

Modeling Uncertainty in Data Integration for Improving Protein Function
Assignment

Brenton E. Louie

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

Doctor of Philosophy

University of Washington

2008

Program Authorized to Offer Degree:
Medical Education
and Biomedical Informatics

UMI Number: 3318214

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform 3318214

Copyright 2008 by ProQuest LLC.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Brenton E. Louie

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

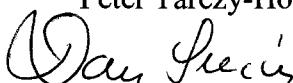


Peter Tarczy-Hornoch

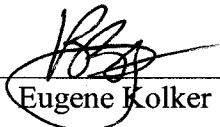
Reading Committee:



Peter Tarczy-Hornoch



Dan Suci



Eugene Kolker

Date:

June 5, 2008

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature Brenton L. W.

Date 6/5/08

University of Washington

Abstract

Modeling Uncertainty in Data Integration for Improving Protein Function Assignment

Brenton E. Louie

Chair of the Supervisory Committee:

Professor Peter Tarczy-Hornoch

Department of Medical Education and Biomedical Informatics

In this work we describe the development and evaluation of the BioMiner system for protein functional annotation. BioMiner is the implementation of a novel uncertainty model for annotation and is based on the Uncertainty in Information Integration (UII) system, a general-purpose data integration system with extended functionality to handle uncertainty in data. The informatics contributions of our work are as follows: 1) we develop and implement a first-in-class uncertainty model for annotation and illustrate the validity of the model, 2) we show that the uncertainty model is reliable by evaluating its robustness through a principled methodology, and 3) we demonstrate that the uncertainty model performs better than existing, commonly utilized, approaches through a rigorous performance evaluation. The application of BioMiner also contributes to the expansion of domain knowledge by accurately identifying functions for proteins of unknown function, a problem of utmost importance to biology.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iii
LIST OF TABLES	x
GLOSSARY	xiii
Chapter 1 : INTRODUCTION	1
1.1 Biological Background and Significance	1
1.2 Research Questions	6
1.3 Related Work	7
1.4 Contributions of this Dissertation	8
1.5 Outline of this dissertation	9
Chapter 2 : ASSIGNING FUNCTION TO PROTEINS	10
2.1 Background: Protein Annotation	10
2.2 Related Work: Current Approaches In Protein Annotation	15
2.3 Challenges for Computational Annotation Systems	21
2.4 Discussion	26
Chapter 3 : BIOLOGICAL DATA INTEGRATION	29
3.1 Overview Of Data Integration Concepts	29
3.2 Related Work: Existing Data Integration Systems	33
3.3 The BioMediator Data Integration System	37
3.4 Uncertainty: The Uncertainty In Information Integration (UII) Project	43
3.5 Discussion	55
Chapter 4 : THE BIOMINER SYSTEM	56
4.1 Design of the BioMiner System for Protein Annotation	56
4.2 Related Work	63
4.3 Implementation of the BioMiner System	64

4.4 Evolution of the BioMiner system	72
4.5 Discussion	73
Chapter 5 : THE ROBUSTNESS OF BIOMINER	76
5.1 The stability of results from the BioMiner system.....	76
5.2 Related Work: Sensitivity analyses in Bayesian Networks.....	79
5.3 A Sensitivity Analysis of the BioMiner System	81
5.4 Study Evaluation Protocol.....	87
5.5 Results	95
5.6 Discussion	106
Chapter 6 : EVALUATION OF BIOMINER	110
6.1 Multiple Evaluations of the BioMiner System.....	110
6.2 Related Work: Evaluating Computational Annotation Systems	112
6.3 Proof of Concept Evaluation #1: Relevance Ranking.....	113
6.4 Proof of Concept Evaluation #2: protein annotation.....	117
6.5 BioMiner Evaluation Study: hypothetical protein annotation.....	124
6.6 Discussion	137
Chapter 7 : SUMMARY AND CONCLUSIONS	141
7.1 Research Contributions	141
7.2 Limitations.....	145
7.3 Summary of Research	148
Bibliography.....	150
Appendix A: Annotation Datasets.....	162
Appendix B: Annotation Method	167

LIST OF FIGURES

Figure Number	Page
Figure 1.1: A genome is the blueprint for creating an organism. It is encoded in DNA, a base-four sequence of nucleic acids (A, T, G, and C). Within a genome are specific subsequences called genes. A gene is a template which encodes a base-twenty sequence of amino acids, or a protein. Proteins carry out all the biological processes in an organism. The term gene and protein are sometimes used interchangeably in the biomedical literature. This is understandable given the “gene-codes-for-protein” relationship, although this is confusing to non-experts. In this dissertation, the terms gene and protein are generally interpreted as synonyms.....	1
Figure 1.2: Novel proteins in the bacterium <i>S. oneidensis</i> allow it to “breathe” hexavalent uranium. This converts hexavalent uranium to uraninite which is insoluble in water and much easier to clean-up (courtesy of Eugene Kolker).....	3
Figure 1.3: The number of biological databases has continued to increase over time, creating a significant data integration challenge in computational protein annotation (courtesy of [12]).....	5
Figure 2.1: The glycolysis pathway, which provides an illustration of protein function. The hexokinase protein is an enzyme which catalyzes the first reaction of the process by which the human body converts sugar to ATP, a form of energy usable by cells (www.biocarta.com).....	10
Figure 2.2: To annotate proteins in the current paradigm, biological researchers manually query and compile data and information from multiple, non-interoperable biological data sources which greatly slows the pace of annotation. (Figure adapted and modified from [12]).....	13
Figure 2.3: A new paradigm for annotation using data integration. Users query software which handles the specific querying and compiling of results from the individual data sources. To the user, all the individual data sources appear as a single database (adapted from [28]).....	14
Figure 2.4: An illustration of the proportions of various types of annotations in the organism <i>Shewanella oneidensis</i> . Only a small percentage of annotations can be ascribed to actual experiments (about 5%). Most proteins have been annotated by computational means (sequence comparisons) or are of unknown function (Courtesy of Eugene Kolker, PhD).	17

Figure 2.5: Systems for computational protein annotation generally follow this three-tiered approach. Tier one is a data-management system, Tier 2 includes a way to query the data, and Tier 3 provides views of the data. This figure is courtesy of [27]..... 20

Figure 3.1: Architecture of the BioMediator system. The core components are the data interfaces or wrappers, the Source Knowledge Base which contains the mediated schema, the Query Processor, and the User Query Interfaces (courtesy of [55]). . 38

Figure 3.2: BioMediator result viewed as a graph with nodes akin to mediated schema concepts and edges referring to relationships between concepts. Results are derived from an initial seed query for the HK1 gene (Gene:symbol="HK1"), after 1 expansion..... 42

Figure 3.3: The same result set as in Figure 3.4 after 4 expansions. The size of the result set (e.g. nodes and edges) becomes large very quickly. Users have difficulty analyzing and selecting relevant information from result sets of this size..... 43

Figure 3.4: Illustration of the Ps metric. The Ps metric is a user's belief in the quality of a particular mediated schema entity from a particular source, which is interpreted probabilistically (0.0-1.0) value. 49

Figure 3.5: Illustration of the Qs metric. The Qs metric is a user's belief in the quality of a particular relationship, as defined in the mediated schema, between two sources, which is interpreted probabilistically (0.0-1.0 value)..... 49

Figure 3.6: Illustration of the Pr metric. This metric is a users belief in a particular data record of the same mediated schema type and data source. It is interpreted probabilistically as a 0.0-1.0 value, which is dynamically determined when a record is retrieved. 50

Figure 3.7: Illustration of the Qr metric. This metric is a user's belief in a particular cross-reference, or link, between two data records. It is interpreted probabilistically as a 0.0-1.0 value, and is dynamically determined when records are retrieved, much like the Pr metric. 50

Figure 3.8: An illustration of a result graph from the UII system annotated with uncertainty metrics. Ps and Pr metrics are assigned to the nodes, which correspond to mediated schema entities. Qs and Qr metrics are assigned to the edges, which correspond to relationships in the mediated schema. 51

Figure 3.9: An illustrated example of the scoring algorithm in the UII system. A hypothetical result graph is shown here with the seed node marked "S". Each node has an associated trial vector (*italics*) of length N which the number of trials

(initially 1,000 in first implementation of UII). Each bit in the trial vector is set according to a “coin-flip” which is based on the uncertainty metric values of the node (e.g. $P_{s_{n1}} * P_{r_{n1}}$). For the seed node, all bits are set to 1, since that is guaranteed to be in the graph. Also associated with each node is a success vector of length N. The links (edges) between nodes also contain success and trial vectors, but they are not shown in this figure. The success vectors are set via a depth-first search (DFS) of the result graph, which is directed. Bits are set in success vectors based on the following operations: 1) for each edge, a new vector is formed by the logical AND of the head node success vector, the edge trial vector, and the tail node trial vector, 2) The bits of the success vector of the tail node are set based on a logical OR operation between the vector generated in the previous step and the current success vector of the tail node., and 3) if the previous OR operation results in the setting of any new bits, the DFS continues on this path. This final step ensures that multiple paths to a node are accounted for, such as the node where the arrows converge in the above figure. The final relevance score for each node is calculated as k/N , which is the number set bits in a nodes trial vector divided by the number of trials..... 53

Figure 3.10: An illustration of a result graph from the UII system annotated with uncertainty metrics as well as global relevance scores. The relevance score can be utilized by an appropriate interface to provide ranked result sets to users, facilitating easier inspection of result sets..... 54

Figure 4.1: A conceptual diagram illustrating the major entities and relationships of the mediated schema in the BioMiner system. A protein of unknown function represents the user or “seed” query. BioMiner then finds other proteins, domains, or families which are related to the query protein (by utilizing the search algorithm in each particular data source). These other entities may provide a free-text description of a biological function or may reference a Gene Ontology term. Some sources, such as the “Gene Database” here, provide descriptions of supporting biological evidence for a particular function as well. (diagram courtesy of Wolfgang Gatterbauer (modified), derived from schema in Figure 4.2..... 60

Figure 4.2: The mediated schema in BioMiner. The primary entities such as Protein, Family, and HierarchicalTerm can be seen here along with their relationships. SequenceSimilarity, ProteinDatabaseHit, and ConservedDomainDatabaseHit are reified relationships in the BioMiner schema which represent the quality of the similarity relationship between a query protein and another protein, protein family, or conserved domain..... 67

Figure 4.3: A portion of the Source Knowledge Base (SKB) in the BioMiner system. The SKB is the mediated schema with data sources and entities and relationship within and between the sources. Included are possible numbers of instantiated entities from each source. Some protein function data sources were omitted but their entities and relationships would be identical to Pfam and TigrFam in this

diagram. Relationships between sources represent direct references between, such as GO term identifiers in Pfam referencing GO terms in Gennav (AmiGO), or are the result of search algorithms such as when NCBIblast is searched with the query protein via BLAST. With the exception of the “UserQuery to Protein” link, relationships can be one-to-many. (This is an instantiation of a portion of the mediated schema in BioMiner (Figure 4.2). The diagram is courtesy of Wolfgang Gatterbauer (modified))..... 67

Figure 4.4: Table view of ranked results from BioMiner system. These results indicate possible functions for the hypothetical protein SO4413. In this case, the top ranking result is “cysteine desulfurases, SufS subfamily” from the TIGRFAM database, but only Family entities are shown in this view. Results for the same hypothetical protein are also shown in the Generic-Genome-Browser (GGB) in Figure 4.5, but the top-ranking result is different..... 70

Figure 4.5: Results from BioMiner as viewed in the GGB. The top hit is “Selenocysteine Lyase” from the COG database. The GGB allows for display of different mediated schema entities (such as ConservedDomains and Families) in the same track, which is called “Functional Domains” in this case. Grouping different entities in the same track allows for easier inspection by relevance score. So, for the hypothetical protein SO4413, the highest ranking result is “Selenocysteine Lyase” from the COG database, and not “Aminotransferase” from the Pfam database as a user might conclude from Figure 4.4. 71

Figure 5.1: Example of a result “graph” from the BioMiner system. The green nodes represent GO terms, as queried in the Gennav database. The “Query1” node represents the initial starting query, a gene of unknown function for example. In this case the GO function “mismatch repair” is pointed to by results from both Pfam and TIGRFAM. It is therefore more likely that “mismatch repair” will achieve a higher relevance score (and higher ranking) than the other three GO terms given that “mismatch repair” has a greater number of paths to it than the other GO terms. 88

Figure 5.2: Effect of adding log-odds normal noise to a default parameter of 0.8. A standard deviation of 0.1 (blue), 0.2 (pink), or 0.5 (green) in this figure indicates a unimodal density function. Standard deviations greater than 1.0 indicate bimodal density functions with most values near 0.0 or 1.0 (black, red, blue). The formula for adding log-odds normal noise and graph are both courtesy of [95]. 93

Figure 5.3: Distribution of perturbed probabilities when the input probability is near or very near 1.0, under various standard deviations. If the input probability is very near 1.0, the perturbed probability also stays near 1.0, even after addition of noise at 2.0 standard deviations. 93

Figure 5.4: One-way sensitivity analysis for eleven database-parameter combinations of simple 0.0-1.0 perturbations. These perturbations are described in section 5.4.3. These are single-parameter perturbations, such as varying the P_s parameter from 0.0 to 1.0 in 0.1 increments for one data source. The labels stand for database/uncertainty metric combinations. For example, EntrezGene- Q_s is the perturbation for the EntrezGene database and Q_s parameter. The X-axis represents the perturbation, i.e. value of the parameter (uncertainty metric). The Y-axis is the *macro-average precision*, or the average of the average precision for each gene under that perturbation. Overall, the macro-average precision of the system under the various perturbations remains near 0.8, which is very close to the macro-average precision of the system under default parameters (0.837). The exception is when parameters values are less 0.2..... 96

Figure 5.5: One-way sensitivity analysis for the five perturbations involving a function or lookup table. These perturbations are described in section 5.4.3. Low, linear, and, high refer to the properties of the functions used in this perturbation, i.e. increasing to 1.0 at a slower or faster rate. The labels stand for database/uncertainty metric combinations. For example, EntrezGene- P_r represents the perturbation for the EntrezGene database and P_r metric. The X-axis represents the perturbation, i.e. the value of the parameter (uncertainty metric). The Y-axis is the *macro-average precision*, or the average of the average precision of each gene under that perturbation. The macro-average precision is stable in most cases with a value near 0.8, near the macro-average precision of the system under default parameters (0.837). 98

Figure 5.6: The average precision of the BioMiner system after addition of log-odds normal noise to all default parameters at 0.5, 1.0, 2.0, and 3.0 standard deviations for each of the twenty genes in the test set. These perturbations are described in section 5.4.4. The labels show the amount of perturbation for each gene. Also included are the average precision for each gene in the random-case, worst-case, or default parameter values. Random and worstAP refer to our baseline conditions (section 5.4.5). The X-axis represents the genes and the Y-axis represents the average precision for the gene, for various perturbations. The average precision only begins to degrade significantly after addition of log-odds normal noise greater than 1.0 standard deviation. This figure also includes random and worst-case average precision for each gene. 100

Figure 5.7: The performance of BioMiner under randomized conditions of its parameter values. These are described in section 5.4.5. The X-axis represents three random perturbation studies. RandomI and RandomII are two separate sensitivity analysis result where all parameters in the BioMiner system were randomly assigned probabilistic values. RandomResults were determined mathematically. The Y-axis is the *macro-average precision*, or the average of the average precision of each gene under that perturbation. The macro-average precision of the system under random assignment is significantly worse than the

macro-average precision under the default parameters (0.473, 0.450, and 0.418 versus 0.837, $p=5.867e-11$, $p=3.029e-12$, and $p=4.967e-13$, respectively). 101

Figure 5.8: Macro-average precision of worst-case, random, and default parameters in the BioMiner system at 100% recall and at 25 results, with 95% confidence intervals. The X-axis represents the perturbation studies and the Y-axis is the *macro-average precision*. The macro-average precision is the average of the average precision of each gene under that perturbation. Random at 100% recall is calculated the same as in section 5.4.5. 104

Figure 5.9: Summary of sensitivity analysis results for the BioMiner system. The Y-axis is the *macro-average precision*, which is the average of the average precision of each gene and is a summary measure of the performance of BioMiner. The X-axis represents various sensitivity analysis studies. “DEF@100” and “DEF@25” refers to macro-average precision at 100% recall and at 25 results respectively. If these are not included in the study name then macro-average precision is calculated at 100% recall. “DEF” stands for default parameters, One-wayI stands for the one-way sensitivity analysis under 0.0-1.0 perturbations, which are described in section 5.3.3. One-wayII stands for the one-way sensitivity analysis under function/lookup perturbations which are also described in section 5.3.3. “Noise(0.5-2.0)” are multi-way sensitivity analysis studies, which are described in section 5.3.4. “WC” stands for worst-case, and “RandomI” and “RandomII” are the performance of BioMiner under randomized conditions which are described in section 5.4.5. Results indicate that the performance of BioMiner is remarkably stable under systematic perturbations of its default uncertainty metrics and significantly outperforms simulated randomized or worst-case performance. 105

Figure 6.1: Results for the BioMiner system for *S. oneidensis* locus (gene) SO0265 displayed in the GGB with results from various databases (under “Functional Domains”) ranked and highlighted according to their relevance scores. The highest ranking result from BioMiner is “Cytochrome c biogenesis factor” (COG4235). It also spans the greatest length of the query protein as compared to the other functional domains. The BioMiner annotation produced in this case would indeed be “Cytochrome c biogenesis factor”, which agrees with the manually produced annotation. 120

Figure 6.2: The *S. oneidensis* locus (gene) SO4413 with results from the BioMiner system displayed in the GGB. The BioMiner produced annotation is “Selenocystine lyase” (COG0520), but the manually produced annotation is “Kynureninase” (COG3844). When this protein is subjected to a BLAST/PSI-BLAST search however, most results are annotated as “Aminotransferase” or “Cysteine desulfurase” (COG1104). 123

Figure 7.1: A result “graph” from the BioMiner system. The green nodes are GO terms, which represent functions predicted by the system. The “Query1” node

represent the initial starting query, a gene of unknown function for example. In this case the GO function “mismatch repair” is pointed to by results from both Pfam and TIGRFAM, as opposed to “ATP binding” and “Mo-molybdopterin cofactor biosynthesis” which are only pointed to by one source each. For the 20 genes used as a gold-standard in the sensitivity analysis study, very good rankings of predicted functions can be achieved simply by considering the number of paths to a GO term. 147

LIST OF TABLES

Table Number	Page
<p>Table 2.1: The percentage of hypothetical genes (i.e. of unknown function) in selected organisms. The percentage of genes with predicted function is only an estimate, as less than 5% of annotations are known to be experimentally derived [16]. In practice, it is often difficult to the true percentages of computational and experimental annotations as this information is not often stored in protein database records. This table is adapted and abbreviated from [17].</p>	12
<p>Table 3.1: A summary of the strengths and limitations of data warehouses and federations, adapted from [51].</p>	31
<p>Table 3.2: Matching needs and requirements of protein annotation and data integration technologies. These needs are best met by the BioMediator Data Integration System (section 3.3).</p>	36
<p>Table 3.3: The probabilistic metrics in the UII data integration system. There are two at the “Set” or database level and two at the individual record level. They represent the uncertainty in data records and the relationship between data records as well as the uncertainty in data sources or links between data sources.</p>	47
<p>Table 4.1: Data sources in the BioMiner federation, a description of their contents, and their rationale for incorporation.....</p>	65
<p>Table 4.2: Data sources in BioMiner with their entities and relationships. Relationships are between databases and only show the references which point “outward”, i.e. refer to other databases.</p>	68
<p>Table 4.3: The most important uncertainty metrics in the BioMiner system. Uncertainty metrics for “Gene” entities are calculated based on their Status Code values from Entrez Gene. Uncertainty metrics for “Evidence” entities are based values specified by Gene Ontology evidence codes, which indicate the supporting evidence for a particular function. Protein->Domain, Protein->Protein, and Protein->Family are relationships which are derived via search algorithms which are associated with a score (evalue, expect), which is then converted to a probabilistic value via the function shown. The “Metric” column refers to UII uncertainty metrics defined in Chapter 3, section 3.4.3.</p>	69
<p>Table 5.1: Example of an average precision calculation. There are eight total results of which four are relevant. Four is also the total number of relevant results (100% recall). For this result set the average precision is $(1.00+0.67+0.75+0.5)/4 = 0.73$. If the average precision of this result set is averaged with other result sets, the</p>	

measure is called *macro-average precision*. (Diagram inspired by Callan, 2007).
..... 84

Table 5.2: 20 genes with reliable function assignment to be used to query the BioMiner system. Note that, on average 46.6% of the function evidence is from “non-computational” (or “non-electronic”) sources (%Non-Computational), such as direct experimentation. This is in contrast to estimates of about 5% overall in protein databases [4]. Non-Computational annotations are generally believed to be more reliable by the biological community. To see the full gene records for each gene, go to: <http://pir.georgetown.edu/cgi-bin/ipcEntry?id=XX> where “XX” is the IProClass identifier. 86

Table 5.3: Worst-case (all relevant results at the bottom of the output list), Random (relevant results dispersed randomly in the output list), and average precision with default parameters of the BioMiner system for the twenty queries at 100% recall (@100) and at 25 results (@25). The “Relevant” column refers to the number of plausible functions assigned by PIR. The “Results” column refers to the number of functions in the results list produced by the system. “WC” refers to worst-case, “RND” refers to random-case, and “DEF” refers to default average precision. ... 103

Table 5.4: Plausible new functions for four genes from the original sensitivity analysis dataset of 20 genes. These were found by inspecting results from BioMiner and are supported by recent publication evidence. 108

Table 5.5: Rankings of new functions (from Table 5.1) by BioMiner (Reliability Rank), and two deterministic methods (In-Edge, and Path-Count). In most cases, the new functions are ranked much higher by BioMiner than by the deterministic methods. Since they are ranked much higher in BioMiner it is much more likely that a human user will discover them using BioMiner than with deterministic approaches. 109

Table 6.1: List of BioMiner system-based annotations in *S. oneidensis* deemed superior or inferior to the manually produced ones. In most of these cases the tool-based annotations suggested an additional function, the manual annotations could not be ruled out. Two cases of equivalent annotations are provided as well for illustrative purposes. 122

Table 6.2: Evaluation metrics describing the annotation Accuracy (ACC) and Conditional Accuracy (CACC) for 12 common protein databases and for the BioMiner system, both untrained (Defaults), and trained (Optimized). The trained BioMiner system outperforms all other protein databases in regards to ACC and all but TIGRFAM(-Pfam) and PIRSF in terms of CACC (see “Ranks”). The untrained BioMiner still outperforms all databases in terms of ACC (although only by one “Agree”), but is ranked fourth in CACC (highlighted cells). The difference

in ACC and CACC for the trained and untrained BioMiner illustrate the improvements gained through training the system.....	132
Table 6.3: McNemar tests between the trained BioMiner system and all other protein databases. The purpose of the McNemar test is to determine differences in the ability of two databases to produce annotations which agree with the gold standard. P-values in all cases indicate significant difference between BioMiner and most other protein databases, the exceptions being TIGRFAM and CDD.	133
Table 6.4: Annotation results from the trained BioMiner system for the first gold-standard reference set of 30 proteins in <i>S. oneidensis</i> in terms of Agree(1), Disagree(0), or Inteterminate(-1) with the gold-standard. Gold-standard functions for these proteins can be found in Appendix A.	134
Table 6.5: Evaluation metrics for a subset of common protein databases from section 6.5.5 and for the trained BioMiner system on the new gold-standard set of 38 proteins. Metrics are described using the same accuracy metrics as in section 6.5.5 (Accuracy (ACC), and Conditional Accuracy (CACC)). Note the improvement between the trained BioMiner system here and the untrained system in Table 6.2.	136
Table 6.6: McNemar tests of BioMiner (trained using conditional accuracy results from <i>S. oneidensis</i>) versus the protein databases. Results are for 38 new gold-standard annotations (in 5 organisms). P-values are significant in all cases.	136
Table 6.7: Annotation results produced by the trained BioMiner system for the 38 new gold standard proteins. In this result set, the top-hits in BioMiner originate from five different databases (Pfam = 4, Protein Clusters (PRK) = 8, TIGRFAM = 9, PIRSF = 4, and cdd = 1).	137
Table A.1: Annotation dataset used in Chapter 6, section 6.3.3. These proteins were not annotated using our method described in Appendix B.	163
Table A.2: Annotation dataset used in Chapter 6, section 6.5.5. These proteins were annotated using our method described in Appendix B. All proteins come from a single organism.	164
Table A.3: Annotation dataset used in Chapter 6, section 6.5.6. All proteins were annotated using our method described in Appendix B. Proteins in this dataset originate from five organisms.	165

GLOSSARY

ANNOTATION: Assigning functional information to genes and proteins.

ANNOTATION ACCURACY: The proportion of correct annotations produced by a computational protein annotation system over the number submitted to the system.

ASSAY: A scientific experiment.

AVERAGE PRECISION: An evaluation metric which emphasizes returning relevant documents earlier in a list of retrieved documents.

BAYESIAN NETWORK: Probabilistic graphical model that represents a set of variables and their dependencies. A possible representation is the probabilistic relationship between diseases and their symptoms.

COMMON DATA MODEL: A formal definition of all the data entities and relationships between data entities in a domain of interest.

COMPUTATIONAL PROTEIN ANNOTATION: Assigning function to proteins using prediction methods, such as sequence similarity.

CURATE: Create or maintain the annotation for a given gene or protein.

CURATED DATABASE: Refers to a database of proteins which have been carefully annotated by researchers, or “curators”.

DATA INTEGRATION SYSTEM: A system to for combining data residing at different sources with different semantics and providing a consisting and unified view of this data to the user.

DATA SOURCE CATALOG: The data sources incorporated into a database federation.

DATA REPRESENTATION UNCERTAINTIES: Inconsistencies in how conceptually similar data can be represented or modeled by different groups.

DATA WAREHOUSE: A centralized repository of data with query capabilities and common semantics.

DATABASE FEDERATION: Consists of components databases that remain autonomous, as opposed to a data warehouse. The databases are interconnected by a network and appear as a single database.

EUKARYOTE: Organisms with cells that contain a defined nucleus.

EXPERIMENTAL PROTEIN ANNOTATION: Assigning function to proteins via “wet-lab” (biological) experimentation.

EXPRESSION: The process by which genes are made into a functional product, or protein, measurable by certain types of assays.

FEDERATED DATABASE: An integrated resource of multiple, heterogeneous databases.

GENE: A segment of DNA which codes for a protein.

GENE ONTOLOGY EVIDENCE CODE: Three-letter classification for the type of supporting evidence for a function.

GENOME PROJECT: Effort to sequence the DNA of an organism.

GLOBAL SCHEMA: A common data model, or “schema”.

GOLD-STANDARD ANNOTATION: An annotation assigned with the utmost degree of certainty, generally involves “wet-lab” experimentation.

HYPOTHETICAL PROTEIN: A protein of unknown function.

INFORMATION RECALL: The proportion of documents relevant to a given query that are retrieved from a corpus of information.

INHERENT DATA UNCERTAINTIES: Uncertainties associated to a particular data item, such as the error-rate for a given biological experiment.

MACRO-AVERAGE PRECISION: The mean of the individual average precision of each query. Also known as *mean average precision*.

MEDIATED SCHEMA: A centralized schema used in data integration contexts. Also called a common data model.

NETWORK RELIABILITY THEORY: A probabilistic graphical model in which the probability of a connection between two nodes in a graph can be calculated.

PATHWAY: A series of biochemical reactions.

PRECISION: The fraction of documents retrieved that are relevant to the user.

PROBABILISTIC NETWORK: A graphical model with associated probabilities on nodes and edges. Also called a probabilistic graphical model.

PROKARYOTE: A cell lacking a true nucleus, usually a bacterium.

PROTEIN: Large organic compounds made of amino acids which perform various functions in organisms.

PROTEIN DATABASE: A repository of information related to proteins.

PROTEIN DOMAIN: A particular subsequence, common to multiple proteins, with an associated function.

PROTEIN FAMILY: A group of proteins often sharing the same function.

PROTEIN INTERACTION: An instance of two or more proteins “binding” to one another, forming a larger protein *complex*.

Ps: A user-defined degree of confidence in a particular data source.

Pr: A user-defined degree of confidence in a data record in a particular data source.

Qs: A user-defined degree of confidence in the relationship between two sources.

Qr: A user-defined degree of confidence in the relationship between two particular data records.

RECALL: The fraction of documents relevant to the user which are retrieved.

RELEVANCE SCORE: A probabilistic score indicating the degree of relevance a given node in a result set is related to the query node.

SEMI-STRUCTURED DATA: Data organized in semantic entities which may have looser constraints than more traditional relational data. Similar to XML but predates it.

SENSITIVITY ANALYSIS: Modifying the numerical values of a probabilistic model to evaluate their effects on results from the model.

THESIS: Either a master's thesis or a doctoral dissertation. In this document, thesis or dissertation refers to a doctoral dissertation.

UNCERTAINTY METRICS: In the UII data integration system, these provide a probabilistic framework for representing uncertainty in data, data sources, and the relationships between them (Ps, Qs, Pr, Qr). They are sometimes referred to as *parameters*. Uncertainty metrics are used to calculate *relevance scores*.

WRAPPER: An external interface to a database.

ACKNOWLEDGEMENTS

The author wishes to express his deep appreciation to the following groups and individuals for making this dissertation possible:

- To Peter Tarczy-Hornoch, Dan Suciu, and Eugene Kolker for their mentorship and support along this journey.
- To the Biomedical Data Integration and Analysis (Bio-DIAG) group for their software support and comraderie: Peter Mork, Ron Shaker, Todd Detwiler, Wolfgang Gatterbauer, Ethion Cadag, Sophia Jeng, Terry Shen, and Janos Barbero.
- To the folks at BIATECH for their invaluable guidance: Roger Higdon, Jared Roach, Natali Kolker, and Gerald van Belle.
- To my family, and the Good Lord, who provide my motivation and support.
- To Elizabeth Stewart who first got me started in this field.
- To the Biomedical and Health Informatics Program for giving me a chance.
- To the National Library of Medine, the National Institutes of Health, and the National Science Foundation for funding me.

Chapter 1: INTRODUCTION

1.1 Biological Background and Significance

There is much excitement about genes and proteins these days in regards to their potential to address serious challenges in human health from an individual or global perspective. But what are genes exactly? We should begin by discussing what a genome is. A genome is the blueprint for making an organism (Figure 1.1).

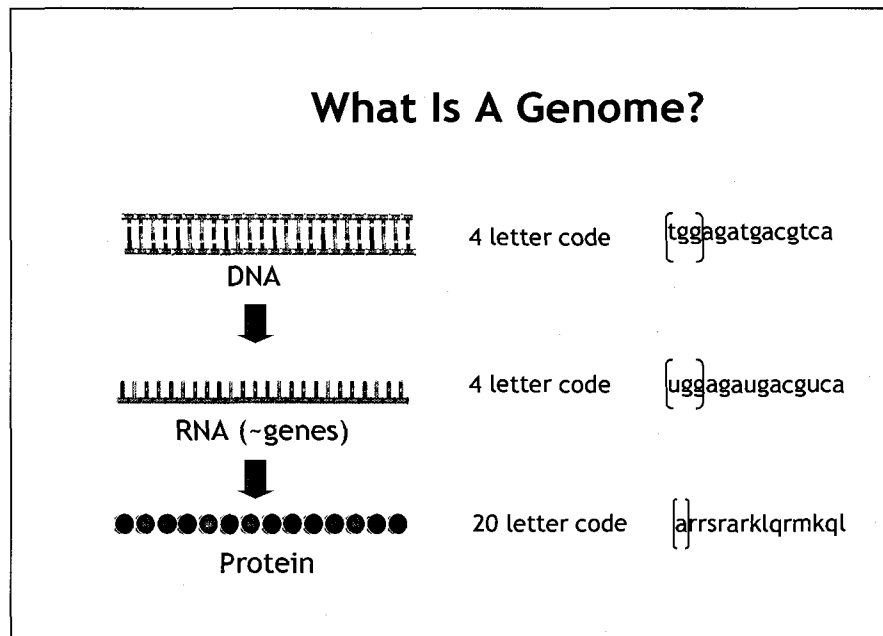


Figure 1.1: A genome is the blueprint for creating an organism. It is encoded in DNA, a base-four sequence of nucleic acids (A, T, G, and C). Within a genome are specific subsequences called genes. A gene is a template which encodes a base-twenty sequence of amino acids, or a protein. Proteins carry out all the biological processes in an organism. The term gene and protein are sometimes used interchangeably in the biomedical literature. This is understandable given the “gene-codes-for-protein” relationship, although this is confusing to non-experts. In this dissertation, the terms gene and protein are generally interpreted as synonyms.

relationship, although this is confusing to non-experts. In this dissertation, the terms gene and protein are generally interpreted as synonyms.

Within a genome are discrete units called genes. There can be thousands to tens of thousands of genes in a particular genome. These serve as templates for the creation of proteins. Proteins are what you can see and touch. They also carry out the necessary biochemical processes and functions for life. Understanding the role of proteins is important. Many diseases are the result of improper protein function, such as apoptosis (or programmed cell death) in cancer. Additionally, novel biological pathways in bacteria may someday be utilized for such things as clean-up of radioactive waste (Figure 1.2).

So how are genes, and genomes, studied? Another way to ask this is: where does the data come from? The DNA sequence of an organism is generally determined by government-funded genome projects, of which there are over 2000+ completed and ongoing. As of January, 2007 there were over 65 billion bases and over 61 million sequence records in GenBank [1]. Within these genome sequences, new genes (and proteins) are being discovered at an astounding rate [2]. With this encouraging story comes a reality-check however. It is not simply enough to discover new proteins. To achieve the aforementioned human health benefits we must find out what these proteins do. In other words, what biological function is performed by each of these proteins? The process of determining the biological function of proteins is known as protein annotation. Unfortunately, the story here is not so encouraging. The functions of a

large percentage of proteins are unknown. These are otherwise known as hypothetical proteins.

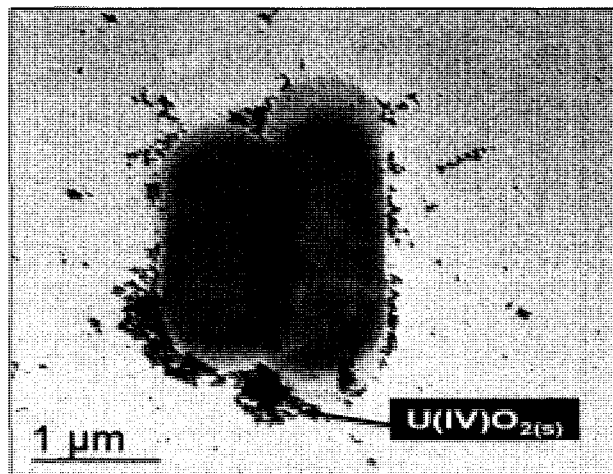


Figure 1.2: Novel proteins in the bacterium *S. oneidensis* allow it to “breathe” hexavalent uranium. This converts hexavalent uranium to uraninite which is insoluble in water and much easier to clean-up (courtesy of Eugene Kolker, PhD).

1.1.1 Hypothetical proteins

What are hypothetical proteins? Before this question is addressed, a distinction should be made between experimental and computational protein annotation. Experimental protein annotation is the usage of “wet-lab” techniques for characterizing the function of proteins. These techniques range from biochemical assays to mutation studies and they all generate new “empirical” data, unlike in computational approaches [3].

Computational protein annotation is the use of computer algorithms to predict protein function [4]. The simplest computational method is to determine the similarity between the amino acid sequence of a protein of unknown function and that of protein of known function (i.e. simple sequence comparison). Hypothetical proteins are therefore

proteins which have not been experimentally characterized and whose functions cannot be deduced by simple sequence comparisons [5]. In regards to annotation, experimental characterization is preferred but infeasible due to its limited efficiency and high cost. Computational annotation is thus by far the norm.

1.1.2 Hypothetical proteins impede biological research

Hypothetical proteins are extremely prevalent. For any sequenced genome, approximated 30-50% of proteins remain hypothetical after the first attempts are made to annotate them [6]. Addressing the issue of annotating these hypothetical proteins is of primary importance in biological research today [7, 8]. This lack of knowledge regarding protein function is an impediment to biological research. For instance, in addition to our previous examples, antibiotic development seeks to target bacterial genes which perform vital functions in bacteria but not in humans [9]. A similar approach is taken with cancer [10]. New “systems biology” approaches seek to determine the interaction among network of genes to better understand biological processes [11]. All of these areas of research depend on knowledge of protein function to varying degrees.

1.1.3 Improvements needed for hypothetical protein annotation

Experimental characterization of protein function is ideal and necessary but is costly and cannot keep pace with the rate that new proteins are being discovered.

Computational approaches exist which predict protein function and are needed to fill the gap but there is much room for improvement as computational approaches are

idiosyncratic and none work well in all cases. Utilizing more types of data and taking into account function predictions from multiple approaches are promising approaches [12], although not without its challenges, especially in regard to integrating the data. Biological data is fragmented in dispersed and heterogeneous data sources, and the problem is getting worse as the number of sources continues to increase (Figure 1.3). New techniques in data integration research could possibly alleviate this issue and improve the computational annotation process. This is the approach taken in this dissertation.

