# Abstract

## Proteomics Data Interoperation with Applications to Integrated Data mining and Enhanced Information Retrieval

Andrew Kendall Smith

2006

This thesis addresses the problem of data integration and interoperation of large-scale, widely distributed and independently maintained data, focusing on biological proteomics data which exemplifies the problem and has a practical need for better interoperation, and shows how such integrated data can be leveraged for important applications such as detailed cross-database queries in support of scientific exploratory data analysis and enhanced information retrieval. Semantic web RDF and RDF databases, which fit the problem well, are used to build two biological data integration systems called YeastHub and LinkHub. YeastHub is a lightweight semantic web data warehouse of joined RDF-formatted biological (yeast) data and allows RDF query access to it. LinkHub focuses on a high-level structuring principal or "scaffold" for biological data, storing biological identifiers (e.g. for proteins, genes, etc.) and the complex relationships among them as a large RDF directed labeled graph; LinkHub is used through web interactive and query interfaces and also complements YeastHub. Through several non-trivial RDF queries of the joined YeastHub and LinkHub data, we demonstrate that practical integrated biological data analysis can be achieved by basic, lightweight methods which don't attempt to solve the complete integration problem.

A key focus of the LinkHub system is support for enhanced information retrieval of web documents and articles from the biomedical scientific literature (PubMed). We

attach documents to identifier nodes in the LinkHub RDF graph and provide for the flexible retrieval of the documents through queries of the RDF graph's relational structure. In addition, we use the LinkHub RDF relational data and attached documents as training sets to construct classifiers for document relevance ranking in support of enhanced automated information retrieval of web or biomedical scientific literature documents related to biological identifiers. The results of experiments done to empirically measure the performance of this enhanced automated information retrieval for proteomics (UniProt) identifier-related documents through the use of a manually curated bibliography of yeast protein-specific literature citations are presented.

# Proteomics Data Interoperation with Applications to Integrated Data mining and Enhanced Information Retrieval

A Dissertation

Presented to the Faculty of the Graduate School

of

Yale University

in Candidacy for the Degree of

Doctor of Philosophy

by

Andrew Kendall Smith

Dissertation Directors: Martin Schultz and Mark Gerstein

Dec 2006

# Contents

# List of Figures

## Acknowledgements

I sincerely thank my advisors Mark Gerstein and Martin Schultz for their guidance and help for completing my thesis and Ph.D. degree. Mark's energy and passion for doing research was inspiring and infectious and was a great motivation for me during my thesis research. It has also been a real pleasure being a member of Mark's lab for the last three years where I have learned and experienced much for which I am grateful. I thank Martin for his invaluable help, advice, technical insights and suggestions, and critical assessments of my thesis work, especially during the latter stages of my research (the "end game"); I also thank Martin for helping to bring out the "CS" in my CS Ph.D. thesis. I thank Drew McDermott, a member of my committee, for his careful, detailed, and insightful reading of early drafts of my thesis by which I was able to greatly improve the thesis. Steven Brenner, in whose computational biology lab I worked before returning to Yale to complete my Ph.D., was my external reader, and I thank him for his help in that capacity but also for teaching me bioinformatics and especially for encouraging and helping me to return to Yale to complete my Ph.D. I thank Kei Cheung, who I have collaborated with since returning to Yale three years ago, for his help and guidance on much of the work in this thesis. I also thank Kei for introducing me to and helping me learn about the semantic web through many discussions and also meetings of the Yale Semantic Web Interest Group (Yale SWIG) which he founded; I feel the semantic web has great potential for advancing science and hopefully this thesis is a small but useful step in that direction. I thank Michael Krauthammer for valuable insights and ideas which were important for the enhanced automated information retrieval aspects of the thesis (chapter 5). I thank fellow CS Ph.D. student and Gerstein Lab and Yale SWIG member

Kevin Yip who was a key collaborator on the YeastHub work of chapter 3 and also helped with the conversion of the relational (MySQL) version of LinkHub to RDF, integrating it into YeastHub, and writing and executing the demo RDF queries over the joined YeastHub / LinkHub data. Finally, I thank my family, my beautiful and talented wife Hehsun and our adorable baby boy Joshua, and my parents and brothers and sisters, for providing love and encouragement during the sometimes difficult road of my Ph.D.

