

Continued on next page

Table D.3 – continued from previous page

Protein Name	**Ratio** $(\frac{malignant}{pre-malignant})$
SP:P29966	0.703947368
SP:P26038	0.743421053
SP:P18206-2	0.802631579
SP:P10909	0.723684211
SP:Q8TDL5-1	0.769736842
SP:P23280	0.532894737
SP:Q8N4F0	0.802631579
SP:O60437	0.717105263
SP:Q99497	0.809210526
SP:P80723	0.822368421
SP:P20061	0.723684211
VE:OTTHUMP00000080290	0.769736842
SP:P08670	0.723684211
TR:Q6N091	0.703947368
SP:P04075	0.631578947
TR:Q8WUK1	0.796052632
SP:P68871	0.190789474
TR:Q0ZCH6	0.684210526
SP:Q01469	0.519736842
SP:P18135	0.756578947
TR:Q8N5K4	0.565789474
SP:P01764	0.677631579
SP:P04745	0.519736842
SP:P01714	0.703947368
SP:P01766	0.736842105
SP:P01814	0.736842105
TR:Q8WY24	0.546052632
TR:Q9UL80	0.796052632

Continued on next page

Table D.3 – continued from previous page

Protein Name	**Ratio** $(\frac{malignant}{pre-malignant})$
TR:Q9UL85	0.815789474
SP:P04209	0.559210526
SP:P04220	0.828947368
TR:Q96K68	0.743421053
SP:P01598	0.756578947
SP:P01622	0.710526316
SP:P02042	0.243421053