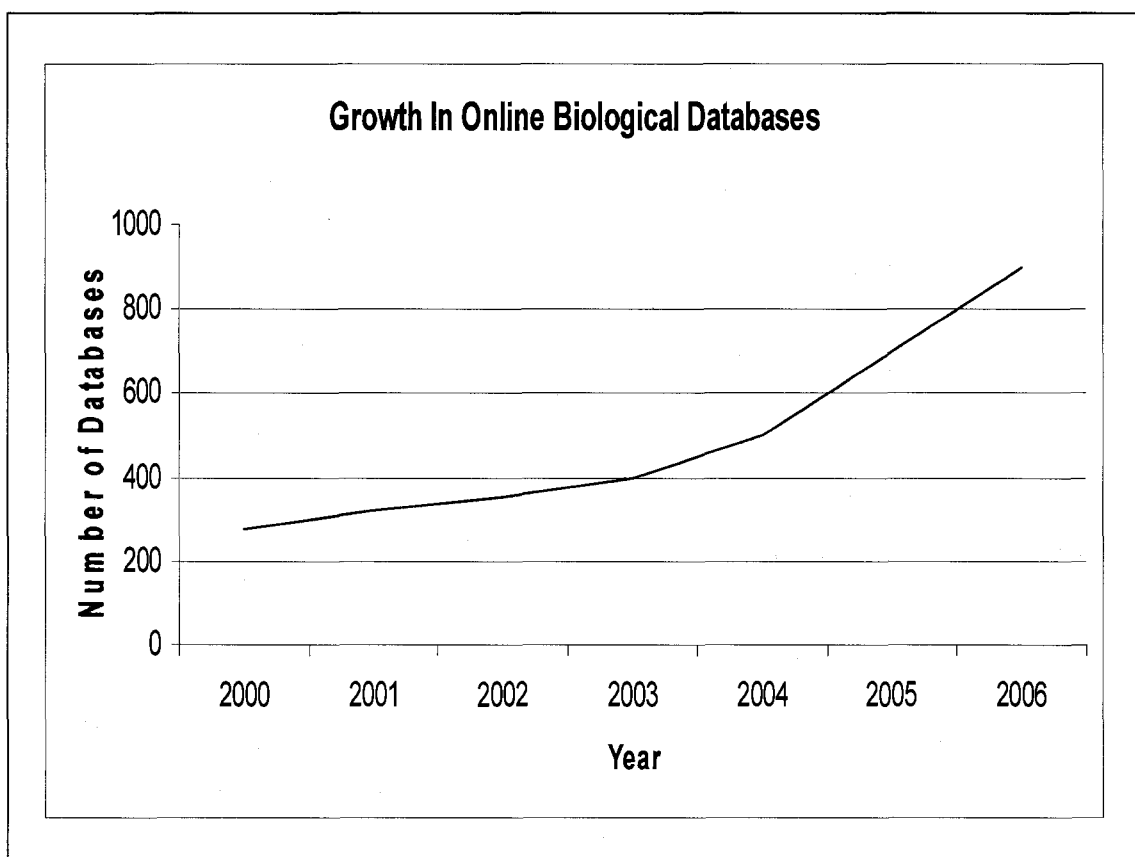


Figure 1.3: The number of biological databases has continued to increase over time, creating a significant data integration challenge in computational protein annotation (courtesy of [12]).

1.2 Research Questions

The general theme of this dissertation revolves around the utilization of cutting-edge data integration techniques for improving protein annotation. The focus is on evaluating the value of incorporating functionality to handle uncertainty in data into a data integration system.

1.2.1 “How well does computational modeling of the uncertainty in the annotation process improve systems for computational annotation of proteins?”

This is the overall research question of this dissertation. Biological data contains inherent uncertainties. Modeling this uncertainty in a data integration system can enable such things as ranking and highlighting of the most “certain” or “relevant” data. How can we model this uncertainty and does this improve systems for computational protein annotation? This is a broad question with many facets. First, there is the question of what is involved in developing and evolving a system for computational modeling of uncertainty in the annotation process. This is addressed in Chapter 4, which discusses the development of the BioMiner system. In addition, two other sub-questions of this overall question, which are addressed in this dissertation, are discussed in the next two sections.

1.2.2 “How robust is our model of uncertainty?”

Uncertainty modeling involves determination of various probabilistic measures (or *parameters*). How are these measures determined? More importantly, how precise do these measures need to be? If the output from our system varies considerably under minor changes in its probabilistic measures then much care must be taken to determine them precisely. If output from the system is “insensitive” (i.e. robust) to minor variations in its probabilistic measures we can have greater confidence in our overall uncertainty model. These questions are addressed by our systematic analysis of the BioMiner system in Chapter 5.

1.2.3 “How does a system based on our model perform versus existing methods?”

Our system for annotating proteins is based on formal data integration techniques and cutting-edge technology for handling uncertainty in data. Does this approach improve upon commonly used existing computational methods for annotating proteins? This is a “real-world” application question. We address this by performance an evaluation of the BioMiner system which is discussed in Chapter 6.

1.3 Related Work

Given the broad nature of this dissertation, related work sections have been included in subsequent chapters. Current approaches for computational protein annotation and challenges facing computational annotation systems are described in Chapter 2. Related work regarding general-purpose data integration systems as well as handling uncertainty

can be found in Chapter 3. Related work using probabilistic graphical algorithms for assigning function to proteins are discussed in Chapter 4. Related work on sensitivity analysis in probabilistic networks can be found in Chapter 5. Finally, limitations of previous evaluation studies of computational annotation systems are discussed in Chapter 6.

1.4 Contributions of this Dissertation

This dissertation describes the creation and evaluation of a probabilistic system for computational protein annotation. The main contributions of this dissertation are:

- The implementation of a novel uncertainty model for computational protein annotation, known as the BioMiner system (Chapter 4, section 4.5). This uncertainty model is the representation the annotation process usually carried out manually by biological experts and is the first implementation of cutting-edge data integration methods, discussed in Chapter 3, section 3.4. This implementation of an uncertainty model allows it to reduce large datasets in a distinctive way: by ranking or highlighting integrated data based on its relevance to a user query. The uncertainty model is also validated in Chapter 5, section 5.5.
- A demonstration of the robustness of the uncertainty model by a principled methodology (Chapter 5, section 5.3). Our results in this regard both ease the burden of choosing parameters in the model and boosts our confidence the results the uncertainty model produces (Chapter 6, section 6.5).

- An evaluation of our uncertainty model for annotation in which we demonstrate improvement in annotation accuracy over standard, commonly utilized, approaches (Chapter 6, section 6.5).

1.5 Outline of this dissertation

This dissertation is organized in the following manner. Chapter 2 provides an introduction to protein annotation, its general methods, and the challenges it faces. Chapter 3 is an overview of general data integration methods which are relevant to biological data integration. It also includes a section on incorporating uncertainty functionality into general-purpose data integration systems. Chapter 4 discusses the development of BioMiner, a system for annotation which utilizes methods described in Chapter 3. Chapter 5 is a validation and evaluation of the robustness of the uncertainty model in BioMiner (Chapter 4). In this evaluation, we outline a general and principled methodology for determining the robustness of the uncertainty model. Finally, Chapter 6 evaluates BioMiner (Chapter 4) versus existing systems for annotation. Additionally, Chapter 6 describes two “gold-standard” annotation data sets and our approach for creating them, allowing for rigorous evaluation of the function predictions of various systems. This dissertation concludes with a summary of the lessons learned while performing this work as well as limitations which suggest possible future directions.

Chapter 2: ASSIGNING FUNCTION TO PROTEINS

2.1 Background: Protein Annotation

The blueprint for life is encoded in each organism's DNA, or genome. Within a genome are smaller discrete units called genes. Genes can be seen as a template for proteins which carry out all biological roles, such as providing structure (e.g. muscles), or facilitating chemical reactions (e.g. enzymes). For example, hexokinase is a protein encoded by the HK1 gene. It performs a specific chemical reaction in the glycolysis pathway which converts sugar into ATP, a form of energy usable by cells in the human body (Figure 2.1).

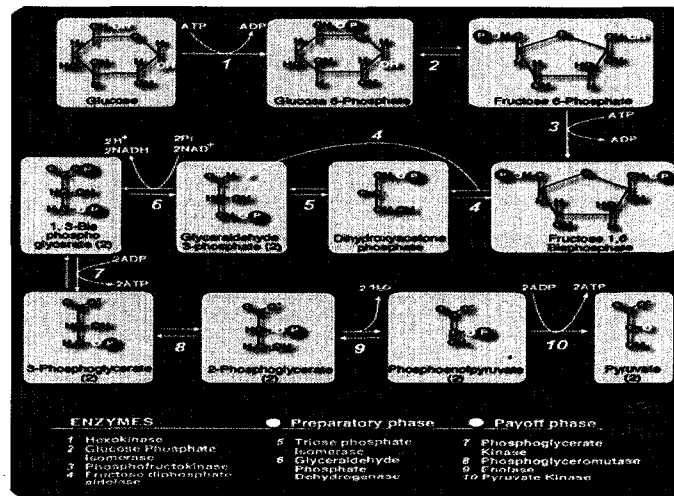


Figure 2.1: The glycolysis pathway, which provides an illustration of protein function. The hexokinase protein is an enzyme which catalyzes the first reaction of the process by which the human body converts sugar to ATP, a form of energy usable by cells (www.biocarta.com).

There is more to this story in regards to human health however. As it turns out, cancer cells appear to utilize the glycolysis pathway to a greater extent than normal (non-cancer) cells, which may open the door to new treatments [13]. As you can see, having this sort of biological knowledge is quite useful. Unfortunately, the biological functions and roles of many proteins are unknown. Worse yet, many thousands of proteins of unknown function are being discovered at an extremely fast rate [14]. This problem, otherwise known as protein annotation, is a fundamental challenge in modern biology [7, 8]. While the protein annotation problem is being addressed to a certain extent, the rate at which it is happening is unacceptably slow [5]. The poor productivity of protein annotation is due, in part, to the fact that the process is highly manual, data intensive, and error-prone. While these are difficult challenges, computational approaches, such as methods adapted from methodologies in data integration research, could possibly alleviate these issues and improve the process of protein annotation. They are discussed in this chapter.

2.1.1 Discovery of new proteins

Genome sequencing efforts are generally government-funded projects which set out to determine the genome sequence of a particular organism. The amount of data they generate is staggering. The Genomes OnLine Database is currently tracking over 3500+ genome sequencing projects world-wide [15]. This amounts to over 80 billion DNA bases, which in turn accounts for an estimated 80 million genes. While sequencing and locating genes is a high-throughput endeavor, the bottleneck is in determining the biological function of proteins encoded by those genes. In fact, within

the genome sequence of any organism approximately 30-50% of proteins are annotated as “hypothetical proteins” (unknown function) [6] (Table 2.1). This large proportion of hypothetical proteins in any organism is a serious impediment which must be addressed in order to advance biology into a more “predictive” science [5].

Table 2.1: The percentage of hypothetical genes (i.e. of unknown function) in selected organisms. The percentage of genes with predicted function is only an estimate, as less than 5% of annotations are known to be experimentally derived [16]. In practice, it is often difficult to the true percentages of computational and experimental annotations as this information is not often stored in protein database records. This table is adapted and abbreviated from [17].

Organism	Genome size (megabases)	Genes coding for proteins	% Genes annotated as hypothetical	% Genes with predicted functions
H. sapiens	3200	20-30,000	50%	45%
S. cerevisiae	12.5	6000	55%	40%
H. influenzae	1.8	1750	37%	58%
C. elegans	97	19,100	48%	47%
L. major	33	8213	64%	31%
T. cruzi	60	25,041	66%	29%
T. bruci	35	10,689	66%	29%
P. falciparum	25	5279	61%	34%
P. yoelii	25	5878	63%	32%

2.1.2 Disparate data sources regarding protein function

Finding information about what is currently known regarding the function of proteins or predicting function for hypothetical proteins is highly dependant on multiple and heterogeneous types of data which is generated by research labs or consortiums and

made available over the internet. Many biological researchers query and traverse these data sources and compile information manually which is cumbersome, ad-hoc, and error-prone (Figure 2.2).

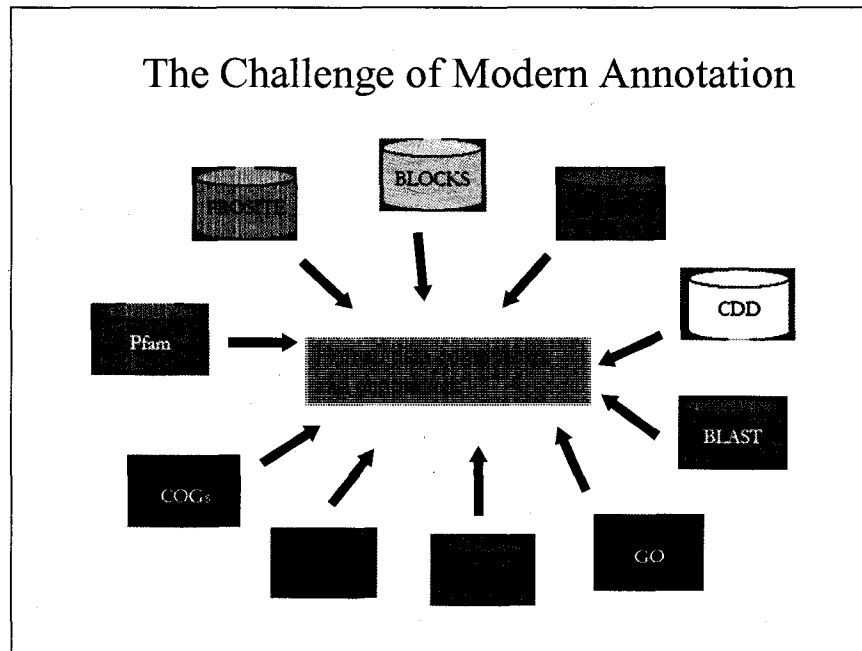


Figure 2.2: To annotate proteins in the current paradigm, biological researchers manually query and compile data and information from multiple, non-interoperable biological data sources which greatly slows the pace of annotation. (Figure adapted and modified from [12]).

It is difficult to create systems which automate the process of integrating annotation data as annotation data sources are often generated independently and can therefore exhibit idiosyncratic user-interfaces, data models, and data types. There are also many hundreds of data sources to choose from [18]. These sources can include databases of previously annotated proteins [19], protein family descriptions [20, 21], and protein interactions [22, 23] or expression [24, 25]. This heterogeneous nature forces biological researchers to have intimate knowledge of the data model and query capabilities of each source. Consequently, researchers may not search all possible

sources or sources may not be queried in a consistent fashion. The quality of protein annotations produced with this manually compiled data may suffer as a result.

2.1.3 Data integration for improving protein annotation

Managing all of this biological information has been described as trying to “swim in a sea of data” [26]. Computational approaches have been attempted to alleviate this “data overload” but none have sufficiently addressed the problem, at least from a global perspective [27]. Nonetheless, new approaches in data integration research may ultimately make a new paradigm possible. In this new paradigm, a biological researcher makes a single query into a type of data integration “middleware” which handles the querying and compiling of results from multiple independent databases (Figure 2.3).

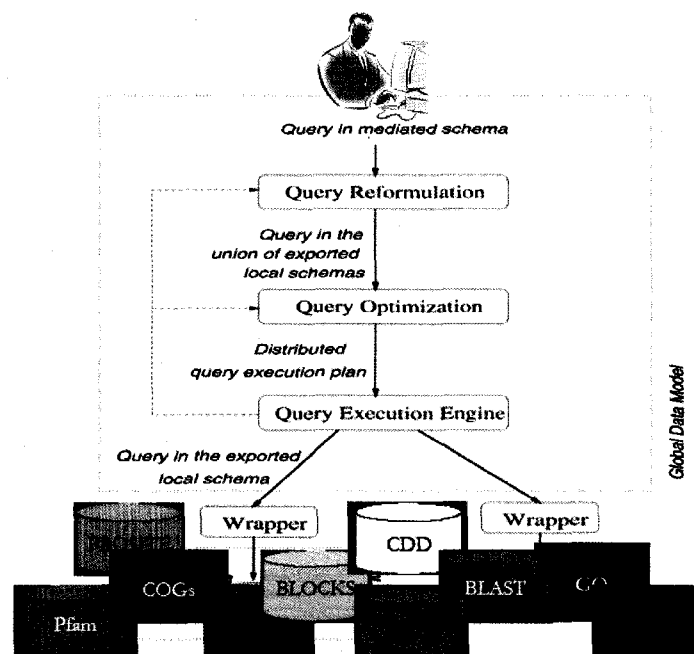


Figure 2.3: A new paradigm for annotation using data integration. Users query software which handles the specific querying and compiling of results from the individual data sources. To the user, all the individual data sources appear as a single database (adapted from [28]).

Biological researchers can now focus on “what” they want rather than “how” to get it [28]. This approach would alleviate a time-consuming step in the annotation process and also lets biological researchers focus more on assigning function to proteins using the integrated data. Moreover, a data integration system could provide more accurate and comprehensive data sets through consistent querying and more exhaustive data searches, potentially benefiting the quality of protein annotations.

2.2 Related Work: Current Approaches In Protein Annotation

How are proteins actually annotated? A key distinction here is the difference between experimental and computational approaches. Experimental approaches involve biological “wet-lab” procedures, the result of which is the generation of new empirical data about a particular protein. Computational approaches can be seen more as annotation “predictions” and they draw heavily from the empirical data generated from the experimental approaches. These work best when human experts inspect and curate their results, although this is not often the case as many computationally-produced annotations have no human involvement, such as in the TrEMBL protein database [29]. To summarize, experimental approaches produce the best annotations. Computational annotations alone are generally understood to not be as reliable but can be improved by expert intervention. Completely automated annotations are the least preferable. Computational annotations however, whether humans are involved or not, are becoming increasingly necessary given the large and growing number of hypothetical proteins and the cost of biological experimentation.

2.2.1 Experimental versus computational protein annotation

Experimental protein annotation involves actual “wet-lab” experiments to characterize the function of a particular protein, or proteins. These experiments can be biochemical assays, which involves observation of a specific chemical reaction, or mutation experiments where a gene is “deleted” in an organism (such as a bacteria) and the effect is observed (such as the loss of the organisms ability to digest sugar for instance).

Experimental characterization is accepted as the gold-standard for protein annotation. It is limited however by its high expense and low-throughput capacity. There is simply no existing, cost-effective, way to experimentally characterize all existing proteins.

Fortunately, this is where computational protein annotation comes in. Most protein annotations are computational as opposed to experimental (Figure 2.4). Contrary to its name, computational protein annotation still requires involvement by users, at least to ensure that obvious annotation errors are avoided. Computational protein annotation is based on what can be called “annotation transfer”. Proteins can sometimes exhibit similar amino acid compositions. If the amino acid sequences of two proteins are similar, they often perform the same function. In the idealized case, a hypothetical protein would be highly similar to an experimentally characterized one – thus transferring the annotation to the hypothetical protein. The similarity between proteins is determined by algorithms which search databases of proteins, most notably BLAST [30]. More advanced methods for determining similarity are based on comparisons using multiple proteins include Hidden Markov Models [20, 21, 31] and RPS-BLAST [32]. Note that all computational methods are dependant on the small

core of experimentally determined protein annotations, which is estimated to be less than 5% of the total. Computational protein annotation can thus be seen as one large “extrapolation” exercise [4].

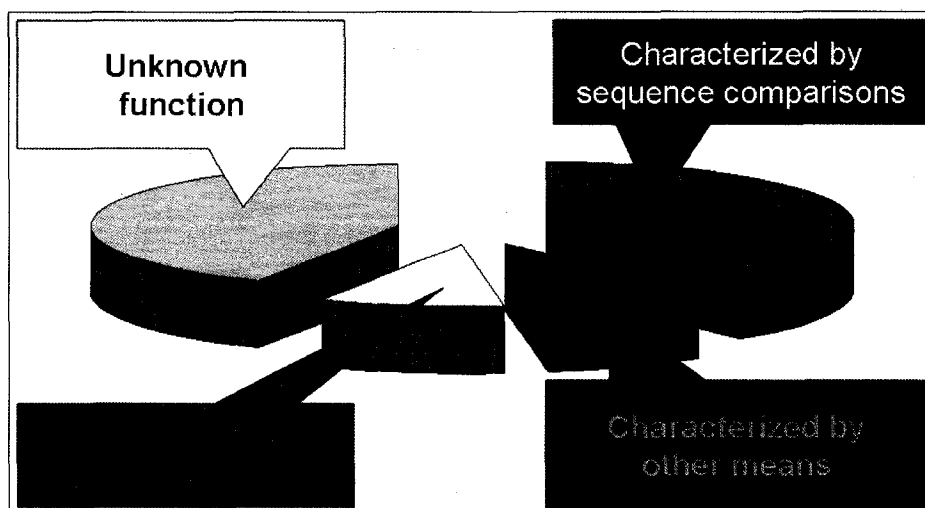


Figure 2.4: An illustration of the proportions of various types of annotations in the organism *Shewanella oneidensis*. Only a small percentage of annotations can be ascribed to actual experiments (about 5%). Most proteins have been annotated by computational means (sequence comparisons) or are of unknown function (Courtesy of Eugene Kolker, PhD).

2.2.2 Computational protein annotation

Computational protein annotation, somewhat contrary to its name, works best when it is performed in conjunction with biological experts. A biological domain expert is generally able to fix obvious errors and resolve ambiguous evidence, such as when the predictions of different computational annotation algorithms don’t agree [33]. They can thoroughly inspect the alignment between two proteins and discern whether or not they are indeed related and thus perform similar functions. This can require deep biological knowledge about particular families of proteins, such as the locations of amino acids critical for functionality. Biological knowledge such as this is not easily encoded in a

computational representation. Biological experts can also evaluate recently published functional evidence, when available, on a protein and gain deeper overall insight into its function. Given the limitations in natural-language processing, computers have a difficult time extracting this kind of information from publications. These attributes can make manual annotation quite accurate, depending on the skill of the annotator.

This accuracy comes at the obvious cost of efficiency [5], making the completely manual annotation approach only feasible for small laboratories studying a single gene or family of related genes. Biologists also tend to apply ad-hoc annotation criteria, which is not often recorded with the annotation. This makes it difficult to assess the quality of the annotation at a later date. Protein function is also very diverse, and any one biologist may only be able to expertly assess the annotations for proteins in their particular area of focus. In addition, since biologists search for information manually, they may only search a couple of familiar data sources for information regarding a protein's function, although there are many to choose from [18, 34]. There is a distinct possibility that important functional information about a protein can be missed given that all data sources are not exhaustively searched. This calls into question the true overall accuracy of manually produced annotations. Biologists may be able to accurately discern functions of proteins given the evidence they have, but if they do not have all the evidence, annotations they produce still may not be *reliable*. This reliability question could potentially be addressed by using methods which more

exhaustively search all available data. This leads us to systems for computationally annotating proteins.

2.2.3 Systems for Computational protein annotation

Systems for computational protein annotation seek to improve computational annotation by providing more integrated views of necessary data. Some of the most well-known systems are: FANTOM [35], Ensembl [36], GeneQuiz [37], BioMediator [12], CDD [32], and InterPro [38]. Most systems of this type follow a similar three-tiered approach of a database, which contains functional models of proteins, a way to query the data, and a user interface (Figure 2.5). Their goal is to assist human users by supplying more integrated views of the data. Some systems of this type are created for a specific time and purpose to coordinate the concurrent efforts of multiple users, such as FANTOM for the annotation of the mouse genome [35]. The Ensembl database, BioMediator, and GeneQuiz, are built to be a general-purpose annotation tools which biologists can utilize for their own purposes, although the amount of computer literacy necessary to accomplish this can be quite high. BioMiner, the computational annotation system developed in this dissertation, falls into this class.

CDD and InterPro can be seen as computational annotation “web-services” which are widely available and supported by the biological community. These are sometimes called either protein or “pattern” databases, such as CDD or InterPro [34]. These pattern databases create computational models of protein families (protein sequences of similar function). The way they work is rather simple. The user provides a protein sequence as input which is searched against an internal database of protein

family models. If the input sequence is deemed to be similar to a protein family model, the function of the protein family model is output as a function “prediction” for the input protein. They can be extremely advantageous in that some biological knowledge about a diverse array of protein families is represented in their models. Thus, a user does not necessarily need to be a biological domain expert. These pattern databases are becoming an increasingly common way for individual biological researchers to computationally annotate proteins.

These systems for computational protein annotation are the way of the future given the need for annotation and the scale of the problem. They are not without their problems however. Challenges these systems face, such as necessary improvements regarding managing data and annotation accuracy, are discussed in the following sections.

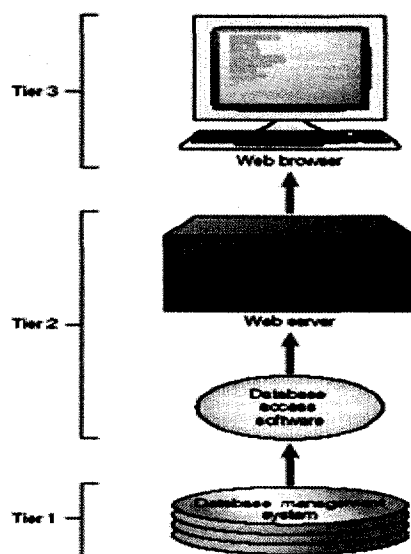


Figure 2.5: Systems for computational protein annotation generally follow this three-tiered approach. Tier one is a data-management system, Tier 2 includes a way to query the data, and Tier 3 provides views of the data. This figure is courtesy of [27].

2.3 Challenges for Computational Annotation Systems

Creating a general-purpose system for computational protein annotation is challenging. Of utmost importance, the primary goal of the system is to enable human users to produce annotations faster and more accurately than completely manual approaches. To achieve this primary goal, the system needs to utilize appropriate data management techniques as well display results in a usable fashion. There are three challenges in particular these systems for computational annotation face and which are addressed in this dissertation: 1) flexible data integration, 2) handling uncertainty in data, and 3) accuracy of annotations.

Many new techniques in data integration have been developed to address these sorts of issues and could be extremely beneficial when utilized by systems for computational protein annotation. They have not however, been systematically applied to and evaluated for this problem. This section discusses the challenges and limitations faced by existing systems for computational protein annotation.

2.3.1 Data integration in computational annotation systems

Data integration is the primary challenge for computational annotation systems.

Biological data sources are continually emerging [18] and many exhibit a high degree of data turnover. Also, each biological data source only contains a particular subset of the available biological information. For a data integration system to perform well in terms of information recall (i.e. the amount of relevant information retrieved), enough biological data sources need to be integrated. Overall, it is very difficult for a data integration system to be flexible in regard to the sources it integrates as well as be

continually up-to-date. Data warehouses are a very common approach, and are employed by virtually all of the systems described here such as FANTOM, Ensembl, and GeneQuiz, as well as the annotation web-services such as InterPro. An exception is BioMediator which is a database federation. Data warehouse approaches have been attempted with limited success, with the Integrated Genome Database (IGD) being one notable failure in particular [27]. The main reason cited for the failure of the IGD was its inability to evolve its data model quickly enough. Data integration challenges for protein annotation and methodologies to address them are discussed in more detail in Chapter 3. In addition, there is an aspect of biological data which presents a data integration challenge. Unlike more traditional sources of data (as in banking or inventory), biological data carries with it inherent uncertainty. Traditional data integration technology does not handle uncertainty in the data well, which severely affects their utility in regards to annotation.

2.3.2 Handling uncertainty in computational annotation systems

The UII project is an NSF funded research project, the primary focus of which is to develop formal frameworks and methodologies for handling uncertainty in data integration systems (Chapter 3, section 3.4). My role in UII is participation with the team developing the formal framework and methodologies as well as being the primary person to pursue, develop, and evaluate applications for the new technology, such as using it for protein annotation. The following content discusses the nature of uncertainty in biomedical data and provides the rationale for the primary focus of the UII project. Note that none of the other data intergration and annotation systems in this

chapter handle explicitly handle uncertainty in data in a formal manner. Data integration systems which do not handle uncertainty explicitly face daunting challenges when integrating biomedical data such as result set “explosion”. This is discussed in more detail in Chapter 3.

Uncertainty is prevalent in biological data and databases and takes many forms. For the purposes of illustration, we have broadly classified this uncertainty into two main categories (adapted from [39]):

- 1) Inherent data uncertainties. These are attributes of biological data itself, and not of its representation. For instance, biological data generated from laboratory experiments is inherently uncertain. Some experimental methods, such as protein interaction assays for example, have estimated error rates of up to 50% [40]. Computational prediction “experiments”, such as algorithms which assess similarity between protein amino acid sequences like BLAST or Hidden Markov Models, are inherently probabilistic [20, 30]. They output scores, called e-values, which is a measure of the degree of chance that two protein sequences are functionally related. Finally, uncertainties can be rooted in the ever-evolving nature of biological data. This is illustrated by “status codes” given to reference gene sequences (RefSeq’s) in GenBank [41]. These refer to the amount of expert curation attributed to a particular RefSeq and changes over time as biological knowledge accumulates for particular genes.

2) Data representation uncertainties. These uncertainties are the result of the mapping of real-world information onto a computational representation. At last count, there were literally hundreds of publicly available biological data sources [18]. For all that data however, there is not agreed-upon common data model or even a common biological identifier [27]. This results in a wide variety of heterogeneous representations which are difficult to reconcile, much less integrate. For instance, GenBank uses RefSeq status codes to represent the amount of evidence attributed to a particular gene but the Gene Ontology uses their own proprietary set of “evidence codes” [42]. It is difficult to compare both representations. Most biological data sources are therefore non-interoperable. This means that querying across data sources may require that linkages be determined by such things as inexact string comparisons, which introduces uncertainty into the data integration process.

The challenge here is that data integration systems generally don't handle data uncertainty well. The result is that data integration systems can produce explosions of results to biological queries, if enough data sources are incorporated (Chapter 3, Figure 3.2 and Figure 3.3). If result sets are too large, humans have a difficult time sorting through them which leads to loss of relevant answers. For a data integration system to perform well they need additional functionality to handle uncertainty in data, which is a burgeoning but still highly emergent area of research (Chapter 3, section 3.4).

2.3.3 Annotation accuracy of computational annotation systems

Computational annotation systems address the issue of improving the productivity of annotation, at least partially. Some systems of this type were created for a specific time and purpose, such as FANTOM for the annotation of the mouse genome [35]. This sort of approach is not sustainable in the long term [43]. Other systems, such as Ensembl and GeneQuiz, are based on data warehouses which are difficult for users, such as biological researchers, to implement, populate, and especially maintain. This impacts the accuracy of their annotations over time. A major reason for this is that biological knowledge is never static. New data and information is constantly being released into public databases. This has the effect of annotations becoming “stale” over time and drives a need to constantly re-annotate proteins [44]. What’s still needed are more general and light-weight tools which can facilitate on-going re-annotation of proteins as well as entire genomes [44]. Pattern databases address this problem to some extent but their accuracy can be improved upon as we demonstrate in Chapter 6.

Perhaps a more difficult problem is proper evaluation of these systems for computational annotation. Evaluation studies are necessary to improve the accuracy and consistency of annotations produced by these systems. Previous evaluation studies have simply compared new annotations to existing ones residing in Genbank which may be erroneous or outdated, or only provide lists of new predicted functions without further validation of these predictions, such as in the case of GeneQuiz [45] and SuperFamily [31], another annotation web-service. Other evaluation studies only provide overall estimates of annotation error rates. For instance, consider the case of

two different groups producing different annotations for the same protein. At least one of these must be in error. Using this method, Brenner estimated the overall annotation error rate to be at least 8% [46]. Devos and Valencia on the other hand, estimated that that annotation error rate can be as high as 30% [47]. It is possible that these inconsistencies can be traced to common pitfalls in the annotation process [48]. Computational annotation systems should account for these potential pitfalls if they are to improve upon manual methods, however proper evaluation studies are necessary to determine true improvement in annotation accuracy.

A major reason for the difficulty in evaluating the performance of computational annotation systems is the lack of gold-standard annotation reference sets [44]. In this dissertation however, we describe our approach for creating reliable and independent gold-standard reference annotation sets in prokaryotic genomes (Chapter 6, section 6.5). This was quite useful in that it facilitated the performance evaluation of several computational annotation systems and provided insights into how accuracy could be improved.

2.4 Discussion

Hypothetical proteins are an impediment to research in biology today. Assigning function to them is challenging given the need for manual inspection and the fractured nature of biological data sources needed to annotate them. Methods for partially automating the process have met with limited success, one of the major challenges being integrating and managing the necessary data. One focus of this dissertation is the creation of a novel and cutting edge prototype system for protein annotation: BioMiner.

BioMiner uses the formal frameworks and methodologies for handling uncertainty in data integration systems developed by the UII project (Chapter 3, section 3.4). BioMiner addresses the key limitations of previous systems for computational annotation just discussed, specifically:

- **Data Integration:** BioMiner utilizes database federation technology which alleviates challenges that data warehouses face in regard to integrating biomedical data. Although federated data integration is not a recent invention, it has only been evaluated for protein annotation to a limited extent. This system is described in Chapter 4.
- **Handling uncertainty in data:** BioMiner explicitly models uncertainty in biomedical data, allowing it to rank, highlight, or filter the most relevant and “certain” information. Handling uncertainty in data for the purpose of annotating proteins is a novel contribution of this dissertation. This is also described in Chapter 4.
- **Annotation accuracy:** the data integration and uncertainty capabilities in BioMiner as well as its explicit handling of uncertainty will improve annotation accuracy by enabling the utilization of more and up-to-date information. Benchmarks of annotation quality for computational annotation systems generated by this dissertation are among the first in regard to their ability to annotate hypothetical proteins. These are demonstrated and described in Chapter 6.

In addition, quality benchmarks for computational annotation systems are difficult to carry out given the lack of independent gold-standard annotation reference sets. These types of studies are vital for assessing and improving annotation accuracy. An additional contribution of this dissertation focuses on the creation of gold-standard annotation reference sets for evaluation purposes, which is also discussed in Chapter 6.

Chapter 3: BIOLOGICAL DATA INTEGRATION

3.1 Overview Of Data Integration Concepts

Data integration is fundamentally about querying across different data sources. These data sources could be, but are not limited to relational or semi-structured databases dispersed across a network. Many concepts and methodologies from the general discipline of data integration have recently been applied in the arena of biomedicine [49, 50]. This chapter highlights those technologies most relevant to data integration problems in biomedicine and discusses the strengths and limitations of each. The technologies discussed are fairly stable and can be readily applied to identifiable data integration problems in biomedicine. Much of the content in this chapter is adapted from [51] and [39].

3.1.1 Where data resides: federation versus warehouse

A data warehouse consolidates all specified data into a centralized repository, often with a generalized, global schema (section 3.1.3). They are reliable and generally provide excellent response time to user queries. Data is under local control which facilitates easier cleansing and filtering of the data. Importing all data and housing it in a single repository can be problematic however. The volume of data can be enormous (especially in biomedicine), and diverse data types from various sources can make it

difficult to create a global schema. There are also maintenance issues. Since data is copied from remote sources, data in the warehouse can become stale if the remote sources continually change their content. Given these considerations, data warehouses in biomedicine may be best suited for highly curated databases which focus on a specific area of research, such as identifying the location of genes on chromosomes [52].

A database federation does not consolidate all data into a central repository. Instead, data is left at the source where it is retrieved only when a query is issued. The underlying databases remain autonomous and may be distributed across a network. The federation maintains a common data model (or *mediated schema*) and relies on schema mappings for integration (section 3.1.3). It interacts with its underlying databases via software interfaces, sometimes called “wrappers” (section 3.1.2). To the user, the federation appears as a single (virtual) database. Data in a federation is thus always up-to-date. However, since no data is housed locally, performance can suffer due to network limitations or query loads on member databases. Also, since local control of the data is limited, cleansing or filtering of the data must be done on-the-fly which may be difficult [53]. Federations which utilize a common data model can also face the same difficulties as warehouses in representing diverse data types, although advances in data modeling such as the use of federations with mediated schemas can alleviate this problem (section 3.1.3). Given these considerations, federations may be best suited for situations where the most up-to-date data from a large number of data sources is

required and the creation of a large, centralized data repository is infeasible.

Table 3.1 provides a summary of the strengths and limitations of data warehouses and federations.

Table 3.1: A summary of the strengths and limitations of data warehouses and federations, adapted from [51].

Architecture	Advantages	Disadvantages
Data warehouse	Fast queries Clean data	Stale data Complex schema Extra storage
Database federation	Current data Flexible architecture Less storage	Slower queries Complex schema Little data cleaning
Database federation + Mediated schema	Current data Flexible architecture Schema tailored to users	Slower queries Little data cleaning Mappings needed from source to mediated schema

3.1.2 Data interfaces

Data interfaces, sometimes called wrappers, facilitate the integration of heterogeneous and distributed databases (or data sources such as web pages), into modern distributed (e.g. federated) systems. In general, their purpose is to interact with the source while providing a standard and common interface [54]. Specifically, wrappers translate incoming queries into the syntax of the specific source and format the results from the

source, which may be in a loosely-structured format such as HTML, into a format which can be easily handled by the integration system, such as XML [55]. Wrappers are generally written once but may require additional maintenance if data sources change their query capability or data formats, something not uncommon in the biomedical domain [27].