# Chapter 1 Introduction and Overview of the Thesis

## *1.1 Integration and Interoperation of Large-scale, Distributed, Independently Maintained Data and its Application to Enhanced Information Retrieval*

### 1.1.1 Classical Approaches to Integration and Interoperation: Data Warehousing and Federation

This thesis addresses the problem of integration and interoperation of *structured, relational data* with particular emphasis on situations where the data of interest is large-scale, widely distributed and different parts of it are independently maintained by many different groups and persons. The thesis describes software systems which cross-reference and integrate such data, manage and maintain it, and provide various interfaces to access it such as through query languages and web interactive interfaces; these systems enable important applications, which will be covered in more detail below, such as detailed cross-database queries in support of scientific exploratory data analysis and enhanced information retrieval or web search. Two well-known general approaches for integration and interoperation of data are *data warehousing* and *federation[1]*. Essentially, data warehousing involves combining multiple distinct databases or datasets (translating to common format and cross-referencing as necessary) into a single, central location, joined under a single unified schema, where they can be commonly accessed and queried. With federation, source databases maintain their integrity but provide all or part of their data in common, structured ways via APIs, XML dumps, etc. (or they provide some kind of standardized query access to it); the integrated view of all the data

is done by a "virtual database" process which accesses each source database to form a logical composite of all of them. For vast, widely distributed and independently maintained data which is the emphasis of this thesis, however, complete centralized integration of the data by traditional data warehousing is impractical and federation must be a part of any solution; we must somehow achieve partial and incremental integration in cooperative, loosely coupled ways by the independent maintainers of the data.

### 1.1.2 Integration and Interoperation Advantages of the Semantic Web: Standardization and Incremental Data Warehousing

The largest example of large-scale distributed data is the World Wide Web (or just "the web") [2], but it does not consist of structured, relational data but rather of an enormous amount of *unstructured, free-text data* (i.e. web documents in HTML) on a myriad of topics. Given the heterogeneous and generally unstructured nature of the web's content, and the web's huge size, the current dominant paradigm for interacting with and finding things on it is by web search engines [3] which are effective at providing coarse-grained topical access to web content; hyperlinks are also a simple, commonly used but limited way to connect and cross-navigate web data. Search engines and hyperlinks, however, do not enable fine-grained cross-site analysis of data. The *semantic web* [4-6] being propelled by the World Wide Web Consortium or W3C [7] allows web information to be expressed in fine-grained structured ways so applications can more readily and precisely extract and cross-reference key facts and information from it without having to worry about disambiguating meaning from natural language texts. Standard and machine-readable ontologies, which are formal specifications of the objects and their relationships and attributes in some domain[8, 9], are also created and their common use, reuse and

extension encouraged to further reduce semantic ambiguity.

Semantic web technologies are important for the data interoperation problems of this thesis because they support what might be termed *incremental data warehousing* as opposed to traditional data warehousing. Traditional data warehousing based on relational database technology requires as its first and most difficult and time-consuming task the creation of a single grand unifying schema to which all the source databases must be mapped. While such an approach can work well for corporations or other groups for which all source data is under internal, central control such an approach to integration of large-scale distributed and independently maintained data is impossible. In the extreme such an approach would be impossible for general interoperation of the vast amount of distributed data on the web of all types --- the primary goal of the semantic web technologies, in fact, is to support such interoperation of distributed web data.

RDF [10] is the core technology of the semantic web and it is used for the systems described in this thesis. RDF models data as a directed labeled graph where the graph's nodes and edges are named by URIs [11] and it makes no a-priori assumptions (beyond nodes and edges being named by URIs) about constraints on the graph. The RDF graph's edges or "triples" consist of one node, a directed edge from that node, and another node (or a scalar value) pointed to by the edge (see figure 5b for a simple example RDF graph). RDF is similar to relational modeling in that both store and manage relationships among entities (RDF and relational modeling in fact have equal modeling power, and systems such as D2RQ [12] can convert between them; also, e.g., RDF can be stored in relational databases in a "triples" table as can be done in RDF databases such as Sesame [13]). The key difference is that relational modeling requires that all data be structured

into different tables and typed through a predefined schema (relational modeling thus has a *closed world* assumption about data), whereas RDF does not require such a priori typing or structuring of data (RDF thus has an *open world* assumption about data[1]). Thus, no actual instance data can be inserted into a relational database unless a schema has been used to define precisely the structure and typing of the database, whereas no such predefined schema is necessary for RDF databases and RDF triples of any types of objects and relationships can be inserted at will into them. A predefined schema does enable efficient query execution by allowing fine-grained index structures to be pre-built, but this comes at the expense of flexibility and effective index structures can still be built for RDF. The fact that edges (or relationships or properties), in addition to nodes, in RDF are uniquely nameable by URI (i.e. they are so-called "first class objects") is also noteworthy and enables truly distributed data structures where anyone can make statements about any object [14].