3.1.3 Common data models

A common data model can be described as a uniform and consolidated view of biomedical data sources (or subset thereof) [56]. Data integration systems may utilize a common data model, which may be called a global or mediated schema [28]. Users pose queries to the mediated schema which alleviates the need to understand and learn the various query capabilities of each source as well as data formats. It can be difficult to create a global schema which properly represents the richness of data from all sources. A mediated schema however, offers more flexibility in that only a desired subset of the data need be modeled. Various mediated schemas may be developed for particular sets of queries and types of data and mediated-schema driven data integration systems may be best suited for queries which span diverse knowledge domains [51].

3.1.4 Query models

The query capabilities of a data integration system depend on the schema language, where most of the focus has been on the relational, or structured, model. In more recent years, especially with the rise of the world-wide-web, more focus has been placed on path-based query models (XQuery) which work on semi-structured data (XML). Both

of these approaches require queries to be explicitly stated, as well as prior knowledge of the schema. Research from the BioMediator group indicates however that scientists in the biomedical domain often have difficulty expressing precise queries, which led to the development of method-based queries [55], an approach akin to browsing the web. The BioMiner system, which is the focus of evaluation in Chapter 5 and Chapter 6, utilizes method-based queries. Much of this section is adapted from [57] which provides a good expanded explanation of query models.

3.2 Related Work: Existing Data Integration Systems

Data integration in biomedicine has generally been an ad-hoc endeavor, with off-the-shelf technology and proprietary interfaces being written for a particular project or specific purpose. Ensembl, for example, is an data integration system and user interface for viewing genome sequence data [36]. Its strength lies in its highly tailored functionality, but this is also its weakness in that new functionality (as in a new type of query) is difficult to incorporate [58]. FANTOM [35], a tool built to annotate the mouse genome in collaborative fashion is similar to Ensembl in many respects. These approaches, while representing a specific solution to a specific problem, are not cost-effective or sustainable over the long term. In addition, an attempt was made to create a single, universal repository for all biological data called the Integrated Genome Database (IGD), which ultimately failed [27]. The main reason cited was the inflexibility and inability of the relational data model of the IGD in regards to evolving fast enough to accommodate the integrated data sources.

Some general data integration approaches have been attempted in the biomedical domain but have met with limited success [27]. This may be due in part to the fact that many users find it difficult to express their queries precisely or have the expertise to formulate them in a query language [55, 59]. Kleisli [60, 61] and TAMBIS, for example, require the user to learn CPL [62], a query language similar to SQL (but more expressive). A possible exception however is what is known as link integration. In link integration, the data integration system manages cross-references between data records which the user may follow in browsing-type fashion. Entrez [63] and SRS [64] are examples of this type of integration. The ease by which these systems are queried likely accounts for much of their success. Unfortunately, they have significant limitations in that data is compiled manually. This is time consuming and sometimes not feasible when a researcher needs information about a significant number of genes for instance. Also, researchers often want to do some automated post-processing of the compiled information, such as filtering out data which they are not confident in. This is more challenging in the link integration paradigm since the data is not integrated semantically, e.g. is not mapped onto a common model like in TAMBIS for instance, which is easier understood by computers and can facilitate easier post-processing of data [12].

The prior data integration and annotation efforts in biomedicine just discussed can be placed into three categories: 1) ad-hoc and tailored approaches, 2) general-purpose platforms, and 3) exploratory, web-based approaches. The pros and cons of each category and representative examples are summarized here:

- Ad-hoc annotation systems in biomedicine
 - Examples: Ensembl [36], FANTOM [35]
 - **Pros:** highly tailored search functionality and user interfaces for annotation.
 - **Cons:** inflexible in that new functionality and data sources are difficult to incorporate. Based on data warehouses which are difficult for users to implement, populate, and especially evolve and maintain [51].

- General data integration platforms
 - Examples: Kleisli [60], TAMBIS [61]
 - **Pros:** flexible data integration, common data model (TAMBIS) which facilitates addition of new functionality. The less rigid data model makes it easier to create a common data schema in these systems than in a data warehouse.
 - **Cons:** these systems are hard to query. They require knowledge of a complex query language and the specifics of the data schema, such as the entities, attributes, and relationships [57].

- Exploratory, web-based data integration
 - Examples: Entrez [63], SRS [64]
 - **Pros:** Easy to query in that the user simply follows web-links. They provide access to heterogeneous sources of data.
 - **Cons:** Sources of data still fixed, no common data model so compilation of data and post-processing is manual, not high-throughput. Search

space is ad-hoc and data integration is inconsistent, depending on the user [51].

Ideally, the best data integration approach for biomedicine would include ease of querying, utilization of a common data model, and flexibility in adding data sources. A federated approach with flexible schema modeling would also be preferable to alleviate hardware, data update, and data modeling concerns (Table 3.2). Given these issues, the BioMediator data integration system, a general purpose data integration system that addresses some of the weaknesses in other such systems, appears to be well suited for the biomedical domain. It is discussed in the next section.

Table 3.2: Matching needs and requirements of protein annotation and data integration technologies. These needs are best met by the BioMediator Data Integration System (section 3.3).

Annotation Requirement	Technology
Up-to-date data No local storage	Database federation
Complex data model Irregular data structure	Semi-structured data model (XML)
Imprecise queries Ease of querying	Link integration or Method-based query model

3.3 The BioMediator Data Integration System

BioMediator is a system built to address data integration needs in the biomedical domain, specifically the integration of web-based biological sequence databases. Its underlying technology addresses limitations faced by the data integration systems just discussed. Specifically, it employs a database federation with a mediated schema. This enables more flexibility in the type and amount of data sources to integrate as well in the creation of the common data model [51]. It is also much easier to query than other general-purpose data integration systems [55], an important aspect for users. The BioMiner system, which was created and described in this dissertation for the purposes of protein annotation (Chapter 4), is built upon BioMediator. This section describes the BioMediator system architecture as well as its design principles in regards to the data integration concepts previously discussed in section 3.1.

3.3.1 Overview

BioMediator is a general-purpose, federated, data integration system. It is driven by a flexible mediated-schema data modeling paradigm and offers method-based querying, which is akin to browsing. Descriptions of the various components of the system are adapted from [55].

3.3.2 Architecture

BioMediator has a modular and highly componentized architecture. It is also extremely flexible, a necessity when integrating multiple web-based biomedical data sources. The core components are the data source wrappers, the source knowledge base, the query

processor, and the user interfaces, all of which are discussed in the following sections. The BioMediator system processes queries as follows: 1) a user seeds the system with an initial query (a mediated schema entity) which is passed to the query processor, 2) The metawrappers/wrappers translate the seed query into source specific queries and pose them to the specific data sources, 3) data sources return data which is mapped onto mediated schema entities (e.g. translated into the common data model), 4) the query processor then generates events which can be used to synthesize a navigable, graph-based representation which may be repeatedly queried, expanded, and grown in the user interface. Results can also be exported for post-processing or viewing in alternative interfaces. Figure 3.1 provides a graphical view of the components of BioMediator architecture.

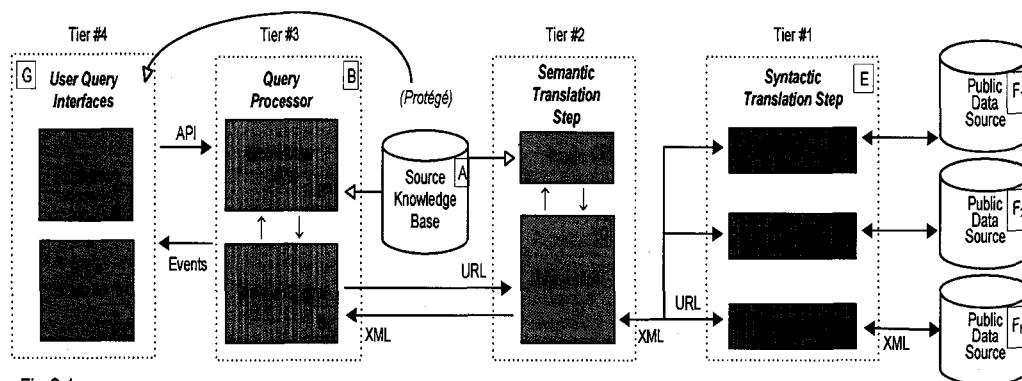


Fig C.1.

Figure 3.1: Architecture of the BioMediator system. The core components are the data interfaces or wrappers, the Source Knowledge Base which contains the mediated schema, the Query Processor, and the User Query Interfaces (courtesy of [55]).

3.3.3 Data interfaces

In the BioMediator system wrappers are implemented as HTTP servlets which accept queries in the form of URLs and handle the syntactic query translation between BioMediator and the data sources. They also accept results from the sources in their native format, such as ASCII text or HTML, and return them to the system as reformatted XML. A specialized wrapper called the metawrapper identifies data entities in the reformatted XML using information supplied in the Source Knowledge Base (section 3.3.4) and maps them onto the common data model [55].

3.3.4 Mediated schema

The BioMediator system is driven by its Source Knowledge Base (SKB), which contains the mediated schema (common data model) as well as data source catalog. The source catalog contains descriptions of the underlying data sources such as the mediated schema concepts they contain as well as interrelationships between sources. The mediated schema can be seen as a common data model which provides a unified view of the biomedical data in the source catalog. It contains hierarchical descriptions of biomedical concepts (i.e. biology) and the relationships between them [55]. A mediated schema differs from a global schema in that only desired concepts need be modeled, which offers greater flexibility in that changes to the data source schemas will not affect the mediated schema [56]. The mediated schema in BioMediator is edited and accessed using the Protégé Knowledge Base [65] and can be swapped in and out as necessary to meet the needs of particular researchers.

3.3.5 Query model

BioMediator does not employ a query language such as SQL. Instead, the approach is to use a method-based, or browsing, query model. The model can be seen as “exploratory” where users follow paths rather than state and explicit query. To query BioMediator, users initiate a seed query by specifying a desired concept type in the mediated schema as well as appropriate attribute-value constraints (e.g. Gene:symbol='HK1'). The system then queries all sources with the specified concept types and retrieves data records satisfying the constraints [39]. More data is retrieved through a process called query expansion. From the initial data retrieved via the seed query, the result set can be expanded or grown by the user, in browsing fashion, by following explicit concept relationships specified in the SKB [55].

3.3.6 Uncertainty extensions

Biomediator has recently been augmented with functionality to enable it to handle uncertainty in data. The new system, called UII for “Uncertainty in Information Integration”, is an NSF-funded project (Chapter 2, section 2.3.2) which researches formal methods for integrating uncertain information. It was formed with the explicit purpose of adding uncertainty functionality to the BioMediator system (section 3.4).

3.3.7 User interfaces

The browser-based query model in BioMediator enables results to be viewed as a graph with data concepts akin to nodes and relationships between concepts referring to edges. The graphical user-interface supplied with the core BioMediator system includes a

component which utilizes the graph-based visualization software TouchGraph [66] for viewing result sets (Figure 3.2, and Figure 3.3). In our experience, a graph-based display is very useful for demonstration purposes and validating result sets. Most casual users however have difficulty interacting with the graph-based display. Results in BioMediator however can be exported as XML which enables alternate interfaces to be implemented such as in the BioMiner system Chapter 4.

3.3.8 Data integration example

BioMediator has demonstrated that it can integrate and compile information from multiple and diverse biomedical data sources [67]. As the number of sources grows however, result sets from BioMediator can become very large. Much of this problem stems from the fact that biomedical data contains inherent uncertainties. Current data integration systems do not handle uncertainty in the data well which can lead to explosions of less relevant answers to queries (Figure 3.2, and Figure 3.3) [39]. This overwhelms human users and impacts the utility of the system. Some attempts were made to incorporate “rule-based filtering” to remove nodes in the result graph based on specified criteria [12], but it is often difficult to determine rules which model human annotation steps, such filtering by a score threshold. Rules may also be idiosyncratic to a particular lab or researcher or may only be applicable in particular situations. Still, the combination of rules plus data integration has benefits. In a study, done as part of Eithon Cadag’s Master’s Thesis [68], that I was involved in which used BioMediator with incorporated rule-based functionality, BioMediator was able to improve upon the existing annotations of 116 randomly selected proteins from GenBank 78% of the time

[12]. This study demonstrated and justified the initial rationale for using data integration to improve protein annotation. Instead of utilizing rule-based methodology, I felt that handling uncertainty in the data as well as the data integration process would provide a more robust and general approach for annotating proteins.

To best address the issue of handling uncertainty in data in a formal way, new functionality was incorporated into the BioMediator system. The BioMediator system plus uncertainty functionality is known as the UII system (for Uncertainty in Information Integration), which is described in more detail in the following sections.

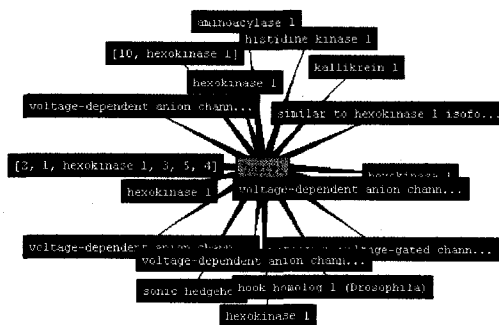


Figure 3.2: BioMediator result viewed as a graph with nodes akin to mediated schema concepts and edges referring to relationships between concepts. Results are derived from an initial seed query for the HK1 gene (Gene:symbol="HK1"), after 1 expansion.



Figure 3.3: The same result set as in Figure 3.4 after 4 expansions. The size of the result set (e.g. nodes and edges) becomes large very quickly. Users have difficulty analyzing and selecting relevant information from result sets of this size.

3.4 Uncertainty: The Uncertainty In Information Integration (UII)

Project

Data integration is insufficient in isolation to meet the needs of biomedical researchers. The problem lies in the inherent uncertainty of biomedical data and that existing data integration systems do not handle uncertainty well. These conditions have given rise to the NSF-Funded UII project (NSF Grant: NSF IIS-0513877), previously mentioned in Chapter 2, section 2.3.2. The principle investigators on the UII project are Peter Tarczy-Hornoch, MD, and Dan Suci, PhD, both of the University of Washington. Other team members include Todd Detwiler and Ron Shaker (software engineers), Wolfgang Gatterbauer, PhD, as well as myself. The overall aim of the UII project is the “design and implementation of information integration systems that handle uncertainty

in data at all levels of the integration process”. The following section, mostly adapted from [39], describes the architecture and functionality of the UII data integration system.

3.4.1 Overview of the UII Data Integration System

The UII system is built on the BioMediator data integration system but incorporates new uncertainty functionality, called “uncertainty metrics”. The uncertainty metrics represent a formal framework for representing uncertainty in data as well as in the process of integrating data. These were conceived and developed by members of the UII project (including myself) and implemented in the BioMediator data integration system by the software engineers (Todd Detwiler and Ron Shaker). Uncertainty metrics (sometimes referred to as parameters) in UII are probabilistic (0.0-1.0) values assigned to all “instantiated” concepts (entities) and relationships in the mediated schema (e.g. when a result set is created). The UII system then automatically generates a summary (or relevance) score for each entity in the result set by accounting for all these uncertainty metrics. This score essentially represents the measure of “belief” between a result set entity and the initial seed query. The benefit of the relevance score is that results can be ranked, highlighted or filtered based on the score which makes it much easier on the user to select relevant information. In this way the “data overload” problem which plagues existing data integration systems may be alleviated.

3.4.2 Related work: data integration and uncertainty

Work in probabilistic databases has been burgeoning. Mystiq [69] is a probabilistic relational database, BIOZON [70] is a graph-based data warehouse which takes into account “fuzzy searches”, and MiMI [71] is a probabilistic XML database. BIOZON in particular ranks information to present to users, very much in the spirit of the UII project. To achieve this, BIOZON uses the link structure (i.e. result graph topology) between data items to rank information. Network Reliability Theory, The uncertainty model on which UII is based, implicitly accounts for this link structure by taking into account the number of paths to nodes, but goes farther in that it represents uncertainty inherent to the individual data entities and links between them. While the approach taken by UII is more computationally expensive, results from our evaluations indicate that ranking by link structure alone is insufficient for protein annotation (see Chapter 6, section 6.5). Also, these projects are focused on creating a centralized database and not an information integration system. A centralized database with a tightly integrated schema allows for such things as powerful query capability and data cleaning, such as redundancy removal [70, 71]. Creating these centralized databases is beyond the capabilities of most small biological research labs however, who often wish to integrate a portion of up-to-date data from multiple and diverse sources of their own choosing. The data sources are, for the most part, fixed in Mystiq, MiMi, and BIOZON.

In regards to protein annotation, the biological community has created centralized web-services to help in this regard. InterPro [72] and CDD [32] are web-

based resources which integrate “pattern” databases. Pattern databases distill information in related proteins of known function to create general descriptors, such as Hidden Markov Models, which can be used to classify proteins of unknown function. There is generally a probabilistic score associated with their predictions and the interfaces to these resources often presents function predictions to users as lists ranked by score. Other than this, they do not handle uncertainty in their data in any explicit manner.

There are many of these web-services to choose from [34] and most operate independently. InterPro and CDD demonstrate the utility of integrating these various pattern databases but there are limitations in their approaches. For both resources, the data sources are fixed and they do not integrate all available sources. CDD does not actually integrate data from its underlying sources and results are returned in a haphazard way, i.e. it is sometimes difficult to select the best result. InterPro actually integrates data by hand-merging redundant descriptors in its underlying sources, which has obvious limitations. As stated previously, other than by ad-hoc ranking of prediction results by probabilistic score, none of these resources model uncertainty explicitly.

3.4.3 Uncertainty metrics

The uncertainty metrics in the UII system are a formal framework for describing the quality of data in the system in four general ways: 1) the quality of a data source, 2) the quality of the cross-references between sources, 3) the quality of a data entity, and 4) the quality of a cross-reference between data entities. The uncertainty metrics are

interpreted probabilistically (0.0-1.0 values) and are summarized in Table 3.3.

They are stored as annotations on the UII result graph (Figure 3.8).

Table 3.3: The probabilistic metrics in the UII data integration system. There are two at the “Set” or database level and two at the individual record level. They represent the uncertainty in data records and the relationship between data records as well as the uncertainty in data sources or links between data sources.

	Set Level (Database)	Record Level (Data record)
Mediated Schema Entity	$p_s \in 0,1 $	$p_r \in 0,1 $
Mediated Schema Relationship	$q_s \in 0,1 $	$q_r \in 0,1 $

The UII system calculates global relevance scores for each entity based on these local metrics (see 3.4.4). The following are expanded descriptions of the four uncertainty metrics with examples:

- 1) P_s measure: This is a quantification of a user’s prior belief in the quality of data records of a particular mediated schema entity from a particular data source (Figure 3.4). Take for example the SwissProt and TrEMBL databases. SwissProt is a carefully curated database of proteins and associated function.

TrEMBL is an analogous database but contains proteins whose functions are computationally derived. Biologists generally trust protein function assignments from SwissProt more so than from TrEMBL. In the UII system, a “Gene” entity from SwissProt would be assigned a higher Ps value than from TrEMBL.

- 2) Qs measure: This is a quantification of a user’s prior belief in the quality of a particular relationship, as defined in the mediated schema, between two sources (Figure 3.5). For example, data records in one source may cross-reference records in another source by globally unique identifiers. Some sources however may reference records from other sources by inexact text-string similarities. The Qs value for the relationship between two sources which identify records between then using unique identifiers should be higher than if inexact text-string matching is employed.
- 3) Pr measure: This measure is a quantification of a user’s belief in a particular data record of the same mediated schema type and data source (Figure 3.6). Unlike the Ps measure which is static and determined a-priori, the Pr measure is dynamic and calculated at runtime. For example, Gene records in Entrez contain a “status code” attribute which indicates up to seven levels of curation for a particular record. Gene records from Entrez could be assigned Pr values according to their level of curation.
- 4) Qr measure: This measure is a quantification of a user’s belief in a particular cross-reference (e.g. link) between two data records (Figure 3.7). It is calculated

dynamically, much like the Pr measure. An example of this could be protein records from different sources which cross-reference each other by a probabilistic similarity algorithm. The Qr measure could be calculated dynamically given the score(s) from the similarity algorithm.

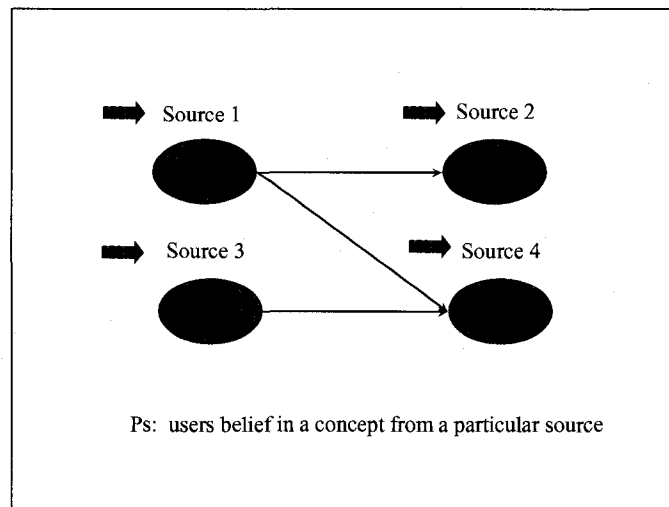


Figure 3.4: Illustration of the Ps metric. The Ps metric is a user's belief in the quality of a particular mediated schema entity from a particular source, which is interpreted probabilistically (0.0-1.0) value.

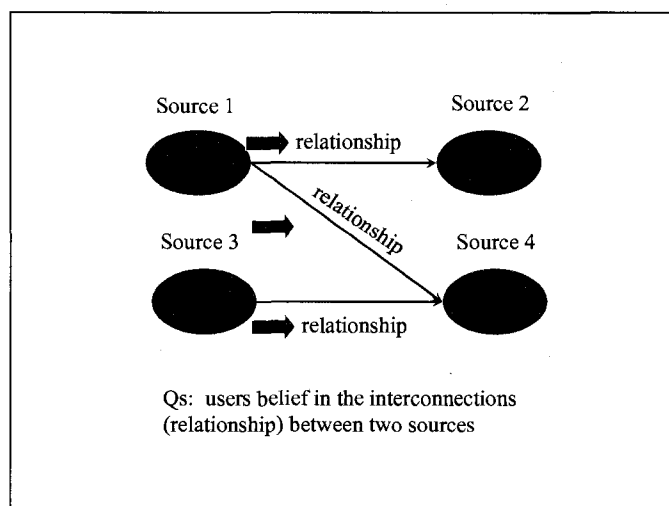


Figure 3.5: Illustration of the Qs metric. The Qs metric is a user's belief in the quality of a particular relationship, as defined in the mediated schema, between two sources, which is interpreted probabilistically (0.0-1.0) value).

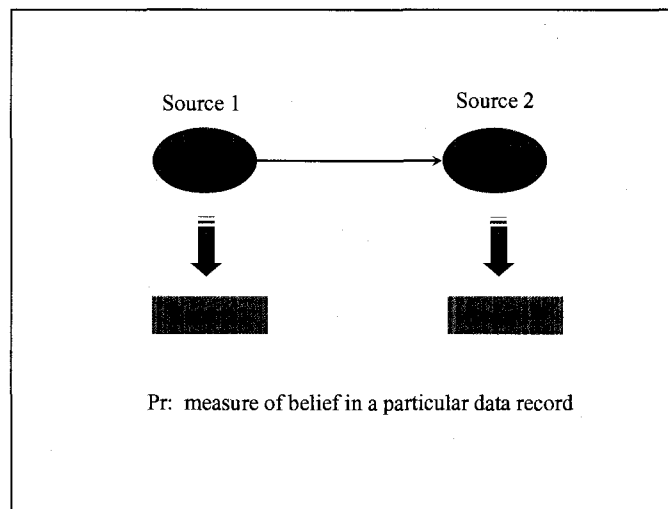


Figure 3.6: Illustration of the Pr metric. This metric is a user's belief in a particular data record of the same mediated schema type and data source. It is interpreted probabilistically as a 0.0-1.0 value, which is dynamically determined when a record is retrieved.

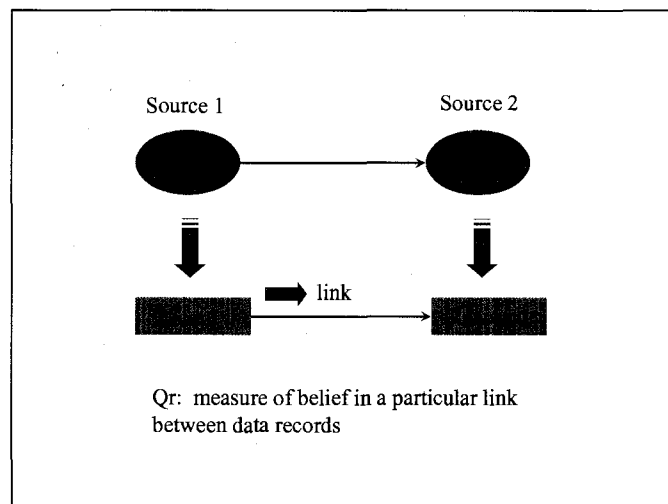


Figure 3.7: Illustration of the Qr metric. This metric is a user's belief in a particular cross-reference, or link, between two data records. It is interpreted probabilistically as a 0.0-1.0 value, and is dynamically determined when records are retrieved, much like the Pr metric.

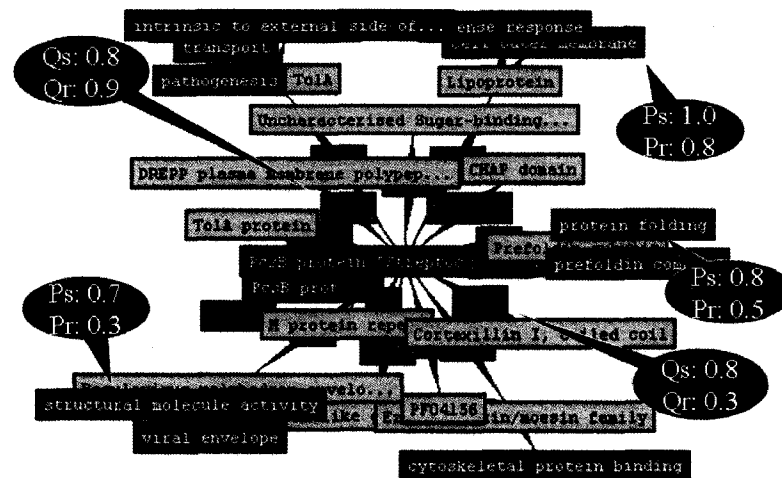


Figure 3.8: An illustration of a result graph from the UII system annotated with uncertainty metrics. Ps and Pr metrics are assigned to the nodes, which correspond to mediated schema entities. Qs and Qr metrics are assigned to the edges, which correspond to relationships in the mediated schema.

3.4.4 Relevance scoring algorithm

The uncertainty metrics in section 3.4.3 are all local measures for a particular entity or relationship, e.g. they are not necessarily comparable between entities for instance.

What we want is a global measure of the relevance of each entity (or node in the result graph), which is based on the uncertainty metrics. To address this problem, we recast it in terms of a network reliability problem [73]. The restated problem is thus: for each node n_1 in the result graph (or network), $Ps_{n_1} * Pr_{n_1}$ is the probability that the node is present in the network. Likewise, $Qs_{e_1} * Qr_{e_1}$ is the probability that a given network link e_1 is available. The relevance of a node is then calculated as the probability that the node is reachable from the initial seed node. This probability is influenced by the quality and quantity of paths from the seed node. The network reliability calculation is intractable for exact probabilities but an efficient method which uses simulation for

approximating the probabilities does exist [74]. For our purposes we simulate, in a single pass, N trials (path traversals) where nodes and edges are included in the traversal with associated probabilities. This is done by storing a randomized N -bit trial vector associated with each node and edge where each bit is a binary value denoting success or failure of a particular trial (based on their uncertainty metrics). In a depth-first search of the graph (beginning from the seed node), we populate a success vector for each node which indicates for each trial whether or not that node is reachable by some path. For each node a count (k) is computed which is the number of times a node could be reached, via some path, over the total number of trials (Figure 3.9). The final score, or relevance, for a node is then estimated using the quantity k/N , where k is the number of set bits in a node's success vector. The choice of N influences the error in the probability estimation, the larger the N the smaller the error. In addition, for any N , the greater the relevance score the better the approximation will be. This means that the algorithm should generally rank the most relevant answers correctly whereas the least relevant results may be slightly misordered. This final relevance score can be interpreted as the probability that any particular node is reachable from the seed node (Figure 3.10). They are useful in that result sets can be sorted according to it and results presented to users as ranked lists.

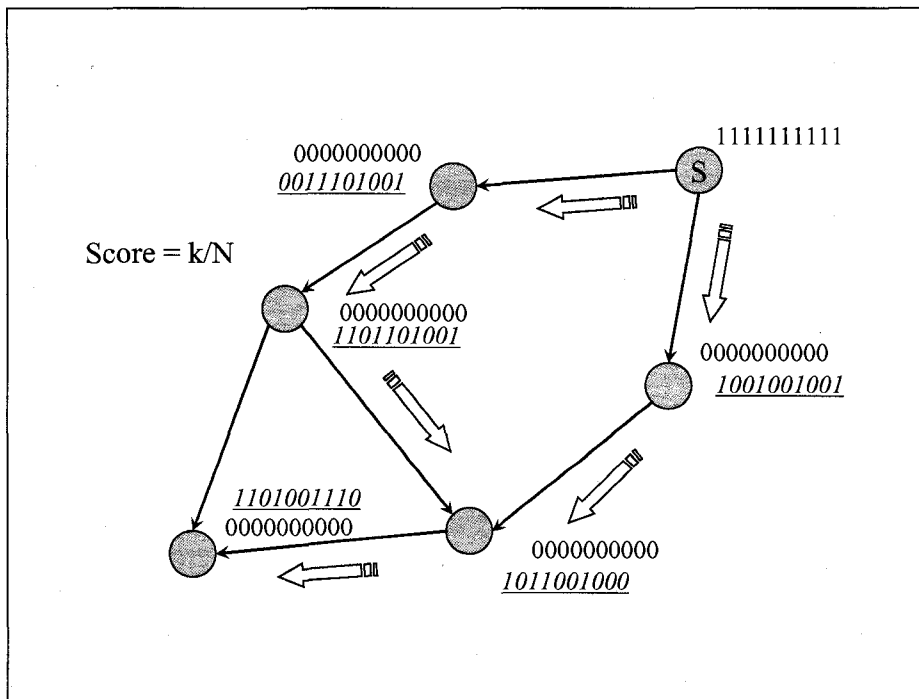


Figure 3.9: An illustrated example of the scoring algorithm in the UII system. A hypothetical result graph is shown here with the seed node marked “S”. Each node has an associated trial vector (italics) of length N which the number of trials (initially 1,000 in first implementation of UII). Each bit in the trial vector is set according to a “coin-flip” which is based on the uncertainty metric values of the node (e.g. $P_{S_{n1}} * Pr_{n1}$). For the seed node, all bits are set to 1, since that is guaranteed to be in the graph. Also associated with each node is a success vector of length N . The links (edges) between nodes also contain success and trial vectors, but they are not shown in this figure. The success vectors are set via a depth-first search (DFS) of the result graph, which is directed. Bits are set in success vectors based on the following operations: 1) for each edge, a new vector is formed by the logical AND of the head node success vector, the edge trial vector, and the tail node trial vector, 2) The bits of the success vector of the tail node are set based on a logical OR operation between the vector generated in the previous step and the current success vector of the tail node., and 3) if the previous OR operation results in the setting of any new bits, the DFS continues on this path. This final step ensures that multiple paths to a node are accounted for, such as the node where the arrows converge in the above figure. The final relevance score for each node is calculated as k/N , which is the number set bits in a nodes trial vector divided by the number of trials.

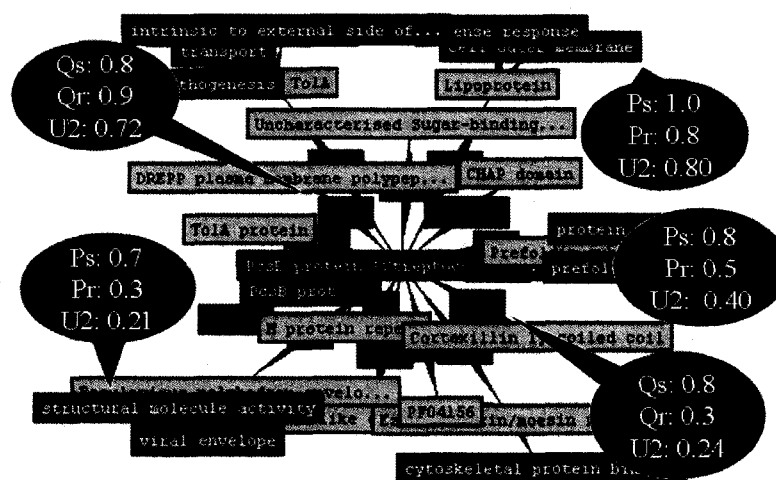


Figure 3.10: An illustration of a result graph from the UII system annotated with uncertainty metrics as well as global relevance scores. The relevance score can be utilized by an appropriate interface to provide ranked result sets to users, facilitating easier inspection of result sets.

3.4.5 The relationship between UII and this dissertation

The uncertainty metrics and relevance scoring algorithm were developed as theoretical work in data integration by members the UII project. The uncertainty functionality was incorporated into the BioMediator data integration system by Todd Detwiler and Ron Shaker, software engineers working on both the BioMediator and UII project. The UII system itself is a remarkable achievement in that it represents a general-purpose data integration system which handles uncertainty in a formal an explicit manner. The purpose of this dissertation is to implement a working instance of the UII system and evaluate it in a real-world application. The mediated schema for annotation proteins, the data source wrappers, the user interface, the values of the uncertainty metrics, the proof-of-concept were all necessary for me to add to the UII system to create a working instance of the UII system. The creation of the working system (BioMiner), and the

performance evaluation, were my efforts to leverage the general-purpose technology in UII system to solve a real-world problem, and are the primary focus of my work in this dissertation (Chapters 4, 5 and 6).

3.5 Discussion

The UII system is a general-purpose data integration system. The challenge is to apply it in a particular domain such as biomedical research, and protein annotation in particular. This is the focus of this dissertation. The UII system is one of the major components of the BioMiner system (Chapter 4), which is evaluated as a tool for protein annotation in Chapter 6. The major design paradigms of the UII system, federated data, flexible, mediated-schema based data modeling, and uncertainty handling align well with the needs of protein annotation. However there is an open question, which is faced by all probabilistic systems, and that is: where do the probabilities come from? We do address this issue specifically for protein annotation in Chapter 4 which talks about the initial probabilistic values in the BioMiner system, Chapter 6 which discusses optimization of the probabilistic values, and in Chapter 5, which evaluates the choice of probabilities in BioMiner in a general and methodological analysis, although it remains an open question.

Chapter 4: THE BIOMINER SYSTEM

4.1 Design of the BioMiner System for Protein Annotation

The BioMiner system is built for a specific purpose in the domain of molecular biology, which is to determine the function of hypothetical proteins (otherwise known as annotation). The rationale of BioMiner is that it is beneficial to utilize function predictions from many different source for the explicit purpose of improve protein annotation accuracy. This rationale is shared by Lee et al in a recent review of computational annotation in Nature Reviews [75]. However, in order to fully utilize different annotation data sources, it is imperative that uncertainty in the data be modeled. This allows for “reduction” of the integrated data sets, e.g. by ranking or highlighting the best function predictions. The development of an uncertainty model for annotation is a novel contribution of this dissertation. This uncertainty model is implemented in the BioMiner system, the details of which are the focus of this chapter. Additionally, the refinement, optimization, and evaluation of BioMiner for annotation are the focus of this dissertation.

BioMiner has two major components which are a data-integration engine, and a user interface. The data integration component of BioMiner is an implementation of UII - a general-purpose, federated, and mediated-schema driven data integration system described in Chapter 3. The user interface in BioMiner is also based on general-

purpose software, the Generic-Genome-Browser [76], which enables it to display results sets in a biologically-relevant manner.

4.1.1 Building on the BioMediator and UII Systems

The choice of a federated data integration engine which explicitly handles uncertainty is based on rationale discussed in Chapter 3. A summary of the main reasons are: 1) molecular biology databases are huge and federated data integration systems overcome the need for a large, centralized data repository, 2) data models in molecular biology can be extremely complex but flexible mediated-schema driven data integration systems can alleviate this problem by allowing the user to model only the relevant data, and 3) integrated data sets can be large and difficult for users to evaluate, so the ability to rank results based on a global relevance measure is important. The user interface for the BioMiner system is the Generic-Genome-Browser (GGB) [76]. It was chosen in part for its ability to display sequence-related information, such as the location of functional domains on a protein sequence for example. More importantly however is the GGBs ability to rank, highlight, or filter data based on a particular score – global relevance scores produced by the UII system in our case. The combination of the data integration engine which produces relevance scores as well as a user interface which can take advantage of the relevance scores makes BioMiner a complete tool for determining the function of hypothetical proteins.

4.1.2 Choice of data sources in BioMiner

The primary data sources incorporated in the BioMiner system federation share a common heritage in that they predict the function of an unknown protein by comparing its amino acid sequence to a database of proteins of “known” function. Various data sources utilize different type of comparison (or “search”) algorithms, such as BLAST [30] or Hidden Markov Models [20], as well as heterogeneous protein databases. The SuperFamily resource utilizes only those proteins with known three-dimensional structure for instance [77]. There exist dozens of data sources of this type [34]. The initial set incorporated into the BioMiner system were selected in consultation with collaborating biologists, Dr. Mark Minie, PhD, and Dr. Eugene Kolker, PhD. They are generally understood to be the most commonly utilized as well as provide the best results. After a pilot evaluation of the system the federation was expanded to include two more HMM data sources (Chapter 6), which were meant to increase the accuracy of the system for annotating hypothetical proteins. Dr. Eugene Kolker was involved in evaluating the initial results from BioMiner in terms of their accuracy for annotating proteins in *Shewanella oneidensis*, a bacterium whose proteins Dr. Kolker has much experience with. Section 4.3.1 contains the list of data sources queried by the BioMiner federated data integration system.

We only chose to incorporate data sources which predict function based on direct amino acid sequence comparison in this version of BioMiner (Table 4.1). Sequence-based analysis is the most common method for predicting function. Other data sources which predict function based on other methods such as protein interaction [23] or

protein interaction [23] or genome localization [78] were not considered in this study, but may be incorporated in future versions of BioMiner.

4.1.3 The Common data model (Mediated Schema) in BioMiner

In general, a mediated schema can be seen as a graph (or network) based representation of a domain where nodes represent entities and edges represent relationships between entities [56] (see Chapter 3, section 3.1.3). In the BioMiner system there are six major entities of interest: Protein, Domain, Family, Gene, Evidence, and Function. To clarify, “Domain” or “Family” entities generally describe functional aspects of proteins or groups of related proteins. “Evidence” describes the type of supporting biological evidence for a particular function, a biochemical assay or computational prediction for instance. A “Function” entity represents the description of a function according to the Gene Ontology [42]. “Domain” or “Family” entities also carry function descriptions but these are generally uncontrolled text. The major relationships in the BioMiner mediated schema are between two Protein entities or Protein and Domains/Family entities. These are meant to represent the relationship between an unknown protein and other proteins, domains, or families of known function. The unknown protein represents the starting “query” in BioMiner and can be related to multiple other proteins, domains, or families (Figure 4.1). The relationships linking the entities are illustrated in Figure 4.2 and Figure 4.3 and we list them here:

- **ClassifiedAs.** This relationship links Domain, Family, and Gene entities to a controlled terminology of gene function (the Gene Ontology).

- **ConservedDomainDatabaseHitRefersToProtein.** This is a similarity relationship between a protein and a function domain, such as in the CDD database.
- **ConservedDomainDatabaseRefersToDomain.** This links a sequence similarity entity to a Domain entity, also as in the CDD database.
- **Gene2Protein.** This relationship links Gene and Protein entities, it is bidirectional (i.e. a Protein can also refer to a Gene).
- **ProteinDatabaseHitRefersToProtein.** This is a similarity relationship between two proteins, as in the BLAST database for example.
- **SequenceSimilarityRefersToProtein.** This is a similarity relationship between a database of Hidden Markov Models and a protein, such as in the Pfam database.
- **SequenceSimilarityRefersToProteinFamily.** This links a sequence similarity entity to a Family entity, again as in the Pfam database.

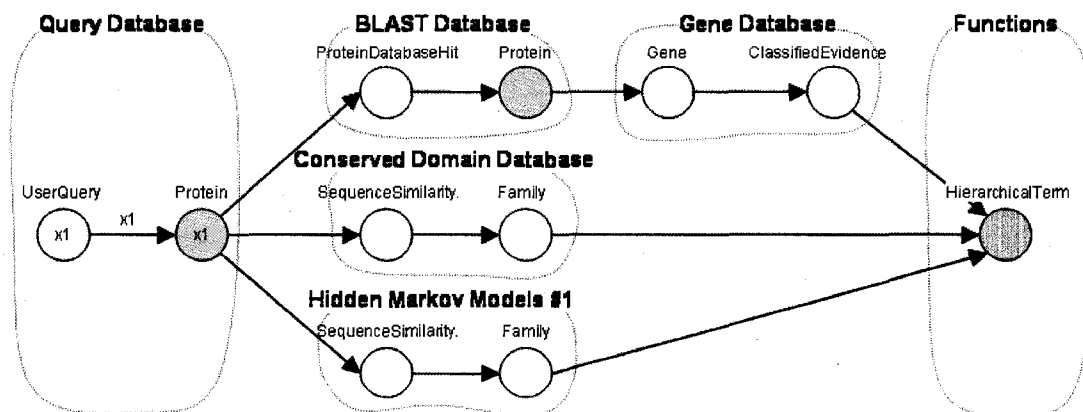


Figure 4.1: A conceptual diagram illustrating the major entities and relationships of the mediated schema in the BioMiner system. A protein of unknown function represents the user or “seed” query. BioMiner then finds other proteins, domains, or families which are related to the query protein (by utilizing the search algorithm in each particular data source). These other entities may provide a free-text

description of a biological function or may reference a Gene Ontology term. Some sources, such as the “Gene Database” here, provide descriptions of supporting biological evidence for a particular function as well. (diagram courtesy of Wolfgang Gatterbauer (modified), derived from schema in **Figure 4.2**.)

4.1.4 Uncertainty in BioMiner

Molecular biological data is inherently uncertain. For example, an algorithm to predict whether or not an unknown protein is a member of a previously characterized protein family may indicate that a protein could potentially belong to several families. A biologist inspecting the results of the algorithm may conclude that the protein actually belongs to a single or subset of the predicted families or none at all. The decision may be made on the probabilistic score of each prediction, among other things. Since the BioMiner system integrates information from multiple “uncertain” data sources, it can return lots of information for a single query protein, on the order of hundreds to thousands of nodes. Without some way to highlight or rank the best or most relevant information, results sets would be very difficult if not impossible for a human user to make sense of. BioMiner is able to rank information by leveraging functionality in its data integration component (the UII system). It performs this by accounting for all “independent” sources of uncertainty for each entity in its result set and then creating a global relevance score for each. These independent sources are the uncertainty metrics for each entity and relationship in the result set (Chapter 3, section 3.4.3). This global score can then be leveraged by an appropriate user interface to achieve functionality such as ranking, highlighting, or filtering of data. The UII system is a general-purpose framework and data-integration system which explicitly handles uncertainty. The

BioMiner system is an implementation of the UII system and a proof-of-concept that the uncertainty model in UII can capture the types of uncertainty necessary in protein annotation and that it is, in fact, useful.

With the help of collaborating biologists, Dr. Mark Minie PhD, and Dr. Eugene Kolker PhD, we were able to characterize the independent sources of uncertainty in BioMiner. The uncertainty can be classified into two broad categories which fit into the general classification of uncertainty in data from Chapter 3, section 3.4.3. The categories of uncertainty in BioMiner are : 1) confidence or trust in a particular entity, and 2) confidence or trust in the relationship between two particular entities. The uncertainty represented in BioMiner maps well to the general uncertainty model in UII Chapter 3, section 3.4. Confidence in a particular entity is best illustrated by the “ClassificationEvidence” entity (see Figure 4.1). Evidence for a biological function generally comes from direct experimental evidence or “electronic” sources. Direct experimental evidence is usually associated with a publication, which biologists have much higher confidence in versus electronic evidence which is usually a computational prediction. This concept maps well to the Ps uncertainty metric in UII (Chapter 3, section 3.4.3).

Confidence in the relationship between two entities is best illustrated by describing the relationship between two proteins. Two proteins can share the same biological function if their amino acid sequences are very similar. The degree that two proteins are similar can be measured by a probabilistic score, such as from the BLAST algorithm or Hidden Markov Models. A single protein of unknown function may be

similar to many characterized proteins (with many different functions). Usually, the function from the most similar characterized protein is selected as the function of the unknown protein, if it is above some generally accepted threshold. Conceptually, the particular similarity between two proteins maps well to the Q_r uncertainty metric in UII (Chapter 3, section 3.4.3). In addition, a biologist may prefer the BLAST algorithm over Hidden Markov Models in general. This is a data-source level concept which maps well to the Q_s uncertainty metric in UII (Chapter 3, section 3.4.3). By taking these sorts of uncertainty into account, BioMiner is able to rank information according to biologically-relevant definitions of confidence or trust. For example, if the ranking takes the form of a sorted list, then functions with associated with proteins (or domains and families) most similar to the query protein with the best types of evidence should be near the top of the list – and could more easily be inspected by a expert biologist.

4.2 Related Work

Probabilistic algorithms such as Network Reliability Theory [79] or related approaches [80, 81] have been previously attempted in the biological domain for inferring functional knowledge about proteins. These approaches differ in that they are creating a static and specific model for predicting gene function and are concerned with training the model. These approaches however differ from ours in that we are utilizing a data integration approach with uncertainty semantics. The advantage of our approach is that, since it integrates data dynamically, it automatically accounts for new information regarding gene function. In light of new information, the other systems must be re-trained.

4.3 Implementation of the BioMiner System

The BioMiner system is built with general-purpose components. Much work was still required however in order to create a system designed for the specific purpose of annotating hypothetical proteins. Wrappers for each data source as well as the mediated schema take a fair degree of time and expertise to create. Uncertainty metrics also add an additional layer of complexity. This section describes the implementation of the major components of the BioMiner system. Much of this content is adapted and modified from [39].

4.3.1 Data interfaces

Wrappers were written to meet necessary wrapper requirements of the UII (BioMediator) system [55]. Table 4.1 summarizes all data sources in BioMiner, a description of their contents, and the rationale for their incorporation into the federation.

The set of data sources in BioMiner is unique, e.g. neither InterPro or BioZon contain the annotation sources integrated in BioMiner. For instance, InterPro does not contain the CDD database and BioZon only contains InterPro. Note that CDD contains multiple data sources, a very important one being PRK [82]. We demonstrate the importance of this database for annotation in Chapter 6, section 6.5.

Table 4.1: Data sources in the BioMiner federation, a description of their contents, and their rationale for incorporation.

Data Source	Description	Rationale
Gennav	Gene Ontology Database	Controlled terminology of function description
BLAST (NCBI)	Protein Database	Protein Function Prediction
CDD	Conserved Functional Domain Database	Protein Function Prediction
Entrez (Gene)	Gene Database	Function Descriptions
Entrez (Protein)	Protein Database	Function Descriptions
PDB	3-D Structures	Function Descriptions
Pfam	Protein Family Database	Protein Function Prediction
PIRSF	Protein Family Database	Protein Function Prediction
PSI-BLAST (UniProt)	Protein Database	Protein Function Prediction
SuperFamily	3-D Structures	Protein Function Prediction
TIGRFAM	Protein Family Database	Protein Function Prediction

Entrez Protein and Entrez Gene [63] were the primary set of data source wrappers needed to retrieve information the amino acid sequence of a seed query protein (of unknown function). They also provided chromosomal location, information which was necessary for display in the Generic-Genome-Browser, the user interface in BioMiner (see 4.3.4). The secondary set of data sources take the protein sequence as input to their various search algorithms and return similar proteins, or domain and family predictions. This set included UniProt [19] and PDB [83], searched by BLAST and PSI-BLAST. BLAST and PSI-BLAST are the basic algorithms used to predict protein function. The Conserved Domain Database (CDD) [32] along with two of its major components

COG, and PRK [82]. These databases describe “domains” or common subsequences in proteins which may carry out important functions in many organisms. Two databases of Hidden Markov Models (HMM), Pfam [20], and SuperFamily [77], were also included. HMM databases are supposed to be more “sensitive” than BLAST or PSI-BLAST in finding very distant family relationships. To standardize gene function nomenclature the Gennav database [84] which provides an interface to the Gene Ontology, was also included. After a pilot evaluation of the system (Chapter 6), the federation was expanded PIRSFScan [85, 86], and TIGRFAM [21], which are also HMM data sources. PIRSFScan concentrates on creating very accurate protein family descriptions and predictions. TIGRFAM tends to focus on prokaryotic proteins.

4.3.2 Common data model (Mediated schema) and mappings

The flow of information in the BioMiner system is from a seed query protein of unknown function to Domains, Families, or other Proteins of known function. If the data source describes functions using GO terms, then GO references are followed to the Gene Ontology database. Unfortunately, not all data sources utilize GO terms so references to the Gene Ontology database were not possible in all cases. A diagram of the mediated schema in BioMiner and Source Knowledge Base (or SKB) can be seen in Figure 4.2 and Figure 4.3. The entities and relationship in each data source can be found in Table 4.2.

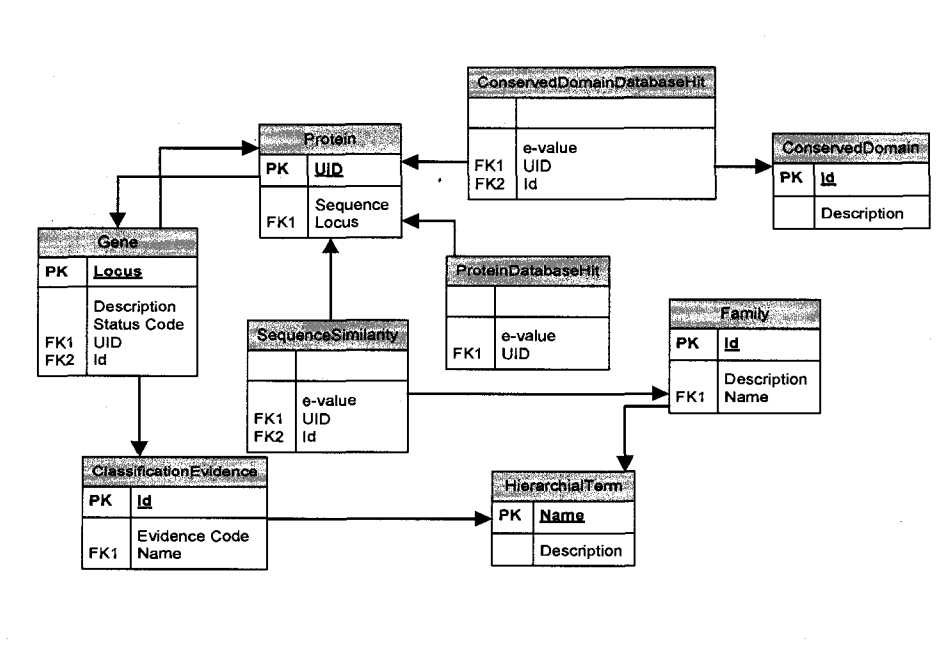


Figure 4.2: The mediated schema in BioMiner. The primary entities such as Protein, Family, and HierarchicalTerm can be seen here along with their relationships. SequenceSimilarity, ProteinDatabaseHit, and ConservedDomainDatabaseHit are reified relationships in the BioMiner schema which represent the quality of the similarity relationship between a query protein and another protein, protein family, or conserved domain.

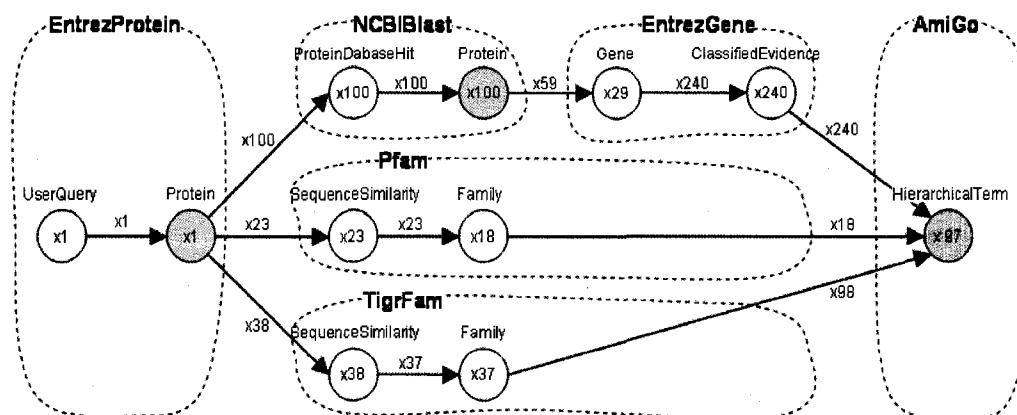


Figure 4.3: A portion of the Source Knowledge Base (SKB) in the BioMiner system. The SKB is the mediated schema with data sources and entities and relationship within and between the sources. Included are possible numbers of instantiated entities from each source. Some protein function data sources were omitted but their entities and relationships would be identical to Pfam and TigrFam in this diagram. Relationships between sources represent direct references between, such as GO term identifiers in Pfam referencing GO terms in Gennav (AmiGO), or are the result of search algorithms such as when NCBI Blast is searched with the query protein via BLAST. With the exception of the "UserQuery to

Protein” link, relationships can be one-to-many. (This is an instantiation of a portion of the mediated schema in BioMiner (**Figure 4.2**). The diagram is courtesy of Wolfgang Gatterbauer (modified)).

Table 4.2: Data sources in BioMiner with their entities and relationships. Relationships are between databases and only show the references which point “outward”, i.e. refer to other databases.

Data Source	Entities	Database Relationships
Gennav	HierarchicalTerm	(none)
BLAST (NCBI)	Protein, ProteinDatabaseHit	Entrez (Protein), Entrez (Gene) PDB
CDD	ConservedDomain, Ortholog, ConservedDomainDatabaseHit	Entrez (Protein)
Entrez (Gene)	Gene, ClassificationEvidence	Gennav, Entrez (Protein)
Entrez (Protein)	Protein	Entrez (Gene)
PDB	Structure	Entrez (Protein)
Pfam	Family, SequenceSimilarity	Gennav, Entrez (Protein)
PIRSF	Family, SequenceSimilarity	Gennav, Entrez (Protein)
PSI-BLAST (UniProt)	Protein, ProteinDatabaseHit	UniProt
SuperFamily	Family, SequenceSimilarity	Entrez (Protein)
TIGRFAM	Family, SequenceSimilarity	Gennav, Entrez (Protein)

4.3.3 Uncertainty metrics

Uncertainty metrics for each entity and relationship in the mediated schema are assigned probabilistic values from 0.0-1.0 and are based on attributes of the data entity or relationship, per the definitions described in Chapter 3, section 3.4.3. For example, “Genes” from EntrezGene are assigned “Status Codes” which indicate the degree to which a gene record has been curated by an expert biologist (more curation is generally

better). It was often difficult to determine a consistent method to assign uncertainty metrics to particular entities from particular sources. Most uncertainty metrics in the BioMiner system ended up being 1.0 (considered the “default”). For instance, all data sources in the BioMiner federation were considered to be of equivalent quality, thus each received a “Ps” value of 1.0. Table 4.3 provides definitions of the most important uncertainty metrics in BioMiner. Much of this section is adapted from [39].

Table 4.3: The most important uncertainty metrics in the BioMiner system. Uncertainty metrics for “Gene” entities are calculated based on their Status Code values from Entrez Gene. Uncertainty metrics for “Evidence” entities are based values specified by Gene Ontology evidence codes, which indicate the supporting evidence for a particular function. Protein->Domain, Protein->Protein, and Protein->Family are relationships which are derived via search algorithms which are associated with a score (evaluate, expect), which is then converted to a probabilistic value via the function shown. The “Metric” column refers to UII uncertainty metrics defined in Chapter 3, section 3.4.3.

Entity or Relationship	Attribute	Metric	Calculations
Gene	Status Code	Pr	Reviewed (1.0) Validated (0.8) Predicted (0.4) Model (0.3) Inferred (0.2)
Evidence	Evidence Code	Pr	IDA (1.0), TAS (1.0), IGI (0.9), IMP (0.9), IPI (0.9), IEP (0.7), ISS (0.7), RCA (0.7), IC (0.6), NAS (0.5), IEA (0.3), ND (0.2), NR (0.2)
Protein->Domain Protein->Protein Protein->Family	evaluate Expect	Qr	$abs\left(\frac{\log_{10}(evaluate)}{300}\right)$

4.3.4 User interface

There are two ways to view results in the BioMiner system. The first is the standard “table” view which is supplied with the basic UII system (Figure 4.4). The second is an implementation of the GGB (Figure 4.5). Both interfaces enable ranking, and highlighting of results based on UII score. Only the GGB enables filtering, such as on a score threshold for instance. The advantage of the GGB is that it displays location information, which may be important from a biological perspective (Chapter 2). The GGB also allows more flexibility in how to display results by enabling the user to display different types of entities in the same “track”. The difference in top-ranking results between Figure 4.4 and Figure 4.5 is due to the GGB displaying both “ConservedDomain” and “Family” entities in the same track, unlike in the table view where only “Family” entities are shown.

Name	Type	Database	UII
cysteine desulfurases, SufS subfamily	Family	TIGRFAM	7.362986394...
Aminotransferase class V	Family	Pfam	6.731520889...
lyxurentase	Family	TIGRFAM	5.862937548...
cysteine desulfurase family protein	Family	TIGRFAM	4.342916702...
cysteine desulfurase family protein	Family	TIGRFAM	3.474335362...
L-aminooxyphosphonate aminotransferase	Family	TIGRFAM	1.737166661...
protein of unknown function (SUF1556)	Family	Pfam	6.514375054...
Aminotransferase class I and II	Family	Pfam	0.0
Cys/Met metabolism PLP-dependent enzyme	Family	Pfam	0.0

Figure 4.4: Table view of ranked results from BioMiner system. These results indicate possible functions for the hypothetical protein SO4413. In this case, the top ranking result is “cysteine desulfurases, SufS subfamily” from the TIGRFAM database, but only Family entities are shown in this view. Results for the same hypothetical protein are also shown in the Generic-Genome-Browser (GGB) in Figure 4.5, but the top-ranking result is different.

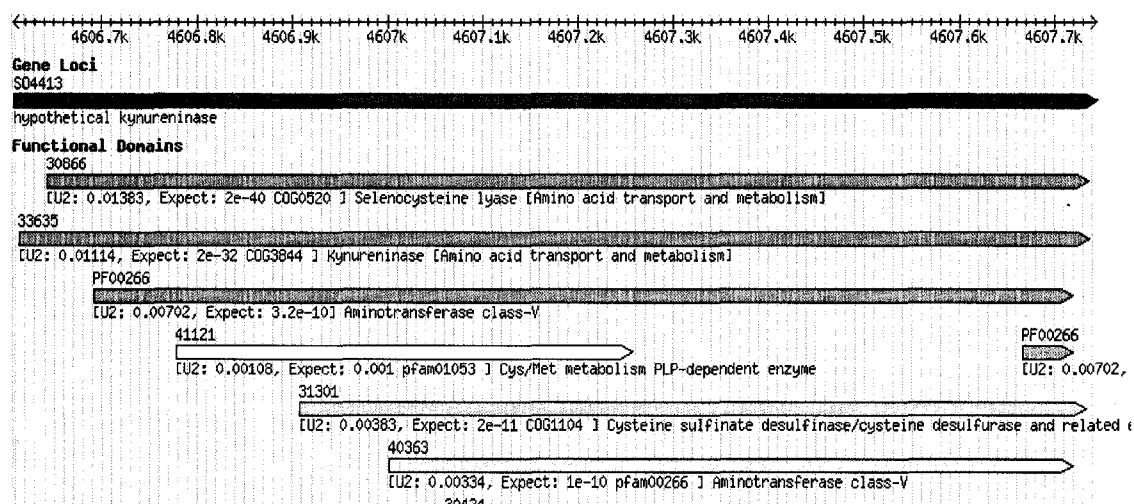


Figure 4.5: Results from BioMiner as viewed in the GGB. The top hit is “Selenocysteine Lyase” from the COG database. The GGB allows for display of different mediated schema entities (such as ConservedDomains and Families) in the same track, which is called “Functional Domains” in this case. Grouping different entities in the same track allows for easier inspection by relevance score. So, for the hypothetical protein SO4413, the highest ranking result is “Selenocysteine Lyase” from the COG database, and not “Aminotransferase” from the Pfam database as a user might conclude from Figure 4.4.

A possible disadvantage of the GGB as it is used in BioMiner is that it is decoupled from the data integration component, e.g. the data model in the GGB is completely separate. Results must be exported from BioMiner and reformatted for the GGB before they can be viewed. This was performed using a Python script which parsed BioMiner output files and transformed them to correspond to the data model in the GGB. While this adds to the number of steps needed to view results, it has an important benefit in that it greatly speeds up the response time from the user perspective. The data integrating step, which involves querying data sources across a network, can be very slow - on the order of tens of minutes. When these results are reformatted and imported into the GGB, the GGB acts as a “cache” where results appear almost instantaneously, greatly reducing burden on the user. Regular updates of

the data in the GGB can be easily performed behind the scenes using BioMiner to retrieve the data and alleviate the potential problem of data becoming stale.

4.4 Evolution of the BioMiner system

The BioMiner system is highly experimental, therefore many aspects of the initial system such as the wrappers, interface, and uncertainty metrics were only prototypes. Once a working system was in place, subsequent evaluations enabled us to refine the system for its intended purpose. The following section describes the modifications to BioMiner after each evaluation. Descriptions of the evaluation studies and their results are in Chapter 6.

4.4.1 BioMiner 1.0

Version 1.0 was the first prototype of BioMiner. The set of data sources did not include TIGRFAM and PIRSFScan. Initial values for the uncertainty metric parameters had just been determined. The user interface was the standard table view (Figure 4.4) as the GGB was not incorporated at this time. Version 1.0 was used in the first “proof of concept evaluation”, described in Chapter 6, section 6.3.

4.4.2 BioMiner 1.1

The key differences between Version 1.0 and Version 1.1 were an optimization of a parameter in the relevance calculation algorithms and the implementation of a new interface to view results. Version 1.1 of BioMiner was used for a pilot protein annotation study to get an indication of its utility in the domain for which it was intended. The proof of concept evaluation indicated that the precision of the

approximation algorithm used to calculate relevance scores needed to be greater. The “trial” parameter in the algorithm was thus set at 10,000 - a ten-fold increase. Results needed to be shown in a biological context for domain experts to inspect so the GGB was incorporated into the system. Functionality regarding ranking and highlighting of result data was also implemented in the GGB. For a description of the pilot protein annotation study and results see Chapter 6, section 6.4.

4.4.3 BioMiner 1.2

The key differences between Version 1.1 and Version 1.2 of BioMiner were the addition of two more HMM data sources and the optimization of its uncertainty metrics. Version 1.2 of BioMiner was used for a hypothetical protein annotation study, which was an evaluation of the system in a real-world use case. The GGB performed well in the previous pilot annotation study and no major changes were implemented there. Results from the pilot protein annotation study indicated that two more data sources should be incorporated into the system to improve its ability to annotate proteins: TIGRFAM and PIRSFScan. A Sensitivity Analysis had also been undertaken and uncertainty metrics in BioMiner were optimized given results from this study (Chapter 5) as well as an annotation “training” set (Chapter 6). For a description of the hypothetical protein annotation study and results, see Chapter 6, section 6.5.

4.5 Discussion

The significant and novel contribution of this chapter is the development and implementation of an uncertainty model for protein annotation (BioMiner), which

builds on the UII/Biomediator system. Building BioMiner and evolving it was necessary to address the overall research question of this dissertation (Chapter 1, section 1.2). The uncertainty model in BioMiner includes the unique set of annotation data sources, the common data model for integrating the sources, and the parameter values for the uncertainty model. Moreover, we validate the uncertainty model in Chapter 5 and demonstrate that the uncertainty model is robust as to choice of probabilistic parameters. Additionally we demonstrate the advantages of the model over existing approaches for annotating proteins in Chapter 6. This is performed in a real-world application scenario with great importance to biologists.

The uncertainty model in BioMiner does add a significant layer of complexity during implementation however. There are a lot of metrics (parameters) to populate and some can be difficult to determine. Data sources may not store necessary data or provide sufficient documentation. Additionally, there is also an open question regarding the choice of values for uncertainty metrics, e.g. some argue that they should to be determined using machine learning techniques, although we partially address this issue in our Sensitivity Analysis study in Chapter 5.

A compelling upgrade to the user interface in BioMiner would be functionality to allow manual tuning of the uncertainty metrics by the user. Users could then express their own preferences or input probabilistic values learned from proprietary data. Of course, one should consider the degree to which uncertainty metrics are exposed given the potential effect it could have on results produced by the system. For instance, only the source-level metrics could be exposed while the data-level metrics remain hidden.

This exploration into the manual tuning of uncertainty metrics by users remains an interesting avenue of future work regarding the BioMiner system.

Chapter 5: THE ROBUSTNESS OF BIOMINER

5.1 The stability of results from the BioMiner system

Pilot studies have indicated that BioMiner with its default parameter values is able to provide plausible rankings of its result sets, at least on a limited basis [39]. However, a common question we encounter is how we determined our parameters in the first place. The concern expressed is that wrong or imprecise parameter values can lead to improper ranking of results by our system in more general cases. To be sure, determining precise parameters for BioMiner is extremely difficult as it generally involves learning them from data or intensive publication searches. This issue is not unique to us. It is also an issue in the medical domain when studying Bayesian belief networks for diagnostic purposes [87]. Certainly the best approach would be to determine precise parameters. Unfortunately, given the challenges regarding this it appears that it will remain extremely difficult into the foreseeable future. An orthogonal approach, such as *sensitivity analysis*, which evaluate how parameter estimates influence the performance of BioMiner can address these issues. For instance, if the performance of BioMiner is not significantly affected by moderate variations in its parameters, this gives us more confidence in the results produced by BioMiner and is an indication that the parameters are sufficiently precise.

5.1.1 BioMiner system parameters

The parameters in the BioMiner system are currently derived in consultation with collaborating biological domain experts. This was done by reviewing documentation from databases incorporated into the BioMiner federation and determining what types of records would be more preferred from within each database. For example, “Genes” from “EntrezGene” are more preferred if their RefSeq status code is “Reviewed” rather than “Inferred”. According to the EntrezGene documentation, “Reviewed” indicates a higher level of human expert curation than “Inferred” for a particular gene record. Thus, our collaborating biologists would prefer to see a “Reviewed” gene ranked higher than an “Inferred” one. In regards to assigning probabilities, RefSeq status codes were then ranked on a 0.0-1.0 scale and set as parameters in BioMiner (Chapter 4, section 4.3.3). For example RefSeq “Reviewed” is assigned 1.0 and RefSeq “Inferred” is assigned 0.2. The approach of using expert biologists to provide rough estimates to populate BioMiner system parameters was advantageous in that parameters were determined rather quickly (several days). Obtaining probabilistic estimates from experts is concerning however given known issues regarding human judgment and probability [88]. A better approach would be to automatically determine precise probabilities from data, e.g. use “machine learning” approaches [89]. Unfortunately, in regards to learning probabilities for BioMiner, there are serious problems with this approach which make it infeasible. Publications, which are often used to estimate probabilities to populate Bayesian networks, are scarce regarding biological databases in this regard. We currently know of no studies regarding the reliability of RefSeq

status codes for instance. Additionally, if studies did exist, the probabilities reported are often not amenable to incorporation into a probabilistic model [90]. Finally, there is a problem with using the data itself (even though there are copious amounts). Biological databases, although interdependent to a large degree, contain inherent overlaps, redundancies, as well as inconsistencies between biological datum. These problems can mislead machine learning algorithms and are not easily resolvable as data lineage, or provenance, is often not recorded [91]. Additionally, data are sometimes not in a format amenable to computation. Database records in the biomedical domain often contain fields which are essentially narrative free-text. Descriptions of gene function are an example of this [42]. Issues such as these make automated learning of probabilities a very difficult challenge.

5.1.2 How important are precise parameters in the BioMiner system?

Determining precise parameters for the BioMiner system, e.g. learning them from data, is intuitively preferable to rough estimates provided by domain experts. However, given that learning probabilities from data is difficult and that rough estimates are relatively easy to come by we address the challenge of determining precise parameters for BioMiner by reformulating the problem in a different way. In lieu of getting precise parameters we instead try to determine how precise they need to be. Another way to say this is that we are now evaluating how sensitive our system is to our choice of parameters. This is called performing a sensitivity analysis. A sensitivity analysis can be described as the systematic variation of initial probabilities to determine their effects on the systems ability to provide plausible results [87]. Often performed on Bayesian

belief networks in the medical domain, a sensitivity analysis provides insights such as determining which probabilities can be roughly estimated and which need to be determined precisely (if any). For our purposes we want to perform a sensitivity analysis on the BioMiner system to determine if our rough parameter estimates generated in consultation from domain experts are accurate enough to get good results or if more precise parameters must be determined.

5.2 Related Work: Sensitivity analyses in Bayesian Networks

Bayesian belief networks are a form of probabilistic graphs used for reasoning under uncertainty in a particular domain. They are a way to efficiently model probabilistic variables and the interdependencies between them [92]. There are numerous studies regarding them in the medical domain. It has been reported quite often that the performance of Bayesian Networks is surprisingly robust to imprecise parameters, for instance:

- Ng and Abramson found that slight variations to Pathfinder, a system for diagnosing lymph-node diseases, had little impact on its performance [93].
- Coupe et al found that most of the parameters in a system to diagnose Ventricular Septal Defect, a common cardiac anomaly, were rather un-influential to the performance of the system when varied [94].
- Henrion et al reported that systematic “noise” added to the parameters of the Computer-based Patient-Case Simulation (CPCS) expert system did little to affect decisions produced by the system [95].

- Kiersztok and Wang produced a contrary study where Bayesian Networks for plane maintenance exhibited some sensitivity to the choice of parameters [96]. In this study however, the authors were very concerned with the probability of rank-order changes in results produced by the system, a very rigid and sensitive evaluation metric not considered necessary by the other studies.
- MYCIN is a rule-based expert system that preceded Bayesian networks and used probability-based certainty factors. Early studies indicated that MYCIN performed equally well with or without its certainty factors, indicating that they were not overly influential to its performance [97].

As we mentioned previously, obtaining accurate probabilities to populate Bayesian networks is a challenge whether they come from data, literature or human experts [98]. In particular, human estimates of probabilities, which we are using in BioMiner, can be imprecise [88]. To address this problem, researchers can perform a sensitivity analysis on their network [94, 99]. A sensitivity analysis is a method which enables a researcher to determine which parameters in the network are the most influential, i.e. the parameters which need to be elicited with greater precision. A common approach is called a one-way sensitivity analysis [100]. This is a method where a single parameter in the network is varied while all the others remain fixed. In this way, parameters which have the greatest effect on the results of the network can be noted and more precise parameter values can be determined. There have been studies however which have indicated that Bayesian networks can produce plausible results even when using imprecise probabilities for all of their parameters [95, 101]. One study found that belief

networks may be more sensitive than previously believed [96]. This study however was highly concerned with rank-order swaps in results provided by the network, something that was important in their particular application and possibly not in others. In general, Bayesian networks appear to be relatively insensitive to fairly large variations in their probabilities, depending on the choice of evaluation measure. This is potentially good news for us. The rankings produced by BioMiner essentially reduce to inference on a probabilistic graph (similar to a Bayesian network). If Bayesian networks are insensitive to variations in probabilities then this suggests that our default parameters, and rough parameter estimates in general, in BioMiner may be accurate enough to produce plausible results. This is our rationale for performing a sensitivity analysis of BioMiner.

5.3 A Sensitivity Analysis of the BioMiner System

To evaluate our choice of parameters we performed a series of sensitivity analyses on the BioMiner system. As previously mentioned, this sort of analysis is generally performed on Bayesian networks. For our study, in addition to a sensitivity analysis we found it necessary to utilize evaluation methods from the information retrieval domain as well. Sensitivity analyses are performed on systems which produce a probability as a result, e.g. the probability of Hodgkin's lymphoma in the case of lymph node diagnosis in Pathfinder for example. However, BioMiner is different in that it produces ranked lists of gene functions. A plausible result from BioMiner indicates that "correct" functions appear higher in the ranked output list, given a protein sequence of unknown function as a query. In other words, we want to measure how well our system ranks a

set of correct functions (genes may have multiple functions) in a list, which may contain many incorrect functions. The appropriate measure here is average precision, an evaluation metric recognized as reasonable by the information retrieval community [102]. So, the question we now ask is: “How sensitive (or robust) is the average precision of the BioMiner system to imprecise probabilities?” If, as in the case of Bayesian networks, BioMiner is insensitive to imprecise probabilities, at least for most of its parameters, then the difficulty of determining precise probabilities can be alleviated (or avoided altogether). Our approach for evaluating the BioMiner system which utilizes the unique combination of approaches described here has, to our knowledge, not been attempted before and represents a novel contribution.