While RDF is open and flexible it is nevertheless possible to define structure and constraints on it through the use of the higher level semantic web technologies RDF Schema (RDFS) [15] and Web Ontology Language (OWL) [16], and the use of these can allow one to achieve, and in fact go beyond, the modeling power of relational databases --- the key is that there is a spectrum of levels of structure and constraints that one can specify for RDF data. It is precisely this spectrum that enables incremental data warehousing, where one can specify just the partial (or none) structure and constraints that are known, leaving the rest for later, and still be able to effectively integrate, store

[1] Note that given the huge and distributed nature of the web (which the semantic web aims to be part of), the open world assumption of semantic web technologies including RDF is necessary and a closed world assumption would be impossible or impractical; also, the open world assumption is important for promoting flexible and independent ontology reuse and extension.

and use data. This spectrum is in fact an important principle explicitly supported on the semantic web, called the *principle of least power* which says essentially to use the simplest technology that will achieve your goals; i.e. powerful, expressive languages inhibit information reuse while simple but useful ones support it [17]. Thus, the systems built and described in this thesis are able to use the semantic web to make valuable partial and incremental progress towards practical data interoperation without having to solve the entire problem at once.

Data warehousing, even the incremental type enabled by the semantic web, nevertheless still involves centralized integration and storage of data. However, it is impractical to consider data warehousing, even in limited, partial ways, as the complete solution for integration of large-scale widely distributed and independent data. It is assumed that no single person or group can have knowledge of all the data or the requisite large resources and time necessary to collect it together into one location: there is simply too much of it and it is growing too fast. At least some federation will have to be part of the solution and we thus must accept and rely on individuals and groups independently releasing data. The only practical way this can lead to some level of global, federated integration is if they all release their data using common standards: standards and their widespread use enable independent, loosely coupled cooperative integration. It would be counterproductive to invent new standards if not necessary, and in fact the goals of this thesis are a very good fit to the semantic web. The semantic web is strongly supported by the W3C and it is increasingly gaining traction as a key platform for data integration, particularly for biological data which is the domain of investigation of this thesis as described below.

### 1.1.3 Hybrid Data Warehousing and Federation

This thesis advocates a combined, hybrid federation and data warehousing approach. A warehousing approach is best both for large, well-known and important databases and also in small, local contexts (e.g. all of a lab's resources, sets of resources on common topics, etc.): the warehouse of large, important, well-known databases can serve as core "backbone" content (i.e. major hubs of data) to which the smaller, local warehouses can connect in a federated fashion, thus enabling global integration of data in an efficient loosely coupled, cooperative way. Efficiency is gained both by eliminating the need for all single source datasets to be individually connected to one another directly (efficiency is gained through the indirect connections to the major hubs) or each directly to the major hubs (the small, local warehouses make a single connection on behalf of all their contained datasets).

### 1.1.4 Information Retrieval and Web Search Enhanced by the Semantic Web

As discussed above, the web and the semantic web are two opposing models for web data, with the web being maximally flexible by consisting of unstructured, free-text data while the semantic web prescribes web data to be expressed in fine-grain structured ways. Search engines are the dominant paradigm for indexing and retrieving web data and documents, while query languages (e.g. the proposed standard RDF query language SPARQL [18]) and inferencing engines (e.g. Racer for reasoning with OWL [19]) are how semantic web data is accessed and used. In the search engine paradigm one tries to get computers to automatically extract meaning and facts from unstructured texts, whereas the semantic web takes the stance that this is not a panacea and we need to

encourage people to disseminate their information and data in a more precise and structured way.

The two approaches basically have inverted strengths and weaknesses. Search engines, because they are fully automated, can achieve vast, close to complete coverage of the web and they allow people to disseminate their content in natural language texts without requiring them to learn possibly arcane data structuring languages and techniques; the key drawback is that, because it is currently too difficult to get computers to extract any detailed meaning from unstructured texts, the level of precision in information requests achievable by search engines is quite limited. The semantic web, on the other hand, allows information and data to be expressed and queried very precisely, and its ontologies promote cooperation and further reduce ambiguity in meaning. The drawback of the semantic web is that, to achieve this fine grain information modeling humans must change the way they create their content to conform to very precise structures, and this is a hindrance to widespread dissemination of semantic web content; consequently the semantic web is still fairly small and it is arguable whether it will ever gain traction and become on par with the standard web.

Currently, the search engine and semantic web worlds are generally separate and do not interact with or make use of each other's technology. As discussed above, the search engine and semantic web worlds largely have strengths and weaknesses that are complementary, with a weakness in one being a strength in the other. This thesis proposes that these two approaches to web data management and retrieval can work together and complement one another and that there are interesting, practical, and useful ways the semantic web and search engine worlds can work with, leverage, and enhance

each other; in particular, the thesis explores how semantic web data can be used to enhance information retrieval or web search. The basic idea is that the semantic web provides detailed information about standard terms and their interrelationships, and, importantly, unstructured web documents can be annotated with those terms as metadata. The terms, their relationships, and the documents that they annotate provide copious information to perform precise information retrieval or web search for free-text documents relevant to those terms (and related terms). This will be explained more concretely below in the biological context. In the future when the semantic web becomes more widespread and structured, relational data about various concepts of interest becomes widely available and easily accessible, web searchers can piggyback on this wealth of preexisting semantic web data, picking and choosing standard terms and their relational subgraphs as they like to use in improving document retrieval. Since search is widely perceived to be such a crucial web application the semantic web's ability to improve search could be of high practical value and an important driving force to help more fully realize the vision of the semantic web. Integration and interoperation of data such as terms, relationships, and annotated web documents are a prerequisite for this vision, however, and this thesis builds practical software systems to support it.