5.3.1 Perturbations on BioMiner system parameters

In a sensitivity analysis on Bayesian networks each probability is systematically varied, or perturbed, to determine its effect on the networks result. For the BioMiner system, this must be performed on each of the four parameters (Ps, Qs, Pr, Qr) for each database in its federation. For our sensitivity analysis study, there are four databases, with parameters that are perturbed, in BioMiner. This means that 16 database-parameter combinations are in play (4 databases * 4 parameters) for each perturbation and test query. If there are twenty test queries corresponding to 20 genes and one perturbation for example, then this amounts to 320 different sets of sensitivity analysis results for a one-way sensitivity analysis (see 5.4.3). The result set expands further if one looks at perturbing multiple parameters at once rather than looking at them one-at-a-time.. If

there are lots database-parameter combinations, perturbations, or test queries, the total number of sensitivity analysis results can grow rather quickly.

5.3.2 *Evaluation measures*

Our sensitivity analysis requires three evaluation measures: precision, recall, and average precision [103]. These measures are quite common in the information retrieval literature [103], but we provide definitions here for clarity. Precision is the percentage of documents retrieved by the system that are relevant to the query. In the case of the BioMiner system this can be described as the fraction of relevant functions for a particular unknown gene over the total number of functions retrieved. The definition of precision is:

$$precision = \frac{relevant \cap retrieved}{retrieved} \quad (5.1)$$

Recall is the percentage of relevant documents actually retrieved. In the case BioMiner, this is the fraction of relevant functions for a particular unknown gene. The definition of recall is:

$$recall = \frac{relevant \cap retrieved}{relevant} \quad (5.2)$$

Precision and Recall are measures used on sets of documents. What we want is a measure that works on ranked lists. Average precision is a measure used to determine if relevant documents occur higher in a list of results. The assumption, from the information retrieval literature, is that a user will prefer that relevant documents occur sooner (to avoid having to search a long list). In regards to the BioMiner system, this

would mean that relevant functions would occur near the top of its results list. In this study, we usually measure average precision at 100% recall (although this is not a requirement of the average precision calculation). An example of an average precision calculation is given in Table 5.1, and the formal definition is:

$$ap = \frac{\sum_{r=1}^N (\textit{precision}(r) \times \textit{relevant}(r))}{\# \textit{relevant}} \quad (5.3)$$

Where $\textit{precision}(r)$ is the precision at result r , and $\textit{relevant}(r)$ is a function which simply determines whether result r is relevant or not [103]. Average precision can be further summarized as *macro-average precision*, which is the mean value of the individual average precision of each query. Macro-average precision is also known as *mean average precision*.

Table 5.1: Example of an average precision calculation. There are eight total results of which four are relevant. Four is also the total number of relevant results (100% recall). For this result set the average precision is $(1.00+0.67+0.75+0.5)/4 = 0.73$. If the average precision of this result set is averaged with other result sets, the measure is called *macro-average precision*. (Diagram inspired by Callan, 2007).

Rank	Relevant	Precision	Recall
1	Y	(1/1)=1.00	0.25
2	N	(1/2)=0.50	0.25
3	Y	(2/3)=0.67	0.5
4	Y	(3/4)=0.75	0.75
5	N	(3/5)=0.60	0.75
6	N	(3/6)=0.50	0.75
7	N	(3/7)=0.43	0.75
8	Y	(4/8)=0.50	1.00

5.3.3 Reference standard: well-annotated proteins

Evaluation measures used in this study require some method for determining whether or not a particular result is relevant to a given query. In the case of the BioMiner system this would be a set of genes (proteins) for which functions are known with high confidence. Generally speaking, genes cannot be assigned functions with 100% certainty, but biologists prefer certain types of evidence over others. Function assignment using direct experimentation and published in a scientific journal rather than prediction methods using computer algorithms (e.g. BLAST) are more highly trusted for example. In addition, for our study genes must have multiple functions as the average precision measure should not be used when there is only a single relevant answer. Also, function descriptions should be made using a controlled vocabulary (the Gene Ontology in our case). Controlled function descriptions facilitate automatic calculation of relevance for a particular result in a given query. This is necessary for our study as the number of experiments and calculation makes manual assessment of evaluation measures infeasible.

For our study we decided to use twenty well-known genes identified from Entrez or the GeneTests database [104] which were assigned functions using terms from the Gene Ontology (GO) by the Protein Information Resource (PIR [85, 105]). Assigned GO functions from PIR are associated with evidence codes which describe the type of evidence used to determine function. These evidence codes are also part of GO. It is therefore possible, in this case, to determine if functions were assigned using direct experimentation or predictive methods (Table 5.2). The evidence for function

assignment in this set of twenty genes is much more enriched for direct experimentation than non-electronic sources. The functions assigned to these genes were deemed as “relevant” results from the system when queried with each particular gene. We should also note that the PIR is not incorporated into the BioMiner database federation.

Table 5.2: 20 genes with reliable function assignment to be used to query the BioMiner system. Note that, on average 46.6% of the function evidence is from “non-computational” (or “non-electronic”) sources (%Non-Computational), such as direct experimentation. This is in contrast to estimates of about 5% overall in protein databases [4]. Non-Computational annotations are generally believed to be more reliable by the biological community. To see the full gene records for each gene, go to: <http://pir.georgetown.edu/cgi-bin/ipcEntry?id=XX> where “XX” is the IProClass identifier.

Gene Name	IProClass Identifier	# Functions	% Non-Computational
ABCC8	Q09428	13	38.4% (5/13)
ABCD1	P3387	15	46.7% (7/15)
AGPAT2	O15120	10	50.0% (5/10)
ATP1A2	P50993	31	32.2% (10/31)
ATP7A	Q04656	35	51.4% (18/35)
CFTR	P13569	19	47.4% (9/19)
CNTS	O60931	8	50.0% (4/8)
DARE	Q9V3T9	18	83.3% (15/18)
EIF2B1	Q14232	11	86.7% (13/15)
EYA1	Q99502	12	16.7% (2/12)
FGFR3	P22607	16	50.0% (8/16)
GALT	P07902	8	12.5% (1/8)
GCH1	P30793	10	40.0% (4/10)
GLDC	P23378	7	42.9% (3/7)
GNE	Q9Y223	13	38.5% (5/13)
LPL	P06858	13	30.8% (4/13)
MLH1	P40692	19	21.1% (4/19)
MUTL	Q68FG1	13	69.2% (9/13)
RYR2	Q92736	18	27.8% (5/18)
SLC17A5	Q9NRA2	13	38.5% (5/13)

5.4 Study Evaluation Protocol

The general sensitivity analysis protocol in our study is to evaluate the average precision of the BioMiner system under various perturbations of the probabilistic values of its parameters. A perturbation in our case simply means introducing some sort of variation to the default probabilistic values. Twenty well-annotated genes from the PIR database serve as queries to the system as well as sets of plausible (e.g. “correct”) answers to determine the relevance of each result output by the system. Function comparison is made using GO terms to facilitate automation, which is necessary given the large number of analyses. Finally, average and macro-average precision is calculated for the each of the result sets from each perturbation for each of the twenty queries.

5.4.1 Selected protein databases

The protein databases incorporated into the BioMiner system which are relevant to our sensitivity analysis study are: Pfam [20], TIGRFAM [21], Entrez [63], and Gennav [84]. The Pfam, TIGRFAM, and Entrez databases provide necessary data (such as protein sequences) as well as function assignments. The Gennav database contains GO terms. Function assignments in Pfam, TIGRFAM, and Entrez described using GO terms actually contain GO identifiers, which serve as “pointers” to function descriptions in GO. GO terms (e.g. gene functions) can achieve higher relevance scores (and thus better rankings) from BioMiner depending on the quality of the path from the query node to the GO term or if multiple results from protein databases point to the same GO term (i.e. multiple paths) (Figure 5.1). These protein databases represent a subset of all

the protein databases incorporated into BioMiner (Chapter 4, section 4.3.1), as not all utilize GO terms. Utilization of GO terms to describe gene function is somewhat sparse and some protein databases incorporated in the BioMiner had to be excluded from the sensitivity analysis study because of this.

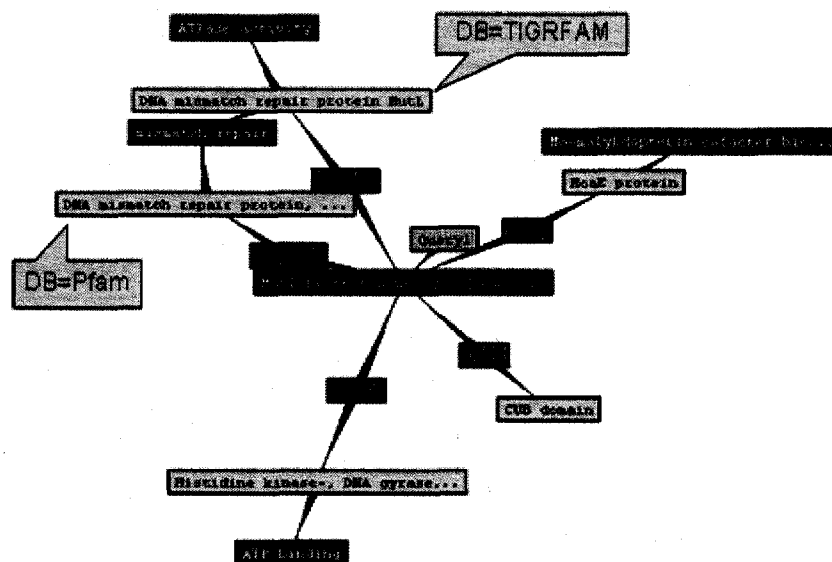


Figure 5.1: Example of a result “graph” from the BioMiner system. The green nodes represent GO terms, as queried in the Gennav database. The “Query1” node represents the initial starting query, a gene of unknown function for example. In this case the GO function “mismatch repair” is pointed to by results from both Pfam and TIGRFAM. It is therefore more likely that “mismatch repair” will achieve a higher relevance score (and higher ranking) than the other three GO terms given that “mismatch repair” has a greater number of paths to it than the other GO terms.

5.4.2 Sensitivity analyses

Evaluation studies of various perturbations on the average precision of the BioMiner system fell into three classifications:

- 1) One-way sensitivity analysis (see 5.4.3) . This is a systematic variation of all probabilistic values for all parameters in the BioMiner system in one-at-a-time

fashion and judging its effect on the average precision of the system. It is a common method for performing sensitivity analysis in Bayesian Networks.

- 2) Multi-way sensitivity analysis (see 5.4.4). This study systematically introduces variations to all of the default probabilities in the BioMiner system simultaneously to judge its effect on the average precision of the system. Only a subset of possible variations can be addressed, as discussed below.
- 3) Sensitivity analysis under random assignment of parameter values.(see 5.4.4)
This looks at the average precision of the BioMiner system when all parameters are assigned completely random probabilistic values and also when relevant results occur randomly in a result list (independent of the system). This is one of the boundary conditions in our study and is necessary to evaluation the impact of the one-way and multi-way sensitivity analyses.

Formulas for each type of study, where necessary, are defined in the next section.

5.4.3 One-way Sensitivity Analysis Perturbations

To perform our one-way sensitivity analysis, we developed several formulas for varying each parameter in the BioMiner system. Many parameters, especially Ps's & Qs's, have default values equal to 1.0. For these parameters, the values were systematically varied between 0.0 and 1.0 in 0.1 increments, with the average precision calculated after each increment. Other parameters are essentially lookup tables, with values assigned to particular values. RefSeq status codes from Entrez Gene records and GO evidence codes fall into this category. Finally, there are several parameters which represent a quantitative similarity score (e.g. BLAST, or HMM e-values). The default parameters

for lookup tables or similarity scores were calculated by using a conversion function. For instance, a text value (e.g. status code), or similarity score (e.g. e-value) was converted into a probabilistic value (between 0.0 & 1.0) (Chapter 4, section 4.3.3). The functions used to create the default parameters behaved linearly. To perform perturbations on these parameters, new functions were written which behaved in non-linear fashion, e.g. rose from 0.0 to 1.0 either more slowly or quickly than the linear function. Note that these perturbations do not include “swaps”, e.g. the relative rankings remain the same but the numerical distance between them may increase or decrease. All perturbation functions are provided here. There are two which apply to lookup tables (RefSeq status codes and GO evidence codes), and one which applies to similarity functions (BLAST, Pfam, and TIGRFAM e-values). They are labeled as “default”, “low”, and “high”. Low and high refer to perturbation functions which rise from 0.0 to 1.0 more slowly and more quickly than the default linear function. The RefSeq status code perturbation functions require a status code *c* as input and then map the code to a probabilistic value. In the following list, each probabilistic value maps to the status code: “WGS”, “GENOME ANNOTATION”, “INFERRED”, “MODEL”, “PREDICTED”, “PROVISIONAL”, “VALIDATED”, and “REVIEWED” respectively:

- 1) Default (linear): 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
- 2) Low: 0.01, 0.05, 0.1, 0.15, 0.2, 0.5, 0.8, 1.0
- 3) High: 0.01, 0.2, 0.4, 0.6, 0.8, 0.85, 0.9, 1.0

The GO evidence code perturbation functions also require an evidence code *c* as input and also map the code to a probabilistic value. In the following list, each probabilistic

value maps to the evidence code: “NR”, “ND”, “IEA”, “NAS”, “IC”, “IEP”, “ISS”, “RCA”, “IGI”, “IMP”, “IPT”, “IDA”, and “TAS” respectively:

- 1) Default (linear): 0.16, 0.23, 0.3, 0.37, 0.44, 0.51, 0.58, 0.65, 0.72, 0.79, 0.86, 0.93, 1.0
- 2) Low: 0.03, 0.05, 0.07, 0.09, 0.1, 0.15, 0.21, 0.3, 0.42, 0.67, 0.85, 0.95, 1.0
- 3) High: 0.04, 0.08, 0.15, 0.42, 0.6, 0.75, 0.84, 0.89, 0.94, 0.97, 0.98, 0.99, 1.0

The similarity score perturbation functions require an e-value e as input and then return a probabilistic score. They are defined as:

$$Default(linear) = abs\left(\frac{\log_{10}(e)}{300}\right) \quad (5.4)$$

$$Low = \frac{1}{(\log_{10}(e) * -1 + 300)} \quad (5.5)$$

$$High = 1 - \frac{1}{\log_{10}(e)} \quad (5.6)$$

5.4.4 Multi-way Sensitivity Analysis Perturbations

Our multi-way sensitivity analysis differs from the traditional multi-way sensitivity analysis in the Bayesian networks literature which typically involves systematically perturbing various combinations of parameters. Given that the cross-product of parameters can become extremely large, at least in our case, we made the decision to introduce variations to all parameters in the BioMiner system simultaneously. We performed this by introducing random “noise” to all default parameters. To achieve this

we followed a method proposed by [95] where normally distributed random noise is added to a log-odds probability (e.g. parameter) and then converted back to a probability. This approach avoids the need for range checks as the new probabilities are never greater than 1.0 or less than 0.0. It also has the added benefit of being able to control the amount of noise added, which depends on the standard deviation parameter. The function to add noise is given by:

$$p' = Lo^{-1}[Lo(p) + e], e = Normal(0, \sigma) \quad (5.7)$$

Where Lo refers to log-odds probability and Lo^{-1} is a function which converts a log-odds probability back into a probability. We performed six separate random noise studies where noise of 0.2, 0.5, 0.8, 1.0, 2.0, and 3.0 standard deviations was added to the default parameters. Adding noise with standard deviations of greater than 2.0 has the effect of setting most parameter values near 1.0 or 0.0. Figure 5.2 illustrates the effect of adding log-odds normal noise to a probability of 0.8. Since this approach depends on randomization, each experiment is repeated 100 times and the average is reported (an experiment refers to a gene/perturbation combination). The one caveat in our case is that many of the default parameters have a value of 1.0, which fails in this formula here. To work around this we decided that any of the BioMiner system parameters greater equal to 1.0 would be re-set to 0.99 before noise was added. In the perturbation function, values very near 1.0 stay close to 1.0 even after addition of substantial noise (Figure 5.3).

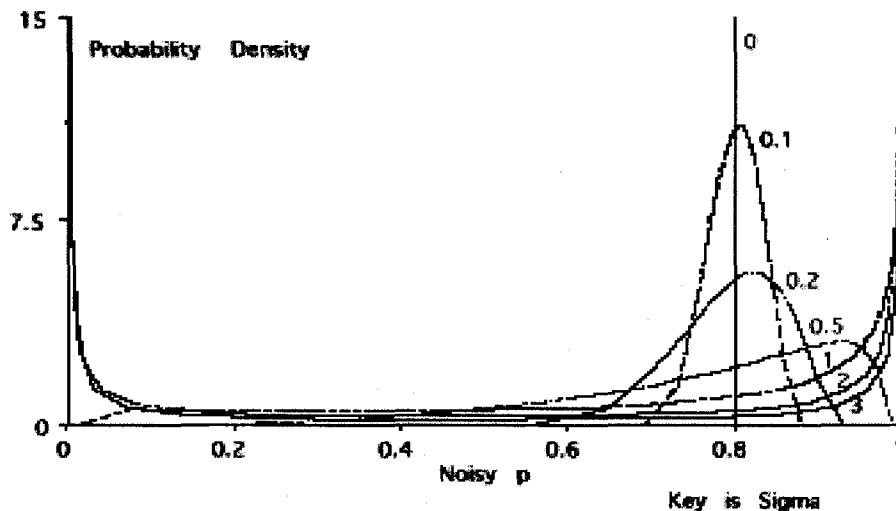


Figure 5.2: Effect of adding log-odds normal noise to a default parameter of 0.8. A standard deviation of 0.1 (blue), 0.2 (pink), or 0.5 (green) in this figure indicates a unimodal density function. Standard deviations greater than 1.0 indicate bimodal density functions with most values near 0.0 or 1.0 (black, red, blue). The formula for adding log-odds normal noise and graph are both courtesy of [95].

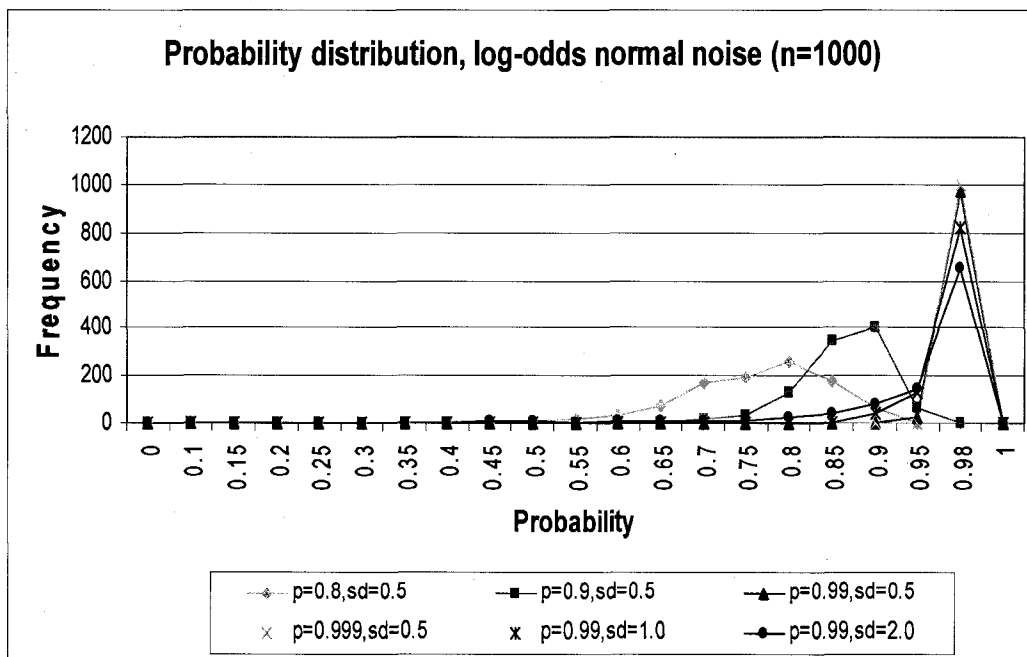


Figure 5.3: Distribution of perturbed probabilities when the input probability is near or very near 1.0, under various standard deviations. If the input probability is very near 1.0, the perturbed probability also stays near 1.0, even after addition of noise at 2.0 standard deviations.

We also decided to evaluate the average precision of the system under complete randomization. In our first study, we assigned all parameters in BioMiner a random probabilistic value, for each particular query. The system then calculated relevance scores and output ranked result lists, on which average precision was calculated. The second study does not involve using the system at all but simply calculates average precision mathematically. Average precision depends on the number of results in a ranked list as well as the number of relevant results. The calculation can be performed by randomly assigning where k relevant items occur in a result list of size n and calculating the average precision. This process should be simulated a large number of times (e.g. 1000) and the random average precision should be the average of the average precision of the simulations. Intuitively the average precision of the system for both randomization studies should be very similar but we performed both methods to increase validity of the results.

5.4.5 Random and Worst-Case Performance

Finally, we introduce a calculation to determine the average precision in the worst-case. The worst-case is where all relevant results occur at the bottom of the results list. Calculation of worst-case average precision (wap) depends on both the number of results from BioMiner (n) as well as the number of relevant results (k) for a particular query, much like in our calculation of random average precision. The formula (courtesy of Wolfgang Gatterbauer) for wap (which assumes 100% recall) is:

$$wap = \frac{1}{k} \sum_{i=1}^k \frac{i}{n-k+i} \quad (5.8)$$

5.4.6 Calculating average precision

For each of the perturbations in each study, the twenty genes were posed as queries to the system and ranked lists of GO terms produced as results. Each of these result sets were compared against the GO terms assigned by PIR and the average precision determined. For each study, the macro-average precision, or average of the average precision of each of the twenty genes, was calculated [103]. Two sample t-tests were also employed to check for significance in average precision for the twenty genes between separate perturbation results in some cases.

5.5 Results

This section describes the results of all sensitivity analyses performed. As an initial benchmark, the macro-average precision of the BioMiner system was determined to be 0.837. This is for the set of 20 genes posed as queries to the system and using PIR assigned functions to determine relevant results.

5.5.1 One-way sensitivity analyses

For the sensitivity analysis study there are four databases in the BioMiner system under consideration as well as the four basic parameters (Ps, Pr, Qs, Qr). This is a total of 16 database-parameter combinations. The average precision for each of the twenty queries was determined for each of the 16 combinations under each perturbation and the macro-

average precision calculated. Figure 5.4 shows aggregated results from one-way sensitivity analyses for simple 0.0-1.0 perturbations (section 5.4.3). There were eleven database-parameter combinations of this type. Given that there are eleven perturbations (0.0-1.0 in 0.1 increments), this resulted in 121 macro-average precision calculations for simple 0.0-1.0 perturbations. In general, the macro-average precision of the system remained stable with a macro-average precision near 0.8, with the exception of parameter values less than 0.2.

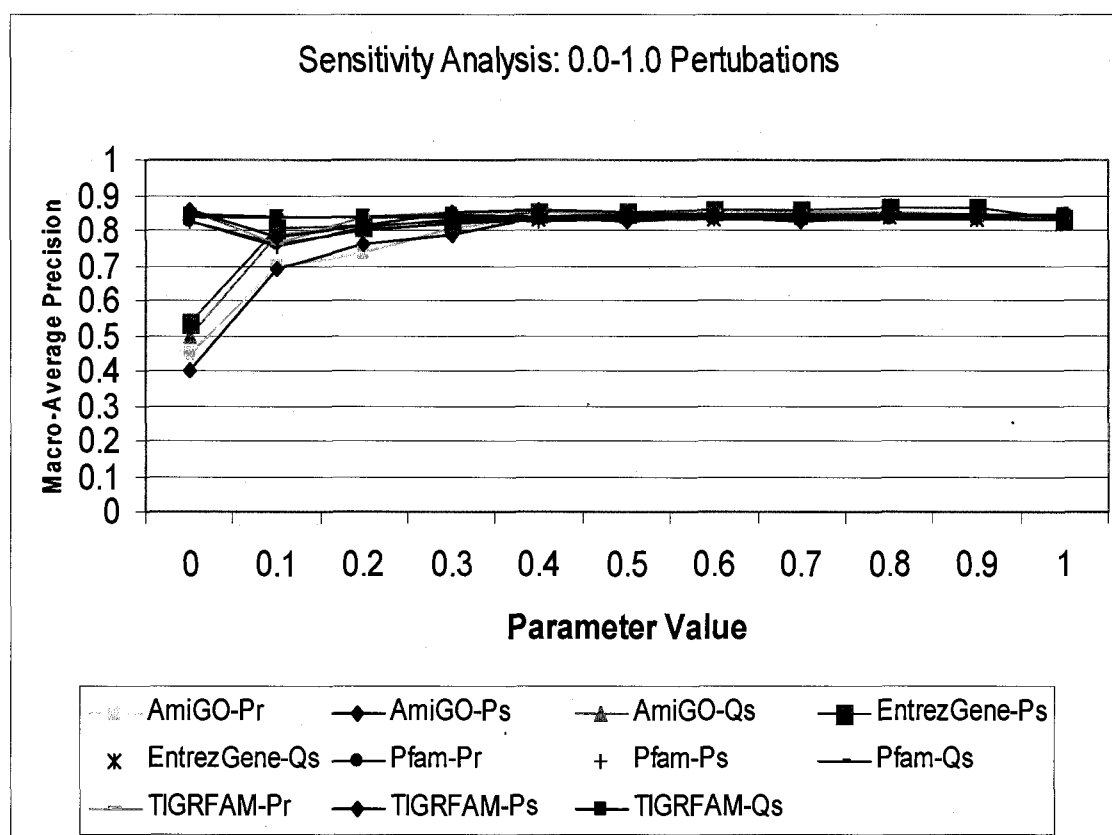


Figure 5.4: One-way sensitivity analysis for eleven database-parameter combinations of simple 0.0-1.0 perturbations. These perturbations are described in section 5.4.3. These are single-parameter perturbations, such as varying the Ps parameter from 0.0 to 1.0 in 0.1 increments for one data source. The labels stand for database/uncertainty metric combinations. For example, EntrezGene-Qs is the perturbation for the EntrezGene database and Qs parameter. The X-axis represents the perturbation, i.e. value of the parameter (uncertainty metric). The Y-axis is the *macro-average precision*, or the average of the average precision for each gene under that perturbation. Overall, the macro-average precision of the system under the various perturbations remains near 0.8, which is very close to the macro-average

precision of the system under default parameters (0.837). The exception is when parameters values are less 0.2.

One-way sensitivity analyses for function or lookup table perturbations are shown on Figure 5.5. There were five database-parameter combinations of this type. Given that there are three perturbations, this resulted in 15 macro-average precision calculations. In all, there were 136 (121+15) macro-average precision calculations each of which involves determining the average precision for 20 queries (genes). This amounts to 2720 total one-way sensitivity analysis experiments. For the vast majority of cases in our one-way sensitivity analysis, the macro-average precision stayed fairly close to the default macro-average precision of 0.837.

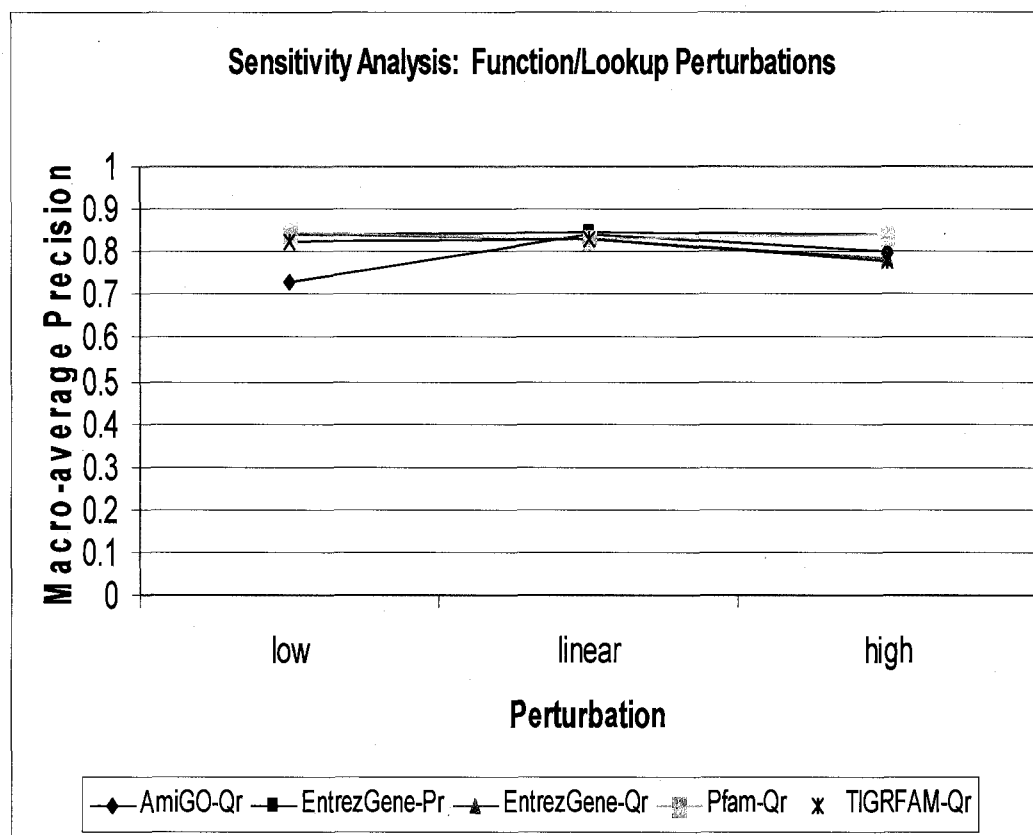


Figure 5.5: One-way sensitivity analysis for the five perturbations involving a function or lookup table. These perturbations are described in section 5.4.3. Low, linear, and, high refer to the properties of the functions used in this perturbation, i.e. increasing to 1.0 at a slower or faster rate. The labels stand for database/uncertainty metric combinations. For example, EntrezGene-Pr represents the perturbation for the EntrezGene database and Pr metric. The X-axis represents the perturbation, i.e. the value of the parameter (uncertainty metric). The Y-axis is the *macro-average precision*, or the average of the average precision of each gene under that perturbation. The macro-average precision is stable in most cases with a value near 0.8, near the macro-average precision of the system under default parameters (0.837).

The BioMiner system appeared to be extremely robust under all one-way sensitivity analysis studies, except for the case where parameter values for database-parameter pairs were extremely low (less than 0.2). Given that the system seemed extremely robust to variations in single parameters we wondered to what degree parameter values matter at all. To test this we attempted a much more severe perturbation. We created a function that would convert an e-value into a 0.0-1.0 value

in a linear fashion (much like the one described in section 5.4.3) but in an inverted fashion, e.g. better e-values scores would be near 0.0 and worse e-values near 1.0 – exactly the opposite of biological relevance. When this function was introduced as a perturbation, the macro-average precision of the system dropped significantly as compared to the default macro-average precision (0.599 versus 0.837, $p=1.639e-05$). It is encouraging however that BioMiner results appear robust under less-severe variations of single parameters. A more realistic scenario however is that all of the parameters in the system will contain some amount of imprecision. To investigate the effect of perturbations to all parameters in BioMiner we developed a method to perform a multi-way sensitivity analysis.

5.5.2 Multi-way sensitivity analyses

To evaluate the BioMiner system under more realistic situation where all of its parameters contain some imprecision we introduced varying degrees of “noise” to each of the default parameters. Noise in our case refers to log-odds normal noise (5.4.4). We varied the amount of noise to be 0.2, 0.5, 0.8, 1.0, 2.0, and 3.0 standard deviations from the default parameter and the results are shown on Figure 5.6.

The macro-average precision of the system only begins to significantly degrade after introduction of log-odds normal noise at greater than 1.0 standard deviations.

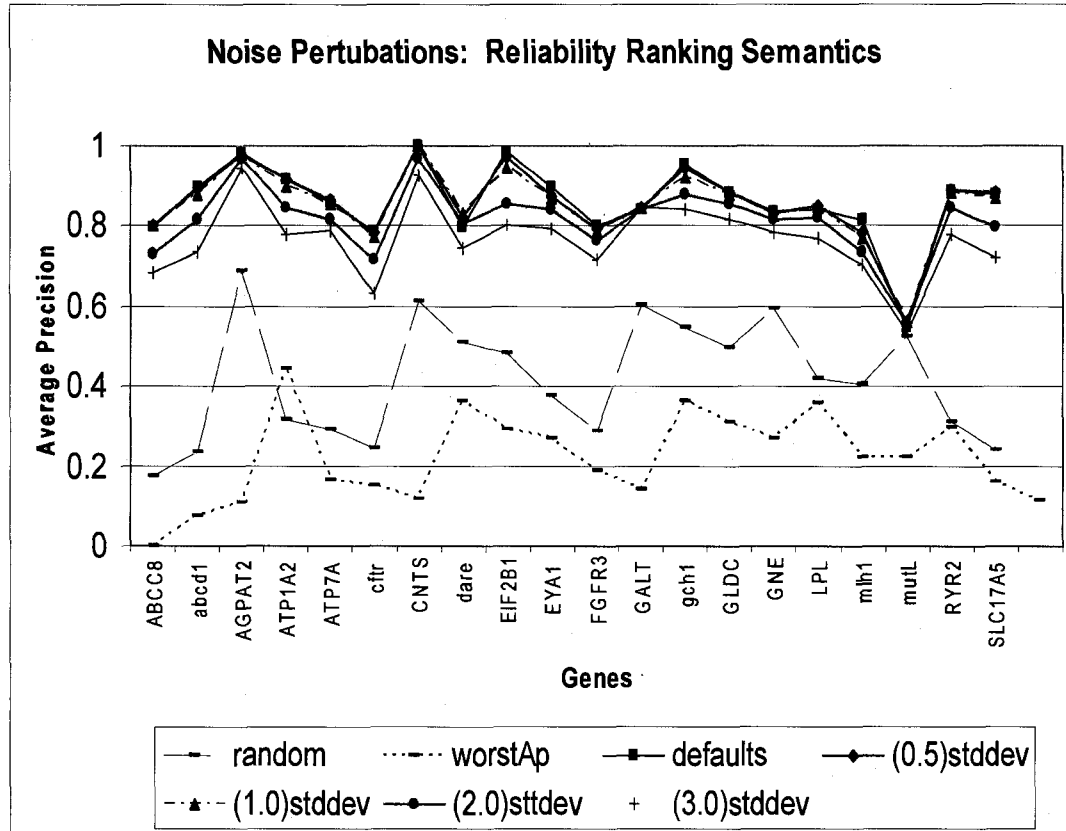


Figure 5.6: The average precision of the BioMiner system after addition of log-odds normal noise to all default parameters at 0.5, 1.0, 2.0, and 3.0 standard deviations for each of the twenty genes in the test set. These perturbations are described in section 5.4.4. The labels show the amount of perturbation for each gene. Also included are the average precision for each gene in the random-case, worst-case, or default parameter values. Random and worstAP refer to our baseline conditions (section 5.4.5). The X-axis represents the genes and the Y-axis represents the average precision for the gene, for various perturbations. The average precision only begins to degrade significantly after addition of log-odds normal noise greater than 1.0 standard deviation. This figure also includes random and worst-case average precision for each gene.

5.5.3 Random assignment of parameter values

For a final sensitivity analysis study, we decided to evaluate the macro-average precision of BioMiner under complete random assignment of its parameter values. In this case, BioMiner must calculate relevance scores and produce a final ranking with the randomly assigned parameter values. We also include a mathematical analysis of the

random average precision of BioMiner results independently of the system, unlike in sections 5.4.3 and 5.4.4. Intuitively, the macro-average precision of both studies should be similar and indeed, they appear to be in this case (Figure 5.7). The macro-average precision of the system under random assignment is significantly worse than the macro average precision of the system with default parameters.

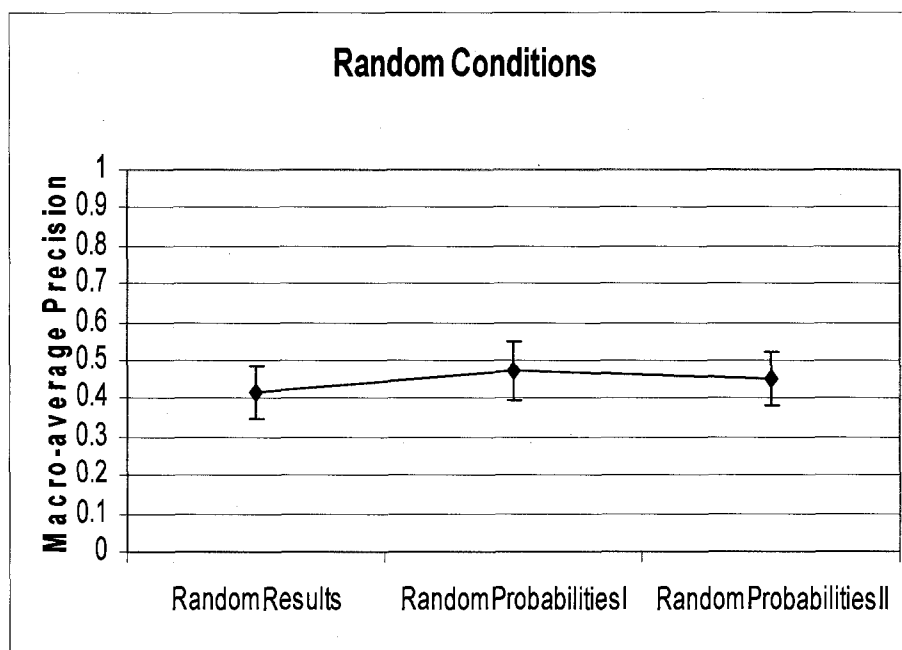


Figure 5.7: The performance of BioMiner under randomized conditions of its parameter values. These are described in section 5.4.5. The X-axis represents three random perturbation studies. RandomI and RandomII are two separate sensitivity analysis result where all parameters in the BioMiner system were randomly assigned probabilistic values. RandomResults were determined mathematically. The Y-axis is the *macro-average precision*, or the average of the average precision of each gene under that perturbation. The macro-average precision of the system under random assignment is significantly worse than the macro-average precision under the default parameters (0.473, 0.450, and 0.418 versus 0.837, $p=5.867e-11$, $p=3.029e-12$, and $p=4.967e-13$, respectively).

5.5.4 Boundary conditions on average precision

To establish baseline performance for the BioMiner system, average precision was calculated in the worst-case (all relevant answers at the bottom of the results list), and random-case, for each of the twenty queries and results were tabulated (Table 5.3). The average precision for default parameters, the number of results, and relevant answers for each query (gene) are included in the table as well. The average precision was calculated under two conditions: 1) at 100% recall, and 2) after 25 results. This was done because the number of results necessary to achieve 100% recall varied considerably for each query (between 8 and 103). Average precision at 25 results was chosen because some queries did not produce very large result sets and we wanted to calculate this metric on a decent number of queries. Intuitively, the average precision under the two conditions should be correlated. Additionally, consider that there is a direct relationship between the average precision calculation (and thus the performance of the system) and the number of results and relevant answers for any particular gene. Note that BioMiner produces 51.8 results for 15.3 relevant answers on average. If 15.3 relevant answers are dispersed randomly in a list of 51.8 results, the average precision will be 0.418 (at 100% recall). Therefore, the average precision of BioMiner should be significantly better than 0.418 to be considered anywhere near a success. This is unlike evaluating web search engines where the ratio between number of results (millions perhaps) and relevant documents (dozens to hundreds) is generally very small. If relevant answers are dispersed randomly in the case of web search engines case the

average precision of the system will be close to 0.0. The results for each of the genes for these three studies are shown in Table 5.3 as well.

Table 5.3: Worst-case (all relevant results at the bottom of the output list), Random (relevant results dispersed randomly in the output list), and average precision with default parameters of the BioMiner system for the twenty queries at 100% recall (@100) and at 25 results (@25). The “Relevant” column refers to the number of plausible functions assigned by PIR. The “Results” column refers to the number of functions in the results list produced by the system. “WC” refers to worst-case, “RND” refers to random-case, and “DEF” refers to default average precision.

Gene	Results	Relevant	WC@100	WC@25	RND@100	RND@25	DEF@100	DEF@25
ABCC8	97	13	0.075	0	0.173	0.065	0.766	0.766
ABCD1	78	15	0.109	0	0.234	0.100	0.887	0.887
AGPAT2	16	10	0.442	na	0.688	na	0.983	na
ATP1A2	108	31	0.164	0	0.316	0.091	0.870	0.894
ATP7A	130	35	0.152	0	0.293	0.072	0.797	0.816
CFTR	90	19	0.119	0	0.248	0.088	0.812	0.822
CNTS	15	8	0.365	na	0.611	na	1.000	na
DARE	39	18	0.290	0.033	0.508	0.335	0.824	0.824
EF2B1	35	15	0.268	0.057	0.482	0.359	0.984	0.984
EYA1	38	12	0.191	0	0.377	0.266	0.787	0.787
FGFR3	65	16	0.142	0	0.288	0.140	0.736	0.790
GALT	15	8	0.365	na	0.605	na	0.841	na
GCH1	21	10	0.312	na	0.548	Na	0.953	na
GLDC	17	7	0.271	na	0.496	na	0.801	na
GNE	24	13	0.360	na	0.596	na	0.823	na
LPL	36	13	0.221	0.018	0.419	0.313	0.836	0.836
MLH1	52	19	0.220	0	0.407	0.217	0.745	0.772
MUTL	28	13	0.297	0.221	0.523	0.475	0.547	0.547
RYR2	66	18	0.158	0	0.310	0.143	0.872	0.934
SLC17A5	66	13	0.113	0	0.241	0.114	0.881	0.927

Macro-average precision was calculated using the results from Table 3. The macro-average precision of BioMiner with default parameters is significantly better than worst-case or random-case at 100% recall (0.837 versus 0.232, $p=7.32e-12$, and 0.418, $p=2.2e-16$ respectively) as well as at 25 results (0.828 versus 0.199, $p=2.289e-13$, and 0.0235, $p < 2.2e-16$, respectively). The macro-average precision of the system

for worst-case, random, and default parameters under both conditions, as well as 95% confidence intervals, are shown on Figure 5.8.

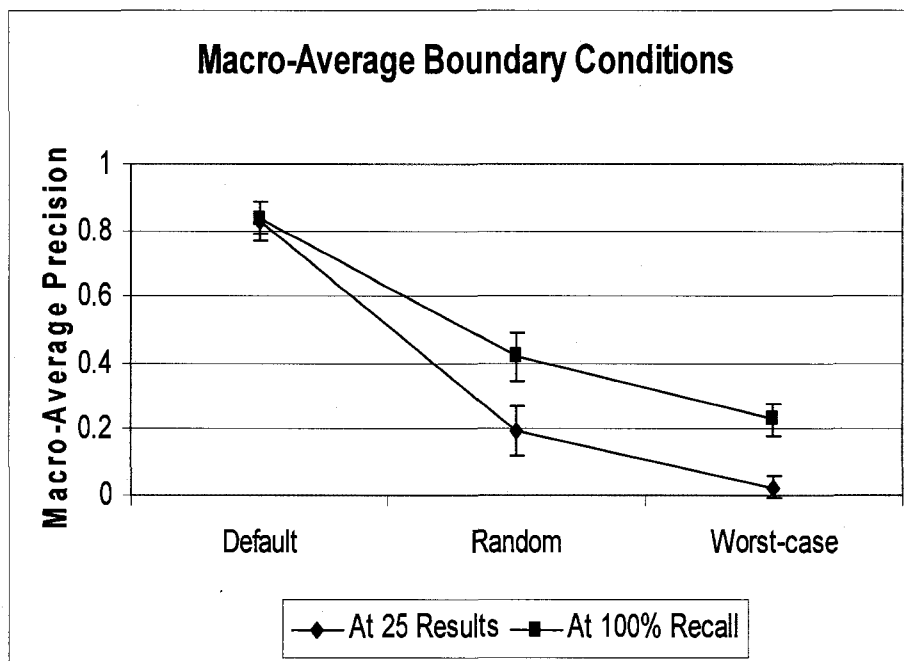


Figure 5.8: Macro-average precision of worst-case, random, and default parameters in the BioMiner system at 100% recall and at 25 results, with 95% confidence intervals. The X-axis represents the perturbation studies and the Y-axis is the *macro-average precision*. The macro-average precision is the average of the average precision of each gene under that perturbation. Random at 100% recall is calculated the same as in section 5.4.5.

5.5.5 Summary of results

Results from the one-way, multi-way, and random average precision studies are summarized on Figure 5.9. Included are default and worst case average precision. The macro-average precision of the BioMiner system remains remarkably stable under various perturbations of single parameters and after the introduction of log-odds normal noise up to 1.0 standard deviation to all the default parameters simultaneously. These

results suggest that significant imprecision can be introduced to the default parameters of BioMiner with little impact on the macro-average precision. It therefore seems increasingly unlikely that parameters need to be determined precisely in BioMiner to get good performance.

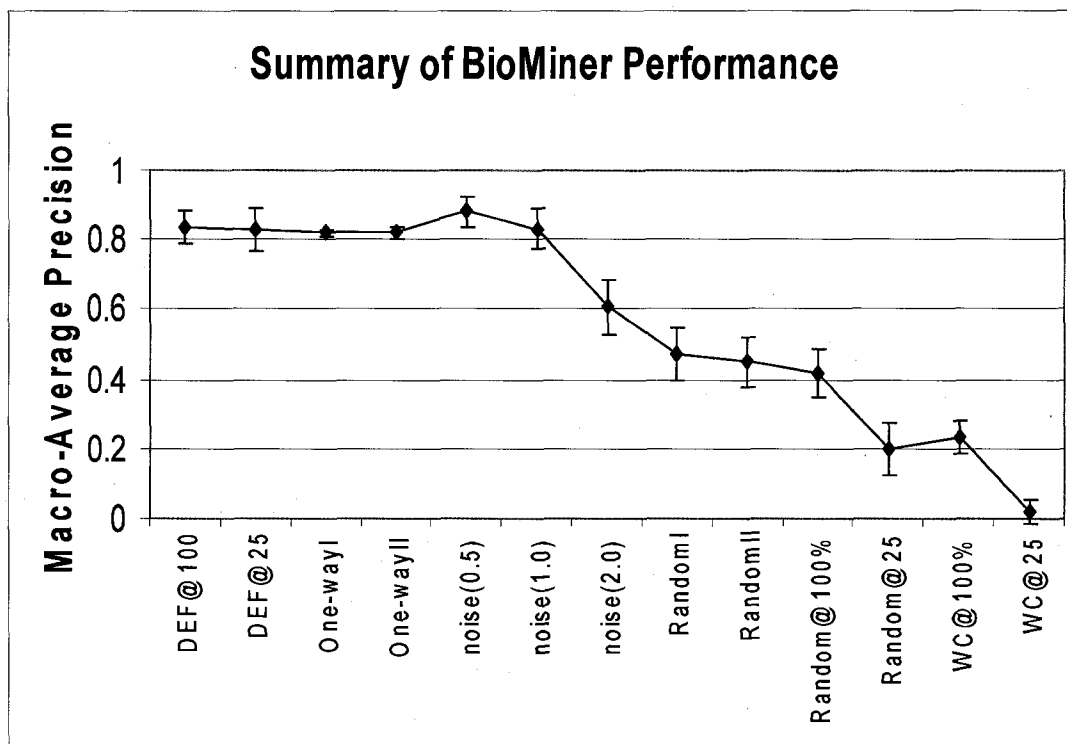


Figure 5.9: Summary of sensitivity analysis results for the BioMiner system. The Y-axis is the *macro-average precision*, which is the average of the average precision of each gene and is a summary measure of the performance of BioMiner. The X-axis represents various sensitivity analysis studies. “DEF@100” and “DEF@25” refers to macro-average precision at 100% recall and at 25 results respectively. If these are not included in the study name then macro-average precision is calculated at 100% recall. “DEF” stands for default parameters, One-wayI stands for the one-way sensitivity analysis under 0.0-1.0 perturbations, which are described in section 5.3.3. One-wayII stands for the one-way sensitivity analysis under function/lookup perturbations which are also described in section 5.3.3. “Noise(0.5-2.0)” are multi-way sensitivity analysis studies, which are described in section 5.3.4. “WC” stands for worst-case, and “RandomI” and “RandomII” are the performance of BioMiner under randomized conditions which are described in section 5.4.5. Results indicate that the performance of BioMiner is remarkably stable under systematic perturbations of its default uncertainty metrics and significantly outperforms simulated randomized or worst-case performance.

5.6 Discussion

BioMiner performs quite well with the default parameters with a macro-average precision of 0.837, especially when compared to the average precision in the worst-case and random scenarios (0.232, and 0.418, respectively). In addition, the performance of BioMiner (evaluated by macro-average precision) remains stable even after applying systematic perturbations to one or all of the values of its default parameters. This addresses the research sub-question regarding the robustness of our uncertainty model (Chapter 1, section 1.2.2). Our studies do indicate that performance can degrade after extremely severe perturbations, such as addition of log-odds normal noise of 2.0 standard deviations or greater (Figure 5.6). However, the degree of noise at this level is an unrealistic scenario as the perturbed values will very likely be near either 1.0 or 0.0. BioMiner is very stable under more realistic noise perturbations at 0.5 standard deviations and below or under virtually any perturbation of a single parameter (Figure 5.9). The results of our study suggest that determining parameter estimates using domain experts are precise enough for BioMiner to perform well and that we can be confident in the function predictions produced by BioMiner.

We believe the reason for the relative insensitivity of the uncertainty model in BioMiner to parameter variations is strongly linked to the topology of the result graph. Consider that we are ranking GO terms which often have multiple paths to them from the query node (Figure 5.1). It is possible then to rank GO terms by “deterministic” (i.e. non-probabilistic) methods which take into account this graph connectivity such as counting the number of incoming edges to a node or the number of paths to a node from

the query. This can also be done in BioMiner by setting all parameters in BioMiner to a single value, say 0.9. According to Network Reliability Theory (NRT), the basis of inference in BioMiner (Chapter 3, section 3.4.4), nodes with a greater number of incoming paths will have a higher reliability score. If all parameters are set to the same values, NRT essentially reduces to simple path-counts (if paths are the same length). As it turns out, the macro-average precision of BioMiner using topology alone (e.g. setting all parameters to 0.9) is basically equivalent that when using the default parameters.(0.838 vs 0.837). While this is an observation and not a mathematical proof, it suggests to us that the insensitivity of BioMiner may have something to do with the topology of the result graphs, e.g. multiple paths may make the network “robust”. This could be the case with Bayesian Networks as well.

That deterministic algorithms perform as well as BioMiner calls into question the value of the uncertainty model in BioMiner (which is probabilistic by nature). However, consider that these are well-studied (well-known) genes and the information about them is highly disseminated, i.e. resides in multiple data sources. This may explain the connectedness we see in our result graphs. An alternative scenario which demonstrates the utility of our uncertainty model is genes, or functions of genes, which are less well-studied (less known). The information for these genes, or functions, may only reside in a single source. The BioMiner result graph in this case would not exhibit the sort of connectivity we see in the more well-studied genes and deterministic methods, such as path or edge-counts, may not perform as well as BioMiner.

As it turns out, this is indeed the case. Upon further inspection of BioMiner result sets, we found eight plausible new functions for four of the original twenty genes. All of these functions are supported by recent publication evidence (Table 5.4). Since they are recent discoveries, it appears that information about these new functions is not well disseminated. Moreover, most of these new functions were ranked very highly by BioMiner and not by deterministic methods (Table 5.5). Probabilistic approaches, such as the uncertainty model in BioMiner, perform better than deterministic ones when information is less well-known or disseminated in multiple data sources. Since it is difficult to determine a-priori if the information about a particular gene is well-known or not, the uncertainty model in BioMiner is advantageous in that it performs well in either case.

Table 5.4: Plausible new functions for four genes from the original sensitivity analysis dataset of 20 genes. These were found by inspecting results from BioMiner and are supported by recent publication evidence.

Gene Name	Evidence (pmids)	New Functions
ABCC8	18025464 (2007)	GO:0006855, GO:0015559, GO:0042493
Cftr	17869070 (2007), 18045536 (2007)	GO:0030321, GO:0042493
EYA1	17637804 (2007)	GP:0007501, GO:0042472
Mlh1	16713580 (2006)	GO:0032137

Table 5.5: Rankings of new functions (from Table 5.1) by BioMiner (Reliability Rank), and two deterministic methods (In-Edge, and Path-Count). In most cases, the new functions are ranked much higher by BioMiner than by the deterministic methods. Since they are ranked much higher in BioMiner it is much more likely that a human user will discover them using BioMiner than with deterministic approaches.

Function (GO)	Gene	Reliability Rank	In-Edge Rank	Path-Count Rank
GO:0006855	ABCC8	21	66	66
GO:0015559	ABCC8	22	67	67
GO:0042493	ABCC8	17	27	30
GO:0032137	mlh1	5	5	4
GO:0030321	cftr	1	23	22
GO:0042493	cftr	24	25	23
GP:0007501	EYA1	4	36	27
GO:0042472	EYA1	14	13	18

Chapter 6: EVALUATION OF BIOMINER

6.1 Multiple Evaluations of the BioMiner System

In this chapter we evaluated protein function predictions produced by BioMiner in three ways. First was a validation of the initial uncertainty metrics of the BioMiner. Second was a pilot study of BioMiner for annotation in which we compare BioMiner results to hand-produced annotations. Finally, and most importantly, against existing and commonly utilized computational annotation methods (see section 6.5).

The BioMiner system is based on new and highly experimental data integration technology that handles uncertainty in data and information. The purpose of the uncertainty functionality is to enable ranking or highlighting of more relevant data, based on parameter values set in the system (Chapter 3, section 3.4.3 and Chapter 4, section 4.3.3). The multiple evaluations carried out here reflect the process from proof-of-concept to utilizing BioMiner for protein annotation and comparing it to existing approaches in a real world-situation. Some of the results of these studies are from [39].

6.1.1 Proof of concept evaluation #1: relevance ranking

This was an initial proof-of-concept evaluation of BioMiner, version 1.0 (Chapter 4, section 4.4.1), for ranking database results, which also represented the first validation of the underlying data integration technology for handling uncertainty (Chapter 3, section 3.4). The evaluation was to determine if BioMiner could correctly rank results of a

single entity type for protein annotation. The uncertainty parameter values were the initial ones chosen in consultation with expert biologists, and not yet evaluated for their precision or robustness, as in a sensitivity analysis (Chapter 5). The relevance score calculation algorithm had not yet been tested on real data and only a portion of the parameter values for BioMiner had been determined. User interfaces were also relatively simple at this point (table view only). The protocol for this study therefore was deliberately kept simple since the behavior of BioMiner was completely unknown at this point.

6.1.2 Proof of concept evaluation #2: protein annotation

This was another proof-of-concept to explore of the ability of BioMiner, version 1.1 (Chapter 4, section 4.4.2), for integrating biological information from multiple databases related to the function of a particular protein, calculate meaningful rankings of results, and utilize them for annotating proteins. The evaluation was to determine if a researcher could create better protein annotations using BioMiner than without (i.e. manually). It had been determined that the initial values for the parameters in BioMiner could correctly rank function predictions from a single database (COG). It was unknown however how well the system would perform (in regards to annotation) under these parameters given that they were estimated in consultation with expert biologists and likely imprecise, and the sensitivity analysis of BioMiner had not yet been performed (Chapter 5). A new, biologically-relevant, user interface was also implemented to facilitate the annotation process (the GGB).

6.1.3 BioMiner evaluation study: hypothetical protein annotation

This was a rigorous and real-world evaluation study of BioMiner, version 1.2 (Chapter 4, section 4.4.3), for annotating hypothetical proteins (e.g. proteins of unknown function). The system had been augmented with what was learned from prior proof of concept evaluations as well as the sensitivity analysis. For this study we compared the results of BioMiner versus two gold-standard sets of annotated, formerly hypothetical proteins. The gold standard sets of proteins were created with an approach that we developed specifically for this study. The evaluation was to compare the performance of BioMiner versus freely-available protein databases, which are the common method for annotating proteins (Chapter 2, section 2.2). In the first part of this evaluation, the parameters in BioMiner were tuned and optimized from the results of the sensitivity analysis (Chapter 5). In the second part of this evaluation, the parameters in BioMiner were trained and optimized further given results from the first part of the evaluation. The procedure we used to optimize the parameters in BioMiner for this evaluation is discussed in the following sections.

6.2 Related Work: Evaluating Computational Annotation Systems

Hypothetical proteins represent the best real-world scenario in which to evaluate computational annotation systems. A difficulty in evaluating these systems on hypothetical proteins however is the overall dearth of gold-standard annotation datasets for true performance assessment [44]. Previous evaluations of computational annotation systems have thus been limited. For example, when assessing the SuperFamily database, Gough et al assigned functions to proteins of formerly unknown

function in multiple genomes [77]. Cadag et al. attempted to demonstrate the benefits of their system by comparing its predictions to the previous Genbank annotations [12]. In both cases, there is no validation of the new functions produced. Our evaluation approach, in which we create “gold-standard” annotation datasets (see 6.5.3), represents another novel contribution of this work by alleviating the issue regarding the lack of gold-standards and allows for a more reliable assessment of performance.

6.3 Proof of Concept Evaluation #1: Relevance Ranking

In this study, BioMiner was evaluated for its ability to correctly classify proteins according to their functional category as defined by the Clusters of Orthologous Groups (COG) [106]. A COG is a family of proteins with similar amino acid sequences who are thought to share direct common ancestry and thus biological function as well. BioMiner performs this by integrating the COG database [82] into its federation. In the COG database, proteins are grouped into COGs based on similarity criteria. In the unicellular (prokaryotic) version, there are over 5000 individual COGs. Proteins in a particular COG are all believed to perform the same biological function which can be very specific for the individual COGs, such a particular biochemical reaction. Each individual COG however is further classified at a higher level into what is known as a COG functional category. There are 25 COG functional categories which describe very general biological functions such as “Transcription”, or “Energy production and conversion”. The purpose of this study was to determine if BioMiner could correctly

classify a particular set of proteins by their COG functional category, which was previously published in [39].

6.3.1 Rationale for proof of concept evaluation #1

The ability of BioMiner, as well as its underlying technology (UII), to properly rank information based on the values of its uncertainty parameters had not been previously evaluated. This study was therefore meant to be an initial “sanity-check” to validate the preliminary uncertainty parameters in BioMiner as well as its results according to some relevant biological metric (i.e. COG functional categorization).

6.3.2 BioMiner system version

The BioMiner system used in this study is version 1.0 (Chapter 4, section 4.4.1) and corresponds to that published in [39]. There were multiple data sources integrated into the BioMiner federation but only results of a single mediated-schema entity type was evaluated (Ortholog), which is not found in many data sources. In addition, although several uncertainty parameters had been implemented, only the one relevant parameter which sets uncertainty values for the COG database was under consideration for this study. Ranked results were inspected in the “table” view of the graphical-user-interface (GUI) (Chapter 4, section 4.3.4).

6.3.3 Reference standard

Evaluation of the ranked results from BioMiner required the development of a reference standard for comparison. In this case, the reference standard was 32 prokaryotic proteins with known COG functional categories as determined by a collaborating

biologist. This is a common way to develop comparison sets in the biological domain as it alleviates potential errors which could be caused by completely automated methods [107]. While this method is reliable, it should not be considered as a gold-standard. A true gold-standard in biology requires some form of direct experimentation on a protein to determine function, such as biochemical assays. Unfortunately, direct experimentation is infeasible in most cases due to time and resource limitations leaving manual expert curation as the next best option.

6.3.4 Study protocol

The protocol used in this evaluation was to determine if BioMiner could rank each of the 32 proteins in the comparison set correctly by its COG functional group. Each of the 32 proteins was submitted as a query to BioMiner and data returned from all databases in the federation. The results from the COG database were ranked by relevance score and viewed in the GUI. The id number of the top-ranked COG result from the system was inspected and its functional category determined from the COG database. If the functional category from BioMiner matched that of the query protein then this was scored as “Agree”. If the functional categories did not match then it was scored as “Disagree”.

6.3.5 Results

Results for the 32 comparison proteins submitted to BioMiner were tabulated. Of the 32 comparison proteins, 14 produced only a single Ortholog result from BioMiner. These were omitted from the final analysis since there were not multiple results to rank,

although the system was correct in all cases. The initial agreement results indicated that the system was able to categorize 77.8% (14/18) of proteins by their correct COG classification, i.e. the top-ranking Ortholog in BioMiner was the correct COG classification for the protein.

6.3.6 Discussion

Upon further review it turned out that three cases where BioMiner categorized the COG category “incorrectly” corresponded to proteins which were actually assigned to multiple COG functional categories. This was an oversight when the comparison set was created as it was assumed that a protein could only belong to a single COG functional category. As it turned out, the system was able to properly categorize at least one of the correct COG functional categories for the protein for each of the three cases, increasing its agreement with the comparison set to 94.4% (17/18).

The single remaining missed case was inspected further to attempt to determine why it was not ranked correctly by BioMiner. It turned out to be related to the relevance score calculation algorithm (Chapter 3, section 3.4.4). The relevance score calculation algorithm uses an approximation method to generate scores which is dependant on a “trial” parameter. The higher this parameter is set the more accurate the scores and subsequent accuracy of the final rankings. Generally speaking, higher accuracy is needed in the case where there are a lot of results to rank which have scores that are extremely close together. The single missed case in this study did indeed have a lot of results from the COG database (nearly 100, by far the most), which had scores very close together. When the trial parameter was increased however (from 1,000 to

50,000 trials) BioMiner was correctly able to classify the protein by its correct COG functional category. This suggests that the trial parameter should be increased in order to have sufficient accuracy when there are lots of results. This does have an impact on the performance of the system in terms of the time it takes to calculate relevance scores. However, the increase in time to calculate relevance scores is still a small fraction of the time it takes to integrate the data (internal observation).

These results for the COG study were not obtained from a true formal evaluation and represent only a preliminary investigation into the performance of BioMiner. However, these results do represent an important proof-of-concept in that a data integration system with uncertainty functionality, can correctly rank information queried from a biomedical database, something that had yet to be demonstrated.

6.4 Proof of Concept Evaluation #2: protein annotation

The previous evaluation demonstrated that BioMiner has the potential to categorize proteins by broad functional class (COG category). Protein annotation however is more useful if more specific functionality can be determined. This often requires manual searching of multiple databases by a biological researcher as well as compilation and inspection of results. Indeed, protein annotation remains a highly manual endeavor [44, 108], advances in automatic methods notwithstanding (Chapter 2, section 2.3). Data and information integration is the first step in annotating proteins [109] and is where much of the manual effort is concentrated. Therefore, this next evaluation was meant to demonstrate proof of concept and explore capabilities of BioMiner for improving the manual process of annotating proteins.

6.4.1 Rationale for proof of concept evaluation #2

This proof of concept evaluation was an attempt to determine if a human expert using BioMiner could produce annotations which match or improve upon those produced using manual methods alone. Improved annotations would be an indicator of the value of a larger search space as well as an integrated view of the data. However, even if annotations produced using BioMiner are only equivalent to those produced manually, a time and labor savings can be assumed due to automation.

6.4.2 BioMiner system version

The BioMiner system used in this evaluation was version 1.1 (see 4.3.2), and included all default parameters, data sources, and the “Generic-Genome-Browser”, an interface for viewing biological sequence data [76]. A critical functionality of the GGB is the ability to rank, highlight, or filter results based on a particular score. For our purposes, we implemented ranking and highlighting functionality based on relevance score (section 6.4.4, Figure 6.1). Also, the trial parameter of relevance score calculation algorithm was increased to 10,000 to increase ranking accuracy given the results from the prior study.

6.4.3 Reference standard

The reference standard utilized here are protein annotations produced manually by domain experts. For this we utilized a functional annotation study regarding the *Shewanella oneidensis* bacterium where annotations were created manually by biological domain experts [5]. Experts utilized multiple protein databases which

included complementary sources of information including homology, genome co-localization, and protein interactions. To compare against annotations produced using BioMiner we selected 41 reference proteins from the *S. oneidensis* study whose annotations were considered by the authors to be of high confidence (Appendix A). However, while these annotations can be considered reliable, they still do not represent a true gold-standard since no direct experimentation was utilized to determine their function. For instance, it's possible that this reference set could still suffer from a major source of annotation error: incorrect database records [48].

6.4.4 Study protocol

First, we describe the heuristics that were used to annotate proteins based on results from the BioMiner system. Protein annotation is a bit of an art which can be difficult to describe in a logical series of steps, although our heuristics cover most cases here. Generally, annotations were derived first from the set of top-ranked functional domains or structures, and finally BLAST/PSI-BLAST results. In some cases however, a functional domain could be selected if it spanned the greatest length of the query protein but was not top-ranked. The final decision was left to the discretion of the expert annotator. An example annotation case using output from the system is shown in Figure 6.1.

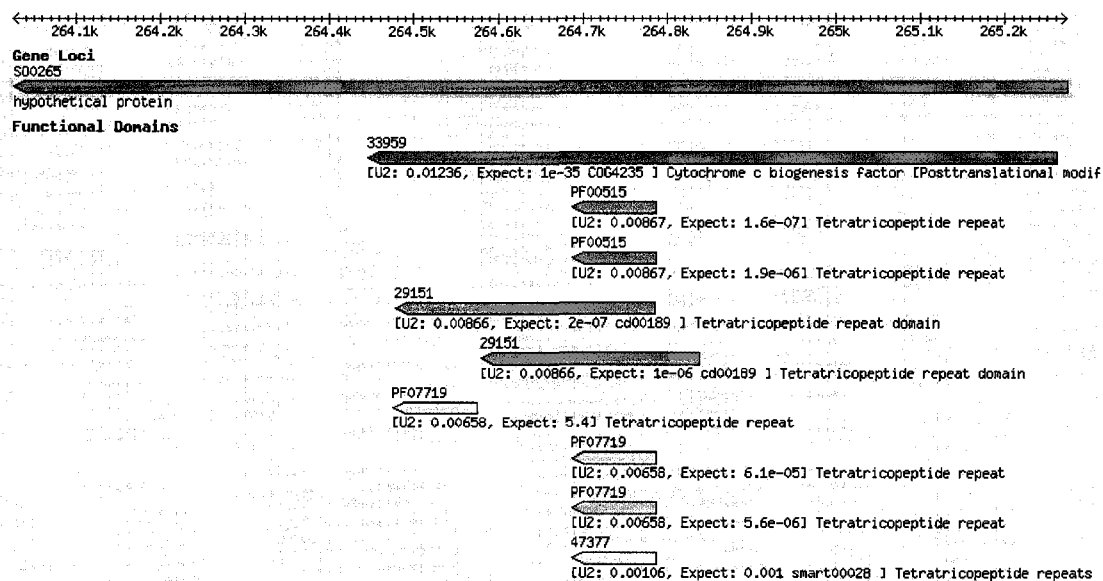


Figure 6.1: Results for the BioMiner system for *S. oneidensis* locus (gene) SO0265 displayed in the GGB with results from various databases (under “Functional Domains”) ranked and highlighted according to their relevance scores. The highest ranking result from BioMiner is “Cytochrome c biogenesis factor” (COG4235). It also spans the greatest length of the query protein as compared to the other functional domains. The BioMiner annotation produced in this case would indeed be “Cytochrome c biogenesis factor”, which agrees with the manually produced annotation.

To evaluate the accuracy and quality of the annotations produced using BioMiner we compared them against the manually produced annotations and scored them according to a method adapted from [12]. Domain experts did the scoring. BioMiner annotations could be deemed inferior to the manual annotations, equivalent, or superior. Superior cases are scored “+1”, inferior “-1”, and equivalent “0” (see Table 6.1 for examples) An inferior case generally refers to an incorrect annotation or what is called an over-prediction, i.e. an enzymatic activity that is too specific. A superior case generally means that an annotation is has been determined to be more accurate or more information, such as an additional biological function can be attributed.

6.4.5 Results

Annotations produced using results from the BioMiner system were compared against the manually produced ones and scored. In 30/41 (73.2%) of cases, the annotations agreed completely. In 4/41 (9.8%) the manual annotations were superior and in 5/41 (12.2%), BioMiner system-based annotations were superior (as judged by the expert reviewer). In two cases, BioMiner annotations could not be performed because the functional evidence was extremely conflicting. It could not be determined how decisions regarding the manual annotations were made, so these remained unresolved. BioMiner annotations considered inferior were either over-predictions (too-specific) or incorrect. In most cases where BioMiner annotations were considered superior it was due to an additional function (e.g. the manually assigned functions were a subset of these). Table 6.1 provides the list of proteins where BioMiner annotations were considered superior or inferior, as well as two examples of equivalent annotations.

Table 6.1: List of BioMiner system-based annotations in *S. oneidensis* deemed superior or inferior to the manually produced ones. In most of these cases the tool-based annotations suggested an additional function, the manual annotations could not be ruled out. Two cases of equivalent annotations are provided as well for illustrative purposes.

Locus	Score	BioMiner Annotation	Manual Annotation
SO1597	1	Dioxygenases related to 2-nitropropane dioxygenase	Omega-3 polyunsaturated fatty acid synthase, PfaD subunit
SO1789	1	Metallo-dependant phosphatases	UDP-2,3-diacylglucosamine hydrolase
SO0363	1	UTP-glucose-1-phosphate uridylyltransferase	Nucleoside-diphosphate-sugar pyrophosphorylase
SO0471	1	2-nitropropane dioxygenase	Flavin-dependant dioxygenase
SO3668	1	Putative heme degradation protein	Heme iron utilization protein HugZ
SO4413	-1	Selenocysteine lyase	Kynureninase
SO1267	-1	Peptidase C26	Glutamine synthetase-associated glutamine amidotransferase
SO4690	-1	4-amino-4-deoxy-L-arabinose transferase and related glycosyltransferases	Undecaprenyl phosphate-sugar: lipid A glycosyltransferase
SO0152	-1	Zn-dependant exopeptidases	Carboxypeptidase
SO0887	0	Porphyromonas-type peptidyl-arginine deiminase	Peptidylarginine deiminase
SO1523	0	ATP-NAD kinase	NAD kinase

6.4.6 Discussion

The process of undertaking this proof of concept annotation study uncovered a couple of key challenges which we discuss here. First, it is only fair to compare any annotations produced using protein databases if they are created at the same time. For instance, the 14.6% (6/14) of BioMiner produced annotations (4 inferior + 2 excluding due to conflicting evidence) which did not match the manually produced ones might be explained by the two-year time difference in when they were created. Protein databases

change their content often so it is difficult to keep manually produced annotations up-to-date given the time and resource allocation required to produce them [43].

The second challenge is more important but also more difficult to address. It revolves around the age-old question: “What is the right answer?” The objective of the BioMiner is to improve protein annotation, not necessarily achieve agreement with human experts. For instance, we assume that BioMiner produced annotations which disagreed with the manually produced ones are incorrect. In reality, BioMiner annotations could be correct given that they were produced with more up-to-date information. There were also cases during the annotation process where results from BioMiner were extremely difficult to interpret. Figure 6.2 illustrates a case where the BioMiner produced one annotation, the manual annotation produces another, and automated results using BLAST searches produces yet another. Which is the correct one, if any?

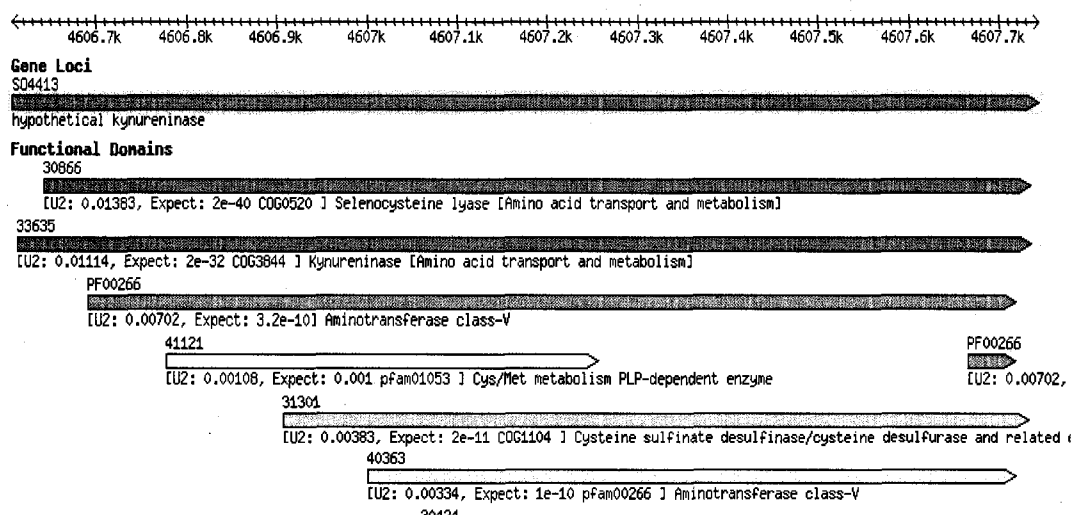


Figure 6.2: The *S. oneidensis* locus (gene) SO4413 with results from the BioMiner system displayed in the GGB. The BioMiner produced annotation is “Selenocysteine lyase” (COG0520), but the manually produced annotation is “Kynureninase” (COG3844). When this protein is subjected to a BLAST/PSI-BLAST search however, most results are annotated as “Aminotransferase” or “Cysteine desulfurase” (COG1104).

Given the limitations identified in this evaluation the results in this evaluation should be taken purely as qualitative and not indicative of the future performance BioMiner. However, it did illuminate two key challenges which must be addressed in future studies of this type. First, annotations created by any method which utilized protein databases should be created *at the same time* to avoid possible inconsistencies if protein databases change their content. Second, an *extremely reliable and independent reference set*, as close to a gold-standard as possible, is needed to help resolve ambiguities and demonstrate performance improvement. An understanding of how this reference set is created is also necessary to ensure reproducibility. Both of these challenges were addressed in our next, and final, evaluation study regarding the annotation of hypothetical proteins.

6.5 BioMiner Evaluation Study: hypothetical protein annotation

This is a study which prospectively evaluates BioMiner versus existing computational annotation systems (protein databases) in their ability to predict the function of hypothetical proteins, an important real-world scenario. It addresses the need to do comparisons between annotations which are simultaneously produced by BioMiner and the protein databases, which eliminates the possibility of underlying data updates (in protein databases) being the reason for difference in prediction quality. This evaluation study does not completely resolve issues regarding the lack of true annotation gold-standards for evaluation purposes but does make significant contributions in that direction. Each system is judged by its ability to annotate proteins of unknown

function (e.g. hypothetical proteins), which addresses a common and important challenge in biomedical research today [8]. For this study we developed two gold-standard reference annotation sets and evaluated the accuracy of function predictions from BioMiner and the protein databases on both sets. In the first part of the evaluation, BioMiner was evaluated using the initial parameters for its uncertainty metrics (tuned slightly from the sensitivity analysis). A critical point here is that, in this default set of uncertainty parameters in BioMiner, results from protein databases are all preferred equally. However, in the first part of the evaluation we observed that the accuracy of protein databases in annotating hypothetical proteins varies considerably (section 6.5.5). By using accuracy results of protein databases from the first part of this evaluation, we were able to train BioMiner to prefer results from more accurate protein databases, where possible. This corresponds to optimizing the “Ps” parameter (Chapter 3, section 3.4.3). The second part this evaluation investigates this trained BioMiner system on a different set of gold-standard reference set to (potentially) observe improvements in its annotation accuracy and compares the accuracy of the trained BioMiner system versus protein databases.

6.5.1 Rationale for hypothetical protein annotation study

The purpose of the BioMiner system is to improve the computational annotation of hypothetical proteins. It therefore should be evaluated against comparable existing systems. Currently, the common systems for annotating proteins are protein databases, of which there are many to choose from. We evaluated the function predictions from BioMiner versus those from several of most familiar protein databases to our biological

collaborators. In addition, we incorporated some upgrades the system (section 6.5.2) which we felt would enhance its performance given the evaluations performed thus far. This study also addresses the key limitations of the prior annotation study to make it much more rigorous. First, all annotations for all methods were created in the same time frame. Second, a simple heuristic for annotating proteins was implemented which eliminated the need for human experts to produce the annotation. A simple heuristic was also necessary to ensure consistency and reproducibility. Finally, we produced two “gold” standard reference sets of annotated proteins to reduce the ambiguities that can arise when comparing methods and to provide a true measure of performance. This gold standard reference sets are 30 proteins from *Shewanella oneidensis*, and 38 proteins from five other bacteria similar to *Shewanella*. All were formerly hypothetical proteins. To create the gold standard reference we developed a novel method which we describe here as well (Appendix B).

6.5.2 BioMiner system version

Two big changes were made to the BioMiner system for this study. These changes were based on an evaluation study of 12 common protein databases in regards to their agreement with a gold-standard annotation reference set created by the authors (see section 6.4.3 and this section). The TIGRFAM [21] and PIRSFScan [85] databases were also added to the BioMiner federation.

The first part of this study provided something very important in regards to BioMiner. The accuracy of protein databases for annotating proteins can vary greatly (between 40% and 100%), suggesting that certain databases generally provide more

precise results (section 6.5.5). TIGRFAM and PIRSFScan were added to the BioMiner federation due to their high accuracy and conditional accuracy respectively, in regards to annotating proteins in the first gold-standard set of 30 proteins from *S. oneidensis*. Given that BioMiner incorporates data from many of these protein databases, it should consider the accuracy of each protein databases when it ranks its final result lists. Accuracies of protein databases, determined in the first part of this study from the 30 proteins in *S. oneidensis*, were therefore input as “Ps” uncertainty parameters in BioMiner (see Chapter 4, section 4.3.3) for corresponding databases in its federation for the second part of this study which evaluated the function predictions of the trained BioMiner system on the second gold-standard set of 38 proteins in five different bacteria. Note that “accuracy” in this case actually refers to “conditional accuracy” as defined in section 6.5.4.

6.5.3 Reference standard

The reference standard in this study approaches that of a gold-standard. The true gold-standard for protein function assignment is direct experimentation, such as biochemical assays for instance. Our method is separated from this by one degree of separation. It is based on high amino acid sequence similarity between a hypothetical protein and an experimentally characterized one. The experimental characterizations come from published literature (PubMed). Once experimentally characterized proteins are identified, specific steps are applied to assign function to hypothetical proteins. These steps can be found in Appendix B.

While this method is not a true gold-standard as no direct experimentation is performed it avoids the main potential error which could arise in our previous reference standards: incorrect annotations in protein database records [48]. Our method also helps to ensure a high degree of reproducibility, something that can be difficult to achieve when proteins are annotated manually by domain experts (section 6.4). In addition our method addresses a common barrier to evaluation of annotation methods, which is the lack of gold standards [44], and thus has more general application. By using our method we were able to create 30 annotations for formerly hypothetical proteins in *S. oneidensis*, and 38 annotations in five other bacteria. These were used as a reference standard in this study (see Appendix A). Note that it is common for research groups to study and annotate proteins in single organisms [5, 110].

6.5.4 Study protocol

The amino acid sequence of each protein in the gold standard sets were submitted to the BioMiner system as well as twelve common protein databases which are often used for annotation. All sources were queried in the month of October 2007. The protein databases and their versions, if indicated, were BLOCKS v14.3 [111], Clusters of Orthologous Groups (COG) v.1.00 and Protein Clusters (PRK) v1.00 [82, 112], the Conserved Domain Database (CDD) v2.12 [32], InterPro [72] v16.1, Pfam [20] v22.0, PIRSFScan [85], SuperFamily [77] v1.69, SMART [113] v5.1, SwissProt [29] release 54.4, TIGRFAM [21] v7.0, and UniProt [19] release 37.4. For each source, only the top-ranked prediction by e-value was used to compare against the gold-standard

annotation for each particular protein. This greatly reduced uncertainty in the annotation process as well as ensured reproducibility, unlike in our prior study where domain experts performed the annotation. Unlike in the sensitivity analysis, GO terms were not utilized. Instead, descriptions from Functional Domain or Family entities (as classified by mediated schema) were evaluated instead.

In the evaluation, function predictions from each source could either “Agree” or “Disagree” with the gold-standard or be “Indeterminate.” Agree is where the protein database (PD) agrees with the gold standard function (GS), or $PD = GS$. Indeterminate is where the protein database returned no results or no function was specified by the PD (e.g. “hypothetical” or “unknown” function). Disagree is where the $PD \neq GS$ and $PD \neq \text{Indeterminate}$. Accuracy (ACC) is the number of “Agrees” over the total number of gold-standard proteins (30). Conditional Accuracy (CACC) is the number of “Agrees” over “Agrees” + “Disagrees”. ACC can be seen as analogous to coverage, or the ability to annotate high percentage of a given set of proteins. CACC can be seen as more akin to precision, e.g. the coverage of the database may be low but annotations produced by it tend to be correct. The evaluation is the comparison of accuracy and conditional accuracy of each database as well as BioMiner.

It is sometimes challenging to compare annotation results from different protein databases in that function descriptions can be heterogeneous, which makes it difficult to determine agreement in some cases. We accounted for synonyms in enzyme names by utilizing the Brenda database [114] and biological function description using the Gene Ontology whenever this was possible. Ultimately we decided the best approach was to

remain as true as possible to the gold-standard description of function and judge agreement very strictly. For instance, the result from a protein database may have described an enzyme function on different hierarchical level than that of the gold standard or may have only indicated a common structural fold. While these sorts of results are not necessarily incorrect from a biological perspective, we scored them as “Disagree”.

Finally, note that a difficulty in evaluating protein databases is that some are aggregates, e.g. are composed of multiple component databases. The aggregate databases this study are CDD, InterPro, TIGRFAM, and SMART. The interface to some of these allows the user to select from among component databases. This is how we got results from COG and PRK for instance, which are components of CDD. Decomposing and evaluating all aggregate databases can be onerous, InterPro has 15 component databases for example. We chose to evaluate TIGRFAM and InterPro in two ways: 1) with all component databases, and 2) with Pfam removed, i.e. TIGRFAM(-Pfam) and InterPro(-Pfam). We evaluated CDD with all components and also COG and PRK separately. We did this because Pfam, COG, and PRK both perform well according to the gold-standard and we wanted to observe any benefit of aggregating databases.

To test for significant differences between the ability of computational annotation systems to assign function to proteins in the gold-standard datasets correctly, McNemar tests were employed. A McNemar test is a form of chi-square test for matched pair data. The matched pairs in this case are function predictions from two

computational annotation systems which agree with the gold-standard function or not in a 1 or 0 categorization. For example (1,0) could mean that the BioMiner prediction agreed with the gold-standard function whereas the prediction from Pfam did not, for a single protein in the gold-standard dataset. These are tabulated on a 2x2 table on which the McNemar test is performed.

6.5.5 Results: Gold-standard reference set #1 (30 proteins)

Tabulated results for all protein databases as well as for the BioMiner system for each protein in the gold-standard are shown on Table 6.2. The accuracy of BioMiner, under the initial default metrics was 70.0% and the conditional accuracy is 87.5%. This is better than the best performing protein database, albeit only by a slight margin.

However, after training the BioMiner system using the conditional accuracies of each individual protein databases, the accuracy of BioMiner increases to 83.3%, which is much better than all other protein databases in this study. In addition, the conditional accuracy of BioMiner is 96.2%, which was better than all but TIGRFAM(-Pfam) and PIRSFScan. The accuracy of BioMiner however was much better than these two.

Table 6.2: Evaluation metrics describing the annotation Accuracy (ACC) and Conditional Accuracy (CACC) for 12 common protein databases and for the BioMiner system, both untrained (Defaults), and trained (Optimized). The trained BioMiner system outperforms all other protein databases in regards to ACC and all but TIGRFAM(-Pfam) and PIRSF in terms of CACC (see “Ranks”). The untrained BioMiner still outperforms all databases in terms of ACC (although only by one “Agree”), but is ranked fourth in CACC (highlighted cells). The difference in ACC and CACC for the trained and untrained BioMiner illustrate the improvements gained through training the system.

System or Database	Agree	Disagree	Indeter.	ACC (Rank)	CACC (Rank)
BioMiner (Optimized)	25	1	4	83.3% (1)	96.2% (2)
BioMiner (Defaults)	21	3	6	70.0% (1)	87.5% (4)
TIGRFAM	20	7	3	66.7% (2)	74.1% (6)
CDD	19	5	6	63.3% (3)	79.2% (5)
SwissProt (BLAST)	19	2	9	63.3% (3)	90.5% (3)
InterPro	18	10	2	60.0% (4)	64.3% (10)
PRK	18	1	11	60.0% (4)	94.7% (2)
UniProt (BLAST)	18	9	3	60.0% (4)	66.% (9)
UniProt (PSI-BLAST)	16	11	3	53.3% (5)	68.0% (8)
Pfam	15	10	5	50.0% (6)	60.0% (12)
COG	14	9	7	46.7% (7)	60.9% (11)
SMART	14	11	5	46.7% (7)	56.0% (13)
InterPro (-Pfam)	13	6	11	43.3% (8)	68.4% (7)
TIGRFAM (-Pfam)	12	0	18	40.0% (9)	100.0% (1)
SuperFamily	10	15	5	33.3% (10)	40.0% (15)
BLOCKS	7	10	13	23.3% (11)	41.2% (14)
PIRSFScan	5	0	25	16.7% (12)	100.0% (1)

In addition, results from McNemar Tests indicate that the difference in accuracy between the trained BioMiner system and the protein databases is significant, except in the case of TIGRFAM and CDD (Table 6.3). There is the potential issue of over-fitting however given that these results for BioMiner were obtained from the same set on which it was trained. This problem is addressed in the next section. Annotations

next section. Annotations produced by the BioMiner and how they scored for each of the gold-standard proteins can be seen on Table 6.4.

Table 6.3: McNemar tests between the trained BioMiner system and all other protein databases. The purpose of the McNemar test is to determine differences in the ability of two databases to produce annotations which agree with the gold standard. P-values in all cases indicate significant difference between BioMiner and most other protein databases, the exceptions being TIGRFAM and CDD.

Database	McNemar Test Result
TIGRFAM	$p=0.0736$
CDD	$p=0.1138$
SwissProt (BLAST)	$p=0.0412$
InterPro	$p=0.0233$
PRK	$p=0.0233$
UniProt (BLAST)	$p=0.0233$
UniProt (PSI-BLAST)	$p=0.0269$
Pfam	$p=0.0044$
COG	$p=0.0098$
SMART	$p=0.0026$
InterPro (-Pfam)	$p=0.0015$
TIGRFAM (-Pfam)	$p=0.0009$
SuperFamily	$p=0.0007$
BLOCKS	$p=0.0001$
PIRSFScan	$p=0.0000$

Table 6.4: Annotation results from the trained BioMiner system for the first gold-standard reference set of 30 proteins in *S. oneidensis* in terms of Agree(1), Disagree(0), or Intermittent(-1) with the gold-standard. Gold-standard functions for these proteins can be found in Appendix A.

Locus	Score	BioMiner Annotation
SO_0342	1	Probable AcnD-accessory protein PrpF (TIGR02334)
SO_0506	1	UbiD family decarboxylases (TIGR00148)
SO_0887	1	Porphyromonas-type peptidyl-arginine deiminase (UniProt:A4SRQ5_AERS4)
SO_1313	1	Anhydro-N-acetylmuramic acid kinase (PIRSF500155)
SO_1523	1	ATP-NAD(H) kinase (PIRSF500155)
SO_1597	1	PfaD family protein (TIGR02814)
SO_1608	1	7-cyano-7-deazaguanine reductase (TIGR03138)
SO_1789	1	UDP-2,3-diacetylglucosamine hydrolase (TIGR01854)
SO_1851	1	Possible SAM-dependant methyltransferase (UniProt:Q4QP66_HAEI8)
SO_1963	1	Homogentisate 1,2-dioxygenase (UniProt:Q12M82_SHEDO)
SO_2042	-1	Hypothetical protein (PRK05363)
SO_2043	-1	Ferric reductase domain protein (UniProt:A0KXE3_SHESA)
SO_2593	1	Glutamate dehydrogenase (PIRSF036761)
SO_2603	0	GAF domain-containing protein (UniProt:Q8D9F7_VBVU)
SO_2614	1	Aminodeoxychorismate lyase (UniProt:Q0HVX7_SHESR)
SO_2627	1	ATP-dependant Clp protease ClpS (UniProt:A3D5F3_9GAMM)
SO_3014	1	Segregation and condensation protein B (UniProt:Q4KGA7_PSEF5)
SO_3015	1	Chromosome segregation and condensation protein ScpA (UniProt:A5W0B2_PSEPU)
SO_3367	1	TRNA (Guanine-N(7))-methyltransferase (UniProt:A1EPZ4_VBCH)
SO_3436	1	TRNA pseudouridine synthase D, TruD (UniProt:A0KU86_SHESA)
SO_3542	1	Phosphoketolase (PIRSF017245)
SO_3578	-1	UPF0124 protein yfiH (UniProt:YFIH_ECOLI)
SO_3367	-1	Uncharacterized protein with pyridoxamine 5'-phosphate oxidase domain (PIRSF004633)
SO_3668	1	HutX protein (UniProt:A3E9A9_VBCH)
SO_3957	1	3-deoxy-D-manno-octulosonate 8-phosphate phosphatase (PIRSF006118)
SO_4227	1	S-adenosyl-L-methionine dependant methyltransferase MraW type (PIRSF004486)
SO_4398	1	D-tyrosyl-tRNA (Tyr) deacylase (TIGR00256)
SO_4413	1	Kynureninase (UniProt:A0TQL3_9BURK)
SO_4677	1	3-deoxy-D-manno-octulosonic-acid kinase (PRK01723)
SO_4680	1	CDP-glycerol:poly(Glycerophosphate) glycerophosphotransferase (UniProt:A5NDA1_9GAMM)

6.5.6 Results: Gold-standard annotation set #2 (38 proteins)

There is a concern with the results in section 6.5.5. The problem is that the BioMiner system was tested on the same set of proteins with which it was trained, i.e. the conditional accuracies of the protein databases in *S. oneidensis* were utilized as “Ps” values in BioMiner. The trained BioMiner was then run on the same set of *S.*

oneidensis proteins. There is thus the possibility of over-fitting to the test set. To address this we created a new gold-standard set of 38 proteins, using our gold-standard method (see Appendix B), which can be found in Appendix A. Unlike the previous gold-standard set, these proteins come from several different types of bacteria. This potentially provides a greater diversity of protein types. Note that SO_0887 is included in both gold-standard sets, but with different annotations. It appears that SO_0887 is similar to two enzymes with very similar functions. Depending on your biological point-of-view both functions are plausible. It is however, slightly more similar to the “agmatine deiminase” enzyme, which is the assigned function in this case.

These new proteins were submitted to BioMiner as well a subset of the protein databases evaluated in section 6.5.5, the best performing ones in terms of overall accuracy. The results are shown in Table 6.5 and Table 6.6. BioMiner still provides the highest function prediction accuracy and McNemar tests are significant in all cases, demonstrating that the uncertainty metrics are not over-fit to the training data. Moreover, correct results from the BioMiner originated from five different data sources: Pfam (3), PRK (5), TIGRFAM (6), PIRSF (1), and cdd (2) (Table 6.7). As a cautionary note, BioMiner appears to perform better without incorporating PSI-BLAST results. PSI-BLAST differs somewhat from the other protein databases in BioMiner in that protein model descriptors are created automatically using the query protein. This approach may be less accurate and could increase the amount of “noise” in BioMiner results. Both versions of BioMiner, however, outperform all other protein databases in this study.

Table 6.5: Evaluation metrics for a subset of common protein databases from section 6.5.5 and for the trained BioMiner system on the new gold-standard set of 38 proteins. Metrics are described using the same accuracy metrics as in section 6.5.5 (Accuracy (ACC), and Conditional Accuracy (CACC)). Note the improvement between the trained BioMiner system here and the untrained system in Table 6.2.

System	Agree	Disagree	Indeter.	ACC (Rank)	CACC (Rank)
BioMiner	26	7	5	68.4% (1)	78.8% (1)
PRK	19	7	12	50.0% (2)	73.1% (2)
COG	14	17	6	36.8% (3)	45.2% (5)
TIGRFAM	13	8	15	34.2% (4)	61.9% (3)
PSI-BLAST	13	12	6	34.2% (5)	52.0% (4)
InterPro	13	19	4	34.2% (6)	40.6% (6)
Pfam	11	19	2	28.9% (7)	36.7% (7)

Table 6.6: McNemar tests of BioMiner (trained using conditional accuracy results from *S. oneidensis*) versus the protein databases. Results are for 38 new gold-standard annotations (in 5 organisms). P-values are significant in all cases.

Database	McNemar Test Result
Pfam	p=0.001
COG	p=0.014
PRK	p=0.023
TIGRFAM	p=0.002
UniProt (PSI-BLAST)	p=0.043
InterPro	p=0.010

Table 6.7: Annotation results produced by the trained BioMiner system for the 38 new gold standard proteins. In this result set, the top-hits in BioMiner originate from five different databases: (Pfam = 4, Protein Clusters (PRK) = 8, TIGRFAM = 9, PIRSF = 4, and cdd = 1).

Locus	Score	Original Database	BioMiner Annotation
SO_0025	Agree	Protein Clusters(PRK11104)	hemG, protoporphyrinogen oxidase
SO_0599	Agree	PIRSF (PIRSF006361)	ATPase
SO_0706	Agree	TIGRFAM (TIGR03071)	couple_hipA: HipA N-terminal domain
SO_0828	Agree	Protein Clusters (PRK09489)	rsmC, 16S ribosomal RNA m2G1207 methyltransferase.
SO_0887	Disagree	Pfam (PF04371)	PAD_porph, Porphyrin-type peptidyl-arginine deiminase
SO_1267	Agree	Protein Clusters (PRK11366)	puuD, gamma-glutamyl-gamma-aminobutyrate hydrolase
SO_1431	Agree	Pfam (PF06192)	TorD: Cytoplasmic chaperone TorD
SO_2484	Indeter	Protein Clusters (PRK03826)	hypothetical protein
SO_3967	Indeter	Protein Clusters (PRK03537)	hypothetical protein
SO_4537.2	Disagree	COG (COG0612)	Pqql, Predicted Zn-dependent peptidases
SO_0946	Agree	TIGRFAM (TIGR01730)	RND_mfp: efflux transporter, RND family, MF
spr0592	Disagree	COG (COG4221)	Short-chain alcohol dehydrogenase of unknown specificity
spr1622	Agree	Pfam (PF08270)	PRD_Mga, M protein trans-acting positive regulator (MGA) PRD domain
spr1332	Agree	Protein Clusters (PRK11565)	dkgA, 2,5-diketo-D-gluconate reductase A
spr1057	Disagree	TIGRFAM (TIGR02254)	YjjG/YfnB: HAD superfamily (subfamily IA)
spr1052	Agree	TIGRFAM (TIGR00797)	MATE Efflux
spr1805	Agree	TIGRFAM (TIGR00010)	hydrolase, TatD family
spr1839	Agree	CDD (cd00115)	Low molecular weight phosphatase family
YPO2631	Agree	Pfam (PF03573)	OprD, outer membrane porin, OprD family
YPO1104	Indeter	Protein Clusters (PRK11548)	hypothetical protein
YPO2155	Disagree	COG (COG3713)	OmpV, Outer membrane protein V
YPO0747	Indeter	Protein Clusters (PRK06778)	hypothetical protein
YPO2559	Indeter	Protein Clusters (PRK03826)	hypothetical protein.
DP0843	Agree	PIRSF (PIRSF000138)	alpha-hydroxy acid dehydrogenase, FMN-dependent
DP2277	Agree	Pfam (PF02016)	Peptidase_S66, LD-carboxypeptidase
DP2637	Agree	Protein Clusters (PRK00454)	GTPase EngB
DP2904	Agree	PIRSF (PIRSF004486)	S-adenosyl-L-methionine dependent methyltransferase, MraW type
DP1439	Agree	Protein Clusters (PRK10860)	tRNA-specific adenosine deaminase.
DP1954	Agree	TIGRFAM (TIGR03162)	ribazole_cobC: alpha-ribazole phosphatase (3.1.3.73)
DP0196	Agree	Protein Clusters (PRK09575)	multidrug efflux pump VmrA
NMC2078	Agree	TIGRFAM (TIGR02727)	MTHFS_bact: 5,10-methenyltetrahydrofolate s
NMC1453	Disagree	TIGRFAM (TIGR00387)	glycolate oxidase, subunit GlcD
NMC0498	Agree	TIGRFAM (TIGR02011)	IscA: iron-sulfur cluster assembly protein
NMC0361	Disagree	Pfam (PF02525)	Flavodoxin_2, Flavodoxin-like fold
NMC1815	Agree	PIRSF (PIRSF006118)	deoxy-D-manno-octulosonate 8-phosphate phosphatase
NMC1077	Agree	Protein Clusters (PRK10348)	ribosome-associated heat shock protein Hsp15
NMC1442	Agree	TIGRFAM (TIGR00453)	ispD: 2-C-methyl-D-erythritol 4-phosphate c
NMC1576	Agree	TIGRFAM (TIGR00521)	coaBC_dfp: phosphopantothienoylcysteine deca

6.6 Discussion

The rationale behind the creation of the BioMiner system is that integrating data and handling uncertainty in data can improve prokaryotic protein annotation. Results of this formal evaluation address our research sub-question regarding the performance of our

uncertainty model versus existing methods (Chapter 1, section 1.2.3), and indicate three major findings in this regard:

- Integration of multiple protein databases can improve the quality of function predictions for hypothetical proteins.
- Data integration is not enough in isolation. A data integration system needs to have an incorporated uncertainty model in the data to rank results properly. Training the system, or learning its parameter values from data, appears to improve performance significantly.
- Independent, gold-standard reference annotation sets are vital for evaluating the performance of systems for annotating proteins.

Our results show that data integration improves protein annotation; the aggregated protein databases (TIGRFAM, CDD, InterPro) performed better than their individual components (Pfam, COG, etc.) in regards to their predictions agreeing with both gold-standard annotation sets, and that BioMiner, the trained version, performed the best overall. Note that SwissProt and UniProt results could be inflated since proteins in them are often annotated using other protein databases (e.g. InterPro). This is likely due to the fact that the combination of protein databases in the BioMiner federation is not found in any existing aggregated protein database, specifically COG, PRK, TIGRFAM, Pfam, and PIRSF. The annotation error rate for the trained version of BioMiner (Disagrees/Total) is 18.4% (7/38), which is higher than the 8% estimated by Brenner [46], but much less than a higher estimated by Devos and Valencia [47]. Further optimization of BioMiner could potentially reduce this error rate.

Data integration alone cannot improve performance. An additional challenge is selecting the correct answer from multiple results provided by the different sources. BioMiner is able to accomplish this via its novel uncertainty functionality. The key to improving the performance of BioMiner was to use the CACC (Conditional Accuracy) results for each of the protein databases as uncertainty parameter settings. This takes into account the fact that the performance of protein databases can vary considerably in regard to their ability to annotate hypothetical proteins. This “training” enabled BioMiner to provide the most accurate result rankings. There was the issue of possible over-fitting BioMiner to the initial training set of 30 proteins in *S. oneidensis*, but when BioMiner was tested on a new gold-standard set of 38 proteins in five different bacterium, it still significantly outperformed all other protein databases. This suggests that, while BioMiner performs well using (possibly) imprecise probability estimates (Chapter 4 and Chapter 5), it also responds to training, which opens the door for further optimization of the system on larger gold-standard annotation sets.

Finally, the importance of independent gold-standard reference annotation sets cannot be understated. Using our gold-standard annotation sets we were able to provide some of the first performance metrics of the accuracy of protein databases, and BioMiner, in regards to their ability to annotate hypothetical proteins, an important and difficult biological problem. These performance metrics can inform the further refinement of protein databases and potentially improve their annotation accuracy. The approach we developed in this study to create gold-standard annotation reference sets therefore represents a significant contribution, although it is certainly not the last word

on the matter. Future work could be focused on augmenting and refining methods to create gold-standard annotation sets for evaluating protein function predictions, a critical aspect in regards to improving protein annotation.

Chapter 7: SUMMARY AND CONCLUSIONS

7.1 Research Contributions

The results of this dissertation address the primary research question introduced in Chapter 1: “How well does computational modeling of uncertainty in the annotation process improve systems for computational annotation of proteins?” Its main contributions are:

- The development and implementation of a novel uncertainty model for computational protein annotation, BioMiner, which is described in Chapter 4 and validated in Chapter 5. BioMiner builds on the BioMediator and UII systems as described in Chapter 3, section 3.3, Chapter 3, section 3.4, and Chapter 4, section 4.1. BioMiner, through its unique combination of annotation data sources and incorporated uncertainty model, enables more accurate annotation of proteins.
- A demonstration of the robustness of our uncertainty model through a principled methodology for analyzing and evaluating the choice of parameter values through multiple sensitivity analyses (Chapter 5). We also describe a general scenario where modeling uncertainty adds value over common alternative, but

non-probabilistic, approaches which we illustrate using a specific example (Chapter 5, section 5.6).

- A rigorous evaluation of the BioMiner system which demonstrates that annotations produced by BioMiner are more accurate than existing computational annotation systems. An additional contribution of this evaluation was the creation of two “gold-standard” annotation sets and the development of a method to create them, which is extremely important for true performance assessment.
- Results from the BioMiner evaluation represents the first performance benchmarks of commonly utilized computational annotation systems (such as InterPro) in a real-world application scenario: assigning function to hypothetical proteins (Chapter 6).

7.1.1 Key Optimizations of the BioMiner system

The default probabilistic values in the BioMiner system treated all sources equally ($P_s = 1.0$ for all sources) as, prior to this study, it was unknown how different protein databases perform in regards to annotating hypothetical proteins. As it turned out, this was a very important component in regards to the performance BioMiner (Chapter 6). By utilizing the conditional accuracy values for incorporated protein databases in the uncertainty model, BioMiner was able rank results based on a continuum which takes into account the quality of the result from a source as well as the overall quality of the source itself. A simple example would be if two sources produced results of equivalent

quality (i.e. same e-value), the result from the more accurate result is ranked higher. This supports the argument for a probabilistic system as this sort of “logic” would be difficult to encode in, say, a rule-based system. This critical use of data source accuracy enabled BioMiner to provide the best overall performance. Moreover, the correct annotations produced by BioMiner originated from five different sources, providing strong evidence for the value of data integration. This does suggest that probability values in BioMiner benefit from training data, however in this study only a single metric was determined in this way (the others were determined by experts).

7.1.2 Robustness of the uncertainty model in BioMiner

The probability values used to populate the uncertainty metrics in the BioMiner system can be seen as “rough” guesses determined in consultation with biological domain experts (i.e. the default metrics). Estimating probabilities from training data can be onerous and if rough guesses can be shown to be accurate enough in most cases, the effort needed to obtain probabilistic values can be greatly reduced. Prototyping and evaluation of the system can then proceed at a faster rate. We demonstrate that our default metrics are indeed accurate via a methodological sensitivity analysis (Chapter 5). Results from the sensitivity analysis study indicated that moderate variations to our default probabilities do not greatly affect the rankings produced by BioMiner (which are indeed fairly accurate). Learning “true” probabilities from training data is likely to only improve the performance of BioMiner when they cannot be estimated by domain experts. The primary example of this is with the “Ps” metric in our uncertainty model which indicates the level of user “confidence” in a particular source. Ps metrics for all

sources were initially identical ($P_s=1.0$), but after subsequent evaluation it was determined that some sources are more accurate than others. When P_s metrics were changed to reflect this, the performance of BioMiner improved.

The method for performing the sensitivity analysis is adapted and modified from the artificial intelligence domain (e.g. Bayesian Networks), and its application to data integration is also a contribution of this dissertation.

7.1.3 Gold-standard annotations datasets for evaluation

An additional contribution of this dissertation is the development of two “gold-standard” annotation datasets for evaluation purposes (Chapter 6). We also describe our approach for creating these datasets which has general applicability (Appendix B).

While not a true gold-standard, as no “wet-lab” experimentation is involved, it is highly reliable, as well as independent, and seeks to alleviate most common errors in computational annotation. It is a manual method however and thus should not be considered in the same class as the automated (i.e. computational) annotation methods evaluated in this dissertation.

A reliable, and independent, gold-standard annotation datasets are extremely important for evaluation purposes. The main reason for this is related to data provenance. Most existing annotations are creating using computational methods. Ideally, it should be possible to trace these annotations back to their original experimental source. Unfortunately, the evidence for an annotation is often not recorded and this evidence trail (i.e. provenance) is broken. The result is that computational annotations are often based not on experimental evidence but on other

computational annotations which are subsequently based on computational annotations and so on (possibly in circular fashion). We should mention here that there are known cases where incorrect annotations have been propagated throughout protein databases. This makes performance evaluation of computational methods difficult even when results are inspected by domain experts. Our approach breaks this cycle by that ensuring that annotations are based on direct experimental evidence of function. This subsequently provides a more trustworthy measure of performance for computational protein annotation methods.

7.2 Limitations

There are several caveats to our sensitivity analysis study. The topology analysis indicates that very accurate rankings can be achieved by “deterministic” (i.e. non-probabilistic) methods such as simply by considering the number of in-links to Gene Ontology (GO) terms (see Figure 7.1 and Chapter 5). In fact, it performs about as well as ranking by relevance score. This begs the question: “Do the probabilities (and the uncertainty model) matter at all?” This is a legitimate concern which we address by describing a general scenario where an uncertainty model provides advantages over deterministic approaches (Chapter 5, section 5.6). Our scenario is even more apparent in our evaluation regarding hypothetical proteins in Chapter 6, section 6.5. Correct annotations produced by BioMiner originated from five different sources and none of the sources are definitive. This may however diminish our sensitivity analysis story somewhat as robustness, generally speaking, depends on the number of paths to correct

the number of paths to correct results being higher than to incorrect ones. If a plausible result is found via only a single path, its ranking will likely be affected by variations in its probabilistic values to a greater degree. Future sensitivity analysis on data integration systems in regards to annotation would well served by evaluating robustness on hypothetical proteins (or less-well characterized genes), where annotation is less “disseminated”. Note that this further strengthens our rationale for developing a reliable and independent (but manual) gold-standard method for annotating hypothetical proteins.

Also, the final evaluation of BioMiner was performed on a small set of bacteria which is likely not be representative of all organisms (eukaryotes for example), or even prokaryotes. The evaluation also compared BioMiner versus available computational methods (protein databases). While these protein databases are in common use, they generally do not utilize recent advances in data integration (although they are integrated databases). Thus it is not a direct “head-to-head” competition of data integration methodologies but rather an evaluation of whether or not data integration can be applied to protein annotation and improve upon existing approaches. Additionally, the sparse or non-use of GO terms by several protein databases required that the evaluations be performed manually. While steps were taken to ensure consistency, idiosyncrasies in function nomenclature sometimes made evaluation difficult. There could be some variation in assessments by different biological domain experts for some proteins for instance.

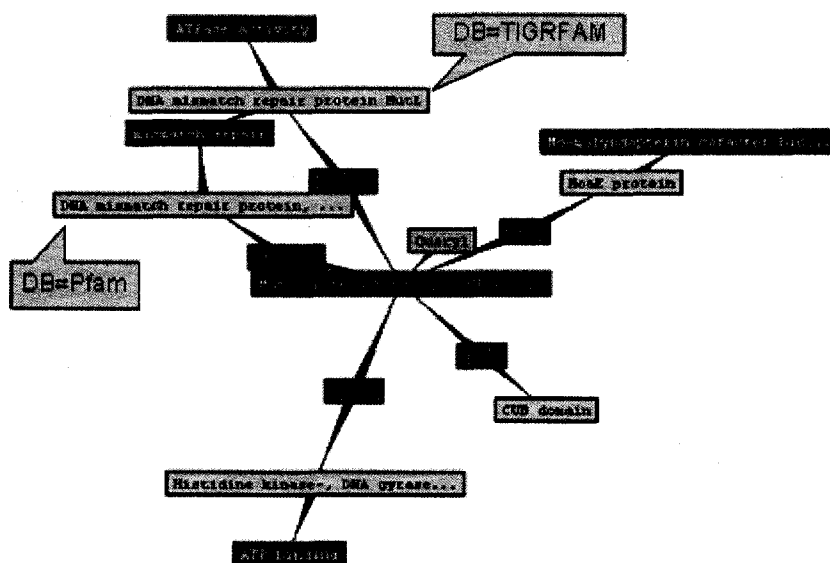


Figure 7.1: A result “graph” from the BioMiner system. The green nodes are GO terms, which represent functions predicted by the system. The “Query1” node represent the initial starting query, a gene of unknown function for example. In this case the GO function “mismatch repair” is pointed to by results from both Pfam and TIGRFAM, as opposed to “ATP binding” and “Mo-molybdopterin cofactor biosynthesis” which are only pointed to by one source each. For the 20 genes used as a gold-standard in the sensitivity analysis study, very good rankings of predicted functions can be achieved simply by considering the number of paths to a GO term.

Databases in the biomedical domain are not independent. Information, such as annotations, percolates freely between them. However, this information percolation is imperfect, therefore data integration methods can still be beneficial, only not in all cases. If nothing is known about a particular gene for instance, then data integration provides no benefit in that the needed information doesn't exist. If information about a particular gene is widely disseminated (i.e. its function is well-known), then data integration provides little benefit in that most sources can provide the necessary information. Somewhere in between is where data integration helps most. This is not a caution against deployment of data integration systems in the biomedical domain as

how “well-known” the functions are for a particular gene is generally not known a-priori. The assumption should be that no single database is definitive and data integration can provide at least a marginal benefit.

Finally, data integration may have its limits. For instance, the BioMiner system, in its current incarnation, only integrates protein sequence databases. There are other types of databases which can be utilized for annotation, such as expression or interaction databases. These could have the effect of improving annotations produced by the system or increasing “noise” in the result sets, as illustrated by inclusion of the PSI-BLAST database in BioMiner. This limit is still to be determined however.

7.3 Summary of Research

In order to realize the promise of genome sequencing efforts, improvements must be made in regard to annotating hypothetical proteins. Given that direct “wet-lab” experimentation on the enormous population of hypothetical proteins is expensive as well as infeasible, computational methods which predict protein function are necessary. There are many available choices in this regard and are available on-line as protein databases. No one single protein database is comprehensive in regard to annotation however, so multiple searches are often necessary but can be onerous on the user. Also, users may not search enough databases to obtain the best possible answer. This is where data integration methodologies can be employed for improvement of protein annotation. Formal data integration methods can enable broader and more consistent annotation searches, increasing the possibility that the best annotation for a given

protein is obtained. These methods, of course, are not without problems of their own. A primary challenge is handling uncertainty in data (particularly in the biomedical domain). Most data integration systems do not handle data uncertainty well which can cause result set “explosion”, making it difficult for users to find answers to queries and decreasing the utility of these systems. In this dissertation however, we show that a lightweight data integration system, BioMiner, with an incorporated and formal uncertainty model of biomedical data can be utilized for protein annotation. Moreover, we show that the determination of most probabilistic values for the uncertainty model may not require intensive machine learning approaches or large data sets but can be simply and quickly estimated by domain experts. Finally, through a rigorous evaluation facilitated by two gold-standard annotation datasets which we developed, we show that BioMiner outperforms existing computational approaches for annotating hypothetical proteins, which expands knowledge in the biological domain.

Bibliography

- [1] D. Benson, I. Karsch-Mizrachi, D. Lipman, J. Ostell, and D. Wheeler, "GenBank," *Nucleic Acids Research*, vol. 35, pp. D21-5, 2007.
- [2] J. C. Venter, D. Rusch, A. Halpern, and G. Sutton, "The Sorcerer II Global Ocean Sampling Expedition: Northwest Atlantic through Eastern Tropical Pacific," *PLoS Biology*, vol. 5, pp. 398-431, 2007.
- [3] E. Koonin and M. Galperin, *Sequence - Evolution - Function*. Boston: Kluwer Academic Publishers, 2003.
- [4] A. Valencia, "Automatic annotation of protein function," *Current Opinion in Structural Biology*, vol. 15, pp. 267-274, 2005.
- [5] E. Kolker, A. Piccone, M. Galperin, R. Smith, C. Giometti, K. Nealson, J. Fredrickson, and J. Tiedje, "Global profiling of *Shewanella oneidensis* MR-1: Expression of hypothetical genes and improved functional annotations.," *PNAS*, vol. 102, pp. 2099-2104, 2005.
- [6] P. Bork, "Powers and Pitfalls in Sequence Analysis: The 70% Hurdle," *Genome Research*, vol. 10, pp. 398-400, 2000.
- [7] M. Y. Galperin and E. V. Koonin, "'Conserved hypothetical' proteins: prioritization of targets for experimental study," *Nucleic Acids Research*, vol. 32, pp. 5452-5463, 2004.
- [8] R. Roberts, "Identifying Protein Function - A Call for Community Action," *PLoS Biology*, vol. 2, pp. e42, 2004.
- [9] M. Black and J. Hodgson, "Novel target sites in bacteria for overcoming antibiotic resistance," *Advanced Drug Delivery Reviews*, vol. 57, pp. 1528-1538, 2005.
- [10] A. Westwell, "Advances in molecular targets in cancer therapeutics," *Drug Discovery Today*, vol. 9, pp. 207-209, 2004.

- [11] T. Ideker, V. Thorsson, J. Ranish, R. Christmas, J. Buhler, J. Eng, R. Bumgarner, D. Goodlett, R. Aebersold, and L. Hood, "Integrated Genomics and Proteomic Analysis of a Systematically Perturbed Metabolic Network," *Science*, vol. 292, pp. 929-934, 2001.
- [12] E. Cadag, B. Louie, P. Myler, and P. Tarczy-Hornoch, "Biomediator Data Integration and Inference for Functional Annotation of Anonymous Sequences," presented at Pacific Symposium on Biocomputing, 2007.
- [13] B. Altenberg and K. Greulich, "Genes of glycolysis are ubiquitously overexpressed in 24 cancer classes," *Genomics*, vol. 84, pp. 1014-20, 2004.
- [14] J. C. Venter, K. Remington, J. Heidelberg, A. Halpern, Y.-H. Rogers, and O. Smith, "Environmental Genome Shotgun Sequencing of the Sargasso Sea," *Science*, vol. 304, pp. 66-74, 2004.
- [15] A. Bernal, U. Ear, and N. Kyrpides, "Genomes OnLine Database (GOLD): a monitor of genome projects world-wide," *Nucleic Acids Research*, vol. 29, pp. 126-7, 2001.
- [16] P. Karp, S. Paley, and J. Zhu, "Database verification of studies of SWISS-PROT and GenBank," *Bioinformatics*, vol. 17, pp. 526-532, 2001.
- [17] E. Worthey and P. Myler, "Protozoan genomes: gene identification and annotation," *International Journal for Parasitology*, vol. 35, pp. 495-512, 2005.
- [18] M. Y. Galperin, "The Molecular Biology Database Collection: 2006 update.," *Nucleic Acids Research*, vol. 34, pp. D3-D5, 2006.
- [19] U. Consortium, "The Universal Protein Resource (UniProt)." *Nucleic Acids Research*, vol. 35, pp. D193-7, 2007.
- [20] E. Sonnhammer, S. Eddy, and R. Durbin, "Pfam: a comprehensive database of protein domain families based on seed alignments," *Proteins*, vol. 28, pp. 405-20, 1997.
- [21] D. H. Haft, J. D. Selengut, and O. White, "The TIGRFAMs database of protein families.," *Nucleic Acids Research*, vol. 31, pp. 371-373, 2003.
- [22] I. Xenarios, L. Salwinski, X. Duan, P. Higney, S. Kim, and D. Eisenberg, "DIP, the Database of Interacting Proteins: a research tool for studying cellular

- networks of protein interactions," *Nucleic Acids Research*, vol. 30, pp. 303-305, 2002.
- [23] C. von Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel, "STRING: a database of predicted functional associations between proteins," *Nucleic Acids Research*, vol. 31, pp. 258-261, 2003.
- [24] M. Ringwald, J. Eppig, D. Begley, J. Corradi, I. McCright, T. Hayamizu, D. Hill, J. Kadin, and J. Richardson, "The Mouse Gene Expression Database (GXD)," *Nucleic Acids Research*, vol. 29, pp. 98-101, 2001.
- [25] R. Edgar, M. Domrachev, and A. Lash, "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30, pp. 207-210, 2002.
- [26] D. Roos, "Bioinformatics--Trying to Swim in a Sea of Data," *Science*, vol. 291, pp. 1260-1261, 2001.
- [27] L. Stein, "Integrating Biological Databases," *Nature Reviews: Genetics*, vol. 4, pp. 337-45, 2003.
- [28] A. Levy, *Logic-based techniques in data integration*. Norwell, MA, USA: Kluwer Academic Publishers, 2000.
- [29] A. Barioch and R. Apweiler, "The Swiss-Prot protein sequence data bank and its new supplement TREMBL," *Nucleic Acids Research*, vol. 24, pp. 21-25, 1996.
- [30] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped Blast and Psi-Blast: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, pp. 3389-3402, 1997.
- [31] J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia, "Sequence Comparisons Using Multiple Sequences Detect Three Times as Many Remote Homologues as Pairwise Methods.," *Journal of Molecular Biology*, vol. 284, pp. 1201-1210, 1998.
- [32] A. Marchler-Bauer, A. Panchenko, B. Shoemaker, P. Thiessen, L. Geer, and S. Bryant, "CDD: a database of conserved domain alignments with links to domain three-dimensional structure," *Nucleic Acids Research*, vol. 30, pp. 281-283, 2002.

- [33] L. Stein, "Genome annotation: from sequence to biology.," *Nature Reviews: Genetics*, vol. 2, pp. 493-503, 2001.
- [34] T. K. Attwood, "The quest to deduce protein function from sequence: the role of pattern databases," *The International Journal of Biochemistry & Cell Biology*, vol. 32, pp. 139-155, 2000.
- [35] K. Kasukawa, M. Furuno, I. Nikaido, H. Bono, D. Hume, C. Bult, D. Hill, R. Baldarelli, J. Gough, A. Kanapin, H. Matsuda, L. Schriml, Y. Hayashizaki, Y. Okazaki, and J. Quackenbush, "Development and evaluation of an automated annotation pipeline and cDNA annotation system.," *Genome Research*, vol. 13, pp. 1542-51, 2003.
- [36] V. Curwen, E. Eyraas, T. Andrews, L. Clarke, E. Mongin, S. Searle, and M. Clamp, "The Ensembl automatic gene annotation system," *Genome Research*, vol. 14, pp. 942-50, 2004.
- [37] M. Andrade, N. Brown, A. Valencia, C. Ouzounis, and C. Sander, "Automated genome sequence analysis and annotation," *Bioinformatics*, vol. 15, pp. 391-412, 1999.
- [38] E. Quevillon, V. Silventoinen, S. Pillai, N. Harte, N. Mulder, R. Apweiler, and R. Lopez, "InterProScan: protein domains identifier," *Nucleic Acids Research*, vol. 33, pp. W116-W120, 2005.
- [39] B. Louie, T. Detwiler, N. Dalvi, R. Shaker, P. Tarczy-Hornoch, and D. Suci, "Incorporating Uncertainty Metrics into a General-Purpose Data Integration System," presented at 19th International Conference on Scientific and Statistical Database Management (SSDBM), Banff, Canada, 2007.
- [40] C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg, "Two Methods for Assessment of the Reliability of High Throughput Observations," *Molecular and Cellular Proteomics*, vol. 1, pp. 349-356, 2002.
- [41] K. Pruitt and D. Maglott, "RefSeq and LocusLink: NCBI gene-centered resources," *Nucleic Acids Research*, vol. 29, pp. 137-140, 2001.
- [42] G. O. Consortium, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium," *Nature Genetics*, vol. 25, pp. 25-9, 2000.

- [43] J. Ashurst and J. Collins, "Gene annotation: prediction and testing.," *Annu Rev Genomics Hum Genet.*, vol. 4, pp. 69-88, 2003.
- [44] C. Ouzounis and P. Karp, "The past, present and future of genome-wide re-annotation," *Genome Biology*, vol. 3, pp. 2001.1-2001.6, 2002.
- [45] I. Iliopoulos, S. Tsoka, M. Andrade, A. Enright, I. Rigoutsos, C. Sander, A. Valencia, and C. Ouzounis, "Evaluation of annotation strategies using an entire genome sequence," *Bioinformatics*, vol. 19, pp. 717-726, 2003.
- [46] S. Brenner, "Errors in genome annotation," *Trends In Genetics*, vol. 15, pp. 132-3, 1999.
- [47] D. Devos and A. Valencia, "Intrinsic errors in genome annotation," *Trends In Genetics*, vol. 17, pp. 429-31, 2001.
- [48] M. Y. Galperin and E. V. Koonin, "Sources of systematic error in functional annotation of genomes: domain rearrangement, non-orthologous gene displacement, and operon disruption," *In Silico Biology*, 1998.
- [49] W. Sujansky, "Heterogeneous Database Integration in Biomedicine," *Journal of Biomedical Informatics*, vol. 34, pp. 295-298, 2001.
- [50] K. A. Karasawas, R. Baldock, and A. Burger, "Bioinformatics integration and agent technology," *Journal of Biomedical Informatics*, vol. 37, pp. 205-219, 2004.
- [51] B. Louie, P. Mork, F. Martin-Sanchez, A. Y. Halevy, and P. Tarczy-Hornoch, "Data Integration and Genomic Medicine," *Journal of Biomedical Informatics*, vol. 40, pp. 5-16, 2007.
- [52] D. Karolchik, R. Baertsch, D. Haussler, and W. J. Kent, "The UCSC Genome Browser Database," *Nucleic Acids Research*, vol. 31, pp. 51-54, 2003.
- [53] H. Muller, "Problems, Methods and Challenges in Comprehensive Data Cleansing," Humboldt-Universitt zu Berlin, Institut fr Informatik, Berlin HUB-IB-164, 2003.
- [54] P. Thiran, J. Hainaut, and G. Houben, "Database Wrappers Development: Towards Automatic Generation," presented at Ninth European Conference on Software Maintenance and Reengineering (CSMR '05), 2005.

- [55] R. Shaker, P. Mork, J. Brockenbrough, L. Donelson, and P. Tarczy-Hornoch, "The BioMediator System as a Tool for Integrating Databases on the Web," presented at Proceedings of the Workshop on Information Integration on the Web, Toronto, ON, 2004.
- [56] P. Mork, A. Y. Halevy, and P. Tarczy-Hornoch, "A Model for Data Integration Systems of BioMedical Data Applied to Online Genetic Databases," presented at Proceedings of the American Medical Informatics Annual Fall Symposium, Washington, D.C., 2001.
- [57] P. Mork, "Peer Architectures for Knowledge Sharing," in *Computer Science and Engineering*, vol. Doctor of Philosophy. Seattle: University of Washington, 2005, pp. 229.
- [58] L. Wong, "Technologies for integrating biological data," *Briefings in Bioinformatics*, vol. 3, pp. 389-404, 2002.
- [59] T. Hernandez and S. Kambhampati, "Integration of biological sources: current systems and challenges ahead," *ACM Sigmod Record*, vol. 33, pp. 51-60, 2004.
- [60] S. Y. Chung and L. Wong, "Kleisli: a new tools for data integration in biology," *Trends in Biotechnology*, vol. 17, pp. 351-355, 1999.
- [61] P. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens, "TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources," *Bioinformatics*, vol. 16, pp. 184-5, 2000.
- [62] S. B. Davidson, V. Tannen, J. Crabtree, G. C. Overton, B. P. Brunk, J. Schug, and C. J. Stoeckert, "K2/Kleisli and GUS: Experiments in integrated access to genomic data," *IBM Systems Journal*, vol. 40, pp. 512-531, 2001.
- [63] D. Maglott, J. Ostell, K. Pruitt, and T. Tatusova, "Entrez Gene: gene-centered information at NCBI," *Nucleic Acids Research*, vol. 33, pp. D54-D58, 2005.
- [64] T. Etzold, A. Ulyanov, and P. Argos, "SRS: information retrieval for molecular biology data banks," *Methods Enzymology*, vol. 266, pp. 114-28, 1996.
- [65] W. E. Grosso, H. Eriksson, R. W. Fergerson, J. Gennari, W. Tu, and M. A. Musen, "Knowledge Modeling at the Millenium (The Design and Evolution of Protege-2000)," presented at Proceedings of the 12th International Workshop on

Knowledge Acquisition, Modeling and Management (KAW' 99), Banff, Canada, 1999.

- [66] A. Shapiro, "TouchGraph: open source software for graph visualization using spring-layout and focus+context techniques," 2006.
- [67] L. Donelson, P. Tarczy-Hornoch, P. Mork, C. Dolan, J. Mitchell, M. Barrier, and H. Mei, "The BioMediator System as a Data Integration Tool to Answer Diverse Biologic Queries," presented at Medinfo, 2003.
- [68] E. Cadag, "Rule-based Automated Gene Annotation Utilizing the BioMediator Data Integration Platform," in *Biomedical and Health Informatics*. Seattle: University of Washington, 2006.
- [69] J. Boulos, N. Dalvi, B. Mandhani, S. Mathur, C. Re, and D. Suci, "MYSTIQ: A system for finding more answers by using probabilities," presented at SIGMOD, Baltimore, Maryland, USA, 2005.
- [70] A. Birkland and G. Yona, "BIOZON: a system for unification, management and analysis of heterogeneous biological data," *BMC Bioinformatics*, vol. 7, 2006.
- [71] M. Jayapandian, A. Chapman, D. States, and H. V. Jagadish, "Michigan Molecular Interactions (MiMI): putting the jigsaw puzzle together," *Nucleic Acids Research*, vol. 35, pp. D566-D571, 2007.
- [72] R. Apweiler, T. Attwood, A. Bairoch, A. Bateman, and E. Birney, "The InterPro database, and integrated documentation resource for protein families, domains and functional sites," *Nucleic Acids Research*, vol. 29, pp. 37-40, 2001.
- [73] C. Colbourn, *The Combinatorics of Network Reliability*. New York, NY, USA: Oxford University Press, Inc., 1987.
- [74] D. Karger, "A Randomized Fully Polynomial Time Approximation Scheme for the All-Terminal Network Reliability Problem," *SIAM Review*, vol. 43, pp. 499-422, 2001.
- [75] D. Lee, O. Redfern, and C. Orengo, "Predicting protein function from sequence and structure," *Nature Reviews Molecular Cell Biology*, vol. 8, pp. 995-1005, 2007.

- [76] L. Stein, C. Mungall, S. ShengQiang, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. Stajich, T. Harris, A. Arva, and S. Lewis, "The Generic Genome Browser: A Building Block for a Model Organism System Database," *Genome Research*, vol. 12, pp. 1599-1610, 2002.
- [77] J. Gough, K. Karplus, R. Hughey, and C. Chothia, "Assignment of Homology to Genome Sequences using a Library of Hidden Markov Models that Represent all Proteins of Known Structure.," *J. Mol. Biol.*, vol. 4, pp. 903-919, 2001.
- [78] R. Overbeek, T. Begley, R. Butler, O. Zagnitko, and V. Vonstein, "The Subsystems Approach to Genome Annotation and its Use in the Project to Annotate 1000 Genomes.," *Nucleic Acids Research*, vol. 33, pp. 5691-5702, 2005.
- [79] S. Asthana, O. King, F. Gibbons, and F. Roth, "Predicting Protein Complex Membership Using Probabilistic Network Reliability," *Genome Research*, vol. 14, pp. 1170-1175, 2004.
- [80] O. Troyanskaya, K. Dolinski, A. Owen, R. Altman, and D. Botstein, "A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *Saccharomyces cerevisiae*).," *PNAS*, vol. 100, pp. 8348-8353, 2003.
- [81] J. Weston, A. Elisseff, D. Zhou, C. S. Leslie, and W. S. Noble, "Protein ranking: from local to global structure in the protein similarity network," *PNAS*, vol. 101, pp. 6559-6563, 2004.
- [82] R. Tatusov, M. Galperin, D. Natale, and E. V. Koonin, "The COG database: a tool for genome-scale analysis of protein functions and evolution," *Nucleic Acids Research*, vol. 28, pp. 33-36, 2000.
- [83] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, pp. 235-242, 2000.
- [84] O. Bodenreider, "GenNav: Visualizing Gene Ontology as a graph," presented at Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium, 2002.

- [85] C. Wu, H. Huang, L. Yeh, and W. Barker, "Protein family classification and functional annotation," *Computational Biology and Chemistry*, vol. 27, pp. 37-47, 2003.
- [86] C. Wu, H. Huang, A. Nikolskaya, Z. Hu, and W. Barker, "The iProClass integrated database for protein functional analysis," *Computation Biology and Chemistry*, vol. 28, pp. 87-96, 2004.
- [87] V. Coupe, L. Van Der Gaag, and J. Habbema, "Sensitivity analysis: an aid for belief-network quantification," *The Knowledge Engineering Review*, vol. 15, pp. 215-232, 2000.
- [88] A. Tversky and D. Kahneman, "Judgment under uncertainty: heuristics and biases," *Science*, vol. 185, pp. 1124-31, 1974.
- [89] D. Heckerman, "Bayesian Networks for Data Mining," *Data Mining and Knowledge Discovery*, vol. 1, pp. 79-119, 1997.
- [90] M. Druzdzel and L. Van Der Gaag, "Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information," presented at Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95), San Francisco, CA, 1995.
- [91] P. Karp, "What We Do Not Know About Sequence Analysis and Sequence Databases," *Bioinformatics*, vol. 14, pp. 753-54, 1998.
- [92] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*: Morgan Kaufmann, 1988.
- [93] D. Heckerman and B. Nathwani, "Towards normative expert systems: part II - probability based representations for efficient knowledge acquisitions and inference," *Methods of Information in Medicine*, vol. 31, pp. 106-116, 1992.
- [94] V. Coupe, N. Peek, J. Ottenkamp, and J. Habbema, "Using sensitivity analysis for efficient quantification of a belief network," *Artificial Intelligence in Medicine*, vol. 17, pp. 223-247, 1999.
- [95] M. Henrion, M. Pradhan, B. Favero, K. Huang, G. Provan, and P. O'Rourke, "Why is Diagnosis Using Belief Networks Insensitive to Imprecision in Probabilities?" presented at Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96), San Francisco, CA, 1996.

- [96] O. Kiersztok and H. Wang, "Another look at sensitivity of Bayesian Networks to imprecise probabilities," presented at Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics (AISTAT2001), 2001.
- [97] B. Abramson and K. Ng, "Toward an Art and Science of Knowledge Engineering: A Case for Belief Networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, pp. 705-712, 1993.
- [98] M. Druzdzal and L. Van Der Gaag, "Building Probabilistic Networks: 'Where Do the Numbers Come From?' Guest Editors' Introduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, pp. 481-486, 2000.
- [99] J. Habbema, P. Bossuyt, and D. Dippel, "Analysing Clinical Decision Analyses," *Statistics in Medicine*, vol. 9, pp. 1229-1242, 1990.
- [100] V. Coupe and L. van der Gaag, "Practicable sensitivity analysis of Bayesian belief networks," presented at Proceedings of the Joint Session of the 6th Prague Symposium of Asymptotic Statistics and the 13th Prague Conference on Information Theory, Statistical Decision functions and Random Processes, Prague, 1998.
- [101] K. Ng and B. Abramson, "A Sensitivity Analysis of Pathfinder: A Follow-up Study," presented at Proceedings of the 7th Annual Conference on Uncertainty in Artificial Intelligence (UAI-91), Los Angeles, CA, 1991.
- [102] C. Buckley and E. Voorhees, "Evaluating evaluation measure stability," presented at Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, Athens, Greece, 2000.
- [103] R. Baeza-Yates and B. Riberio-Neto, *Modern Information Retrieval*. Boston, MA: Addison-Wesley Longman Publishing, 1999.
- [104] R. Pagon, P. Tarczy-Hornoch, M. Covington, P. Baskin, J. Edwards, M. Espeseth, C. Beahler, T. Bird, B. Popovich, C. Nesbitt, C. Dolan, K. Marymee, N. Hanson, W. Neufeld-Kaiser, G. McCullough Grohs, T. Kicklighter, C. Abair, A. Malmin, M. Barclay, and R. Palepu, "GeneTests and GeneClinics: Genetic Testing Information for a Growing Audience," *Hum Mutat*, vol. 19, pp. 501-509, 2002.

- [105] C. Wu, H. Huang, L. Arminski, C. Castro-Alvear, J. Zhang, and W. Barker, "The Protein Information Resource: an integrated public resource of functional annotation of proteins," *Nucleic Acids Research*, vol. 30, pp. 35-37, 2002.
- [106] R. Tatusov, E. V. Koonin, and D. Lipman, "A genomic perspective on protein families," *Science*, vol. 278, pp. 637-7, 1997.
- [107] L. Koski and B. Golding, "The Closest BLAST Hit Is Often Not the Nearest Neighbor," *J Mol Evol*, vol. 52, pp. 540-2, 2001.
- [108] T. Smith and X. Zhang, "The challenges of genome sequence annotation of "The devil is in the details", " *Nature Biotechnology*, vol. 15, pp. 1222-1223, 1997.
- [109] J. Garrels, "Yeast genomic databases and the challenge of the post-genomic era," *Funct Integr Genomics*, vol. 2, pp. 212-237, 2002.
- [110] E. Kolker, K. Makarova, S. Shabalina, A. Picone, S. Purvine, T. Holzman, T. Cherny, D. Armbruster, R. Munson, G. Kolesov, D. Frishman, and M. Galperin, "Identification and functional analysis of 'hypothetical' genes expressed in *Haemophilus influenzae*," *Nucleic Acids Research*, vol. 32, pp. 2353-2361, 2004.
- [111] S. Henikoff, J. Henikoff, and S. Pietrokovski, "Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations," *Bioinformatics*, vol. 15, pp. 471-479, 1999.
- [112] K. O'Neill, W. Klimke, and T. Tatusova, "Protein Clusters: A Collection of Proteins Grouped by Sequence Similarity and Function," NCBI, 2007.
- [113] J. Schultz, F. Milpetz, P. Bork, and C. P. P. & Ponting, "SMART, a simple modular architecture research tool: Identification of signaling domains," *PNAS*, vol. 95, pp. 5857-5864, 1998.
- [114] I. Schomburg, A. Chang, and D. Schomburg, "BRENDA, enzyme data and metabolic information," *Nucleic Acids Research*, vol. 30, pp. 47-9, 2002.
- [115] S. Brenner, C. Chothia, and T. Hubbard, "Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships," *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 6073-6078, 1998.

Appendix A: Annotation Datasets

The following tables list the proteins used as a reference standard in Chapter 6. The “section” column indicates the evaluation in which a particular protein was used.

Proteins used in sections 6.5.5 and 6.5.6 (Tables A.1 & A.2) were created using gold-standard methods #1 and #2 respectively which are described in Appendix B.

The average number of proteins per organism annotated by gold-standard method #2 is between 21 and 43 hypothetical proteins (95% confidence interval, 616 bacterial genomes). All the organisms in the gold-standard annotation set for the evaluation in 6.5.6 had an average number of proteins annotated by gold-standard method #2 (*S. oneidensis*, *S. pneumoniae*, *Y. pestis*, *D. psychrophila*, and *N. meningitis*) (Table A.3).

Table A.1: Annotation dataset used in Chapter 6, section 6.3.3. These proteins were not annotated using our method described in Appendix B.

Locus	Section	Gold-standard annotation	PubMed Ids
SO_0332	6.3.3	Homoserine kinase, type II	Na
SO_0342	6.3.3	PrpF protein required for repair/synthesis of Fe-S center of AcnD	Na
SO_0506	6.3.3	3-octaprenyl-4-hydroxybenzoate decarboxylase UbiD	Na
SO_0887	6.3.3	Peptidylarginine deiminase	Na
SO_1523	6.3.3	NAD kinase	Na
SO_1597	6.3.3	Omega-3 polyunsaturated fatty acid synthase PfaD subunit/2-Nitropropane dioxygenase	Na
SO_1789	6.3.3	UDP-2,3-diacylglucosamine hydrolase	Na
SO_1963	6.3.3	Homogenetisate 1,2-dioxygenase	Na
SO_2593	6.3.3	NAD-specific glutamate dehydrogenase	Na
SO_2614	6.3.3	Aminodeoxychorismate lyase	Na
SO_2627	6.3.3	ATP-dependant Clp protease adaptor protein ClpS	Na
SO_3340	6.3.3	Mechanosensitive ion channel protein MscS	Na
SO_3436	6.3.3	tRNA pseudouridine synthase TruD	Na
SO_4413	6.3.3	Kynureninase	Na
SO_4680	6.3.3	CDP-glycerol:poly(glycerophosphate) glycerophosphotransferase	Na

SO_4719	6.3.3	Periplasmic tungstate-binding protein TupA, component of an ABC-type transporter	Na
SO_0265	6.3.3	Cytochrome c-type biogenesis factor CycH	Na
SO_0337	6.3.3	Endoribonuclease L-PSP	Na
SO_0363	6.3.3	Nucleoside-diphosphate-sugar pyrophosphorylase	Na
SO_0455	6.3.3	TRAP-type dicarboxylate transporter, permease component with fused DctQM subunit	Na
SO_0471	6.3.3	Flavin-dependant dioxygenase	Na
SO_0783	6.3.3	Superfamily I DNA and RNA helicase	Na
SO_1007	6.3.3	Na ⁺ /H ⁺ antiporter NhaC	Na
SO_1267	6.3.3	Glutamine synthetase-associated glutamina amidotransferase	Na
SO_1742	6.3.3	3-oxoacyl-acyl-carrier-protein	Na
SO_1981	6.3.3	Nicotinic acid phosphoribosyltransferase	Na
SO_3051	6.3.3	Mo-dependant oxoreductase maturation factor	Na
SO_3542	6.3.3	Phosphoketolase	Na
SO_3667	6.3.3	Heme iron utilization protein HugZ	Na
SO_3668	6.3.3	Heme iron utilization protein HugX	Na
SO_4227	6.3.3	S-adenosylmethionine-dependant methyltransferase MraW, involved in cell division	Na
SO_4690	6.3.3	Undecaprenyl phosphate-sugar: lipid A glycosyltransferase	Na
SO_0077	6.3.3	Thioesterase	Na
SO_0080	6.3.3	Thioesterase	Na
SO_0110	6.3.3	Metalloprotease, M48 family	Na
SO_0152	6.3.3	Carboxypeptidase	Na
SO_0301	6.3.3	Methyltransferase	Na
SO_0304	6.3.3	Endonuclease, distantly related to archaeal Holliday junction resolvase and Mrr-like restriction enzymes	Na
SO_0311	6.3.3	Fe-S oxireductase	Na
SO_0316	6.3.3	Phospholipid-binding protein, PEBP family	Na
SO_0428	6.3.3	Esterase, alpha-beta hydrolase superfamily	Na

Table A.2: Annotation dataset used in Chapter 6, section 6.5.5. These proteins were annotated using our method described in Appendix B. All proteins come from a single organism.

SO_0342	6.5.5	prpF protein required for repair/synthesis of Fe-S center of AcnD	14702315
SO_0506	6.5.5	3-octaprenyl-4-hydroxybenzoate decarboxylase UbiD	11029449,782527, 12799002
SO_0887	6.5.5	Peptidylarginine deiminase	10377098
SO_1313	6.5.5	Anhydro-N-acetylmuramic acid kinase	15901686, 16452451
SO_1523	6.5.5	NAD kinase	11488932
SO_1597	6.5.5	omega-3 polyunsaturated fatty acid synthase PfaD subunit	12055309
SO_1608	6.5.5	7-cyano-7-deazaguanine reductase	15767583
SO_1789	6.5.5	UDP-2,3-diacylglucosamine hydrolase	12000770, 12000771
SO_1851	6.5.5	Methyltransferase	17010378
SO_1963	6.5.5	homogenetisate 1,2-dioxygenase	10876237
SO_2042	6.5.5	sulfite oxidase subunit YedY	15355966, 16042411
SO_2043	6.5.5	sulfite oxidase subunit yedZ	15355966, 16042411
SO_2593	6.5.5	NAD-specific glutamate dehydrogenase	10924516
SO_2603	6.5.5	Methionine-S-sulfoxide reductase	17535911
SO_2614	6.5.5	aminodeoxychorismate lyase	11011151
SO_2627	6.5.5	ATP-dependant Clp protease adaptor protein ClpS	11931773, 12426582
SO_3014	6.5.5	chromosome segregation and condensation protein B	12100548
SO_3015	6.5.5	chromosome segregation and condensation protein A	12100548
SO_3367	6.5.5	tRNA guanine-N(7)-methyltransferase	12730187
SO_3436	6.5.5	tRNA pseudouridine synthase TruD	12756329
SO_3542	6.5.5	Phosphoketolase	16086247
SO_3578	6.5.5	multicopper polyphenol oxidase (laccase)	16740638
SO_3667	6.5.5	Heme iron utilization protein HugZ	16376031
SO_3668	6.5.5	Heme iron utilization protein HugX	16376031
SO_3957	6.5.5	3-deoxy-D-manno-octulosonate 8-phosphate phosphatase	12639950
SO_4227	6.5.5	S-adenosylmethionine-dependant methyltransferase MraW	10572301
SO_4398	6.5.5	D-tyrosyl-tRNA deacylase	10383414
SO_4413	6.5.5	Kynureninase	9264543, 9477966
SO_4677	6.5.5	3-deoxy-D-manno-octulosonic-acid kinase	10531340, 10952982
SO_4680	6.5.5	CDP-glycerol:poly(glycerophosphate) glycerophosphotransferase	10648531

Table A.3: Annotation dataset used in Chapter 6, section 6.5.6. All proteins were annotated using our method described in Appendix B. Proteins in this dataset originate from five organisms.

SO_0025	6.4.6	Protoporphyrin oxidase	7916647, 3611052
SO_0599	6.4.6	ATPase with strong ADP affinity	15324301
SO_0706	6.4.6	Regulator with hipB	15576765, 17041039
SO_0828	6.4.6	16S rRNA m2G1207 methylase	9873033, 17576679
SO_0887	6.4.6	Agmatine deiminase	12782327
SO_1267	6.4.6	gamma-Glu-GABA hydrolase	15590624, 16499623
SO_1431	6.4.6	twin-argininine leader-binding protein for DmsA and TorA	11309116
SO_2484	6.4.6	Deoxyribonucleoside 5'-monophosphatase	15489502
SO_3967	6.4.6	molybdate transporter subunit	8576221
SO_4537.2	6.5.6	Mitochondrial-processing peptidase subunit beta	8643535, 16554755, 16429126, 2905264, 3044780
SO_0946	6.5.6	Resistance-Nodulation-Cell Division (RND) multidrug efflux membrane fusion protein MexA precursor	15387820, 15722391, 17586626
spr0592	6.5.6	Hydroxyprostaglandin dehydrogenase 15 (NAD)	9099857
spr1622	6.5.6	M protein trans-acting positive regulator	15547255, 16513733
spr1332	6.5.6	2,5-diketo-D-gluconate reductase A	11934293, 16284956
spr1057	6.5.6	dUMP phosphatase	15489502, 17286574, 17189366
spr1052	6.5.6	multidrug efflux protein	15716425, 16954325, 11073914, 9661020
spr1805	6.5.6	DNase, magnesium-dependent	10747959
spr1839	6.5.6	Protein-tyrosine-phosphatase	17008719
YPO2631	6.5.6	Basic amino acid, basic peptide and imipenem outer membrane porin OprD precursor	2118530, 8843159, 16476803
YPO1104	6.5.6	small membrane lipoprotein	17404237
YPO2155	6.5.6	scaffolding protein for murein synthesizing machinery	10037771, 16154998
YPO0747	6.5.6	Protein that enables flagellar motor rotation	16971952
YPO2559	6.5.6	deoxyribonucleoside 5'-monophosphatase,	15489502
DP0843	6.5.6	hydroxyacid oxidase 2 (long chain) (1.1.3.15)	8508789
DP2277	6.5.6	LD-carboxypeptidase	16162494
DP2637	6.5.6	GTPase EngB	16997968
DP2904	6.5.6	S-adenosyl-dependent methyltransferase activity on membrane-located substrates	10572301
DP1439	6.5.6	tRNA-specific adenosine deaminase (3.5.4.4)	16142903, 16700551
DP1954	6.5.6	alpha ribazole-5'-P phosphatase	17209023
DP0196	6.5.6	multidrug efflux pump VmrA	11751837
NMC2078	6.5.6	5, 10-methenyltetrahydrofolate synthetase (3.5.4.9)	17055997

Table A.3 Continued.			
NMC1453	6.5.6	D-lactate dehydrogenase (1.1.1.28)	10509019,11805837
NMC0498	6.5.6	FeS cluster assembly protein	11319236,17244611,15985427
NMC0361	6.5.6	NADPH quinone reductase (1.6.99.6)	16630630,8611590
NMC1815	6.5.6	3-deoxy-D-manno-octulosonate 8-phosphate phosphatase (3.1.3.45)	12639950
NMC1077	6.5.6	ribosome-associated heat shock protein Hsp15	9867837
NMC1442	6.5.6	4-diphosphocytidyl-2C-methyl-D-erythritol synthase	16478479,10518523
NMC1576	6.5.6	fused 4'-phosphopantothenoylcysteine decarboxylase/phosphopantothenoylcysteine synthetase, FMN-binding,	10922366

Appendix B: Annotation Method

The following steps describe the method developed in this dissertation for creating gold-standard annotations in prokaryotic proteins. Reference annotation sets, created using this method were used for performance evaluation purposes in Chapter 6. There are two related methods described here. The first is the initial method we developed and was more useful when annotating proteins one-at-a-time in a single organism. The second method is somewhat more amenable to automation and was used to annotate proteins in more than one organism. Annotations created using either method must reference at least one publication describing direct “wet-lab” experimentation to determine function in the protein similar to the hypothetical protein. The steps for creating gold-standard annotations using method #1 are:

- 1) Compare an experimentally characterized protein versus the full complement of proteins in an organism of interest using BLAST or PSI-BLAST with low-complexity filtering ON. The top-hit must be a hypothetical protein with an *e*-value no greater than $e-04$. E-values at this threshold are considered to be indicative of homology [115].
- 2) The length of the shorter protein should be at least 90% of the length of the longer protein, or they must both contain at least 90% of a commonly shared conserved domain. For example, in our gold-standard set SO_0887 and NP_905579 have greater than 10% length difference but both contain greater than 99% of pfam04371, a 329 base-pair peptidyl-arginine deiminase domain (identified using the Conserved Domain Database).
- 3) If the above criteria are met then the function described in the publication is assigned to the hypothetical protein (from step #1).

It can tedious to manually determine whether or not both proteins share at least 90% of a commonly shared conserved domain so we modified our approach to address this.

The main difference between method #1 and method #2 is that it is easier to programmatically determine similarity in the second method.

- 1) Compare an experimentally characterized protein versus the full complement of proteins in an organism of interest using BLAST or PSI-BLAST with low-complexity filtering ON. The top-hit must be a hypothetical protein with an *e*-value no greater than $e-04$. This step is the same as in method #1.
- 2) The length of the shorter protein should be at least 90% of the length of the longer protein.
- 3) The BLAST alignment length must be ≥ 100 residues. This is the average functional domain size.
- 4) The BLAST alignment must cover at least 90% of the experimentally characterized protein.
- 5) Any unaligned regions of the experimentally characterized protein must be < 40 residues, which is approximately the smallest size functional domain.

VITA

Brenton Louie received his B.A. degree in Biology at the University of California at Santa Barbara in 1993. His relevant experience in the field of BioMedical Informatics includes two years work on the Human Genome Project at the Stanford Human Genome Center and six years developing gene databases at Incyte Genomics. He also was a National Library of Medicine Informatics Predoctoral Fellow From 2004-2007 at the University of Washington in Seattle, and a research associate for one year on the NSF-funded Uncertainty in Information Integration (UII) project. His research interests include biomedical data integration, bioinformatics, artificial intelligence, knowledge representation, and their applications to biomedicine. He received his Ph.D. in Spring of 2008.