## *1.2 Biological Data*

The data domain of focus in the thesis is biological data, with an emphasis on proteomics data. Biological research is producing vast amounts of data, e.g. from high throughput experiments such as genome DNA sequencing projects and DNA microarray experiments, at a prodigious rate. Most of this data is made freely available to the public, and this has created a large and growing number of internet and web-accessible

biological data resources which are characterized by being distributed, heterogeneous, and having a large size variance, i.e. huge, mega-databases such as UniProt [20] and GenBank [21] down to medium, small or "boutique" databases (e.g., Pfam [22], SGD [23] and TRIPLES [24]) generated for medium / small scale experiments or particular purposes. Biological data thus closely matches the above stated emphasis of the thesis and, in addition, there is a pressing practical need to be able to interoperate across it, at least in basic ways, e.g. to better support computational drug discovery. Biological data is also a good domain to work in because it strikes a good balance for complexity of the problem: it is not excessively complex but is still complex enough to be interesting with the potential to achieve useful practical results.

## *1.3 Interoperation of Biological Data*

### 1.3.1 LinkHub and YeastHub

This thesis presents a software architecture and prototype system called LinkHub inspired by the hybrid warehousing and federation approach discussed above. The LinkHub prototype, accessible at either http://hub.gersteinlab.org or http://hub.nesg.org, is practically used to connect together a number of proteomics and related web resources (providing a single point of entry to them all) and connect them all to the major proteomics hub UniProt (see figure 4). The full integration problem is very difficult, however, and could require too much background knowledge of database owners and be too complex for people to bother with, and this could inhibit interoperation of proteomics data. If we must rely on cooperative, loosely-coupled integration, which is the position this thesis takes, then practically we must make our integration mechanisms and

standards as simple as possible, otherwise we risk people not using them. LinkHub does not attempt to solve the complete integration problem for biological data, but instead focuses on a basic but important, high-level structuring principle for biological knowledge, namely identifiers for biological entities (e.g. proteins and genes) and the relationships (and relationship types) among them. Together the identifiers and their relationships form a large directed graph, and this graph serves as an essential "scaffold" to which other types of proteomics data can be linked and interoperated across (see figure 3 for a conceptualization of this graph). In particular, access to additional attributes and data for biological identifiers is through hyperlinks to identifier-specific pages, where these hyperlinks are linked to identifiers' nodes in the graph. LinkHub is a system for storing, managing, and exploring, via web interactive interfaces and query languages, such identifiers and their relationships, and identifier node-linked data. The basic conceptual underpinnings of LinkHub, i.e., the importance of biological identifiers and connecting biological databases by linking them, was given in [25] and LinkHub is a practical system based on and extending these ideas.

While LinkHub is a free-standing system useful on its own as just described, it also serves a complementary role to another system described in this thesis called YeastHub [26]. YeastHub was begun before LinkHub and it attempted to address data integration more generally by transforming datasets to a common RDF format and storing and providing query access to them all together through an RDF database and RDF query languages; YeastHub was essentially a lightweight data warehouse for yeast and other genomic data, based on and using semantic web technologies and serving as a test bed for integrated yeast and genomic data analysis through semantic web technologies. The

problem with YeastHub, however, was that, although it was able to co-integrate many disparate datasets the integration was thin --- the key was numerous and varied connections among the integrated datasets and these were limited in the original YeastHub system. This is precisely the role that LinkHub plays, and it is thus useful and complementary to YeastHub as a "connecting glue" among the datasets in that it makes and stores these cross-references and enables improved integrated access to the YeastHub data. Access to the combined YeastHub / LinkHub data, which is stored in the Sesame RDF database [13] is by RDF query languages. We give demonstration queries below written in SeRQL (Sesame's query language related to RQL) [27] to show one can effectively do the kinds of interesting preliminary scientific investigation and exploratory analysis commonly done at the beginning of research initiatives (e.g. to see whether they are worth pursuing further). These queries make use of information present in both YeastHub and LinkHub (and thus could not be done without joining the two systems), and, again, LinkHub is used as 'glue' to provide connections (both direct and indirect) between different biological identifiers. It is noteworthy that the demo queries roughly duplicate some results from published papers; taking advantage of the combined YeastHub and LinkHub data the queries can be formulated and run to get results in at most a few hours, which is in stark contrast to the extensive effort (days or weeks) likely required for the papers to manually integrate the necessary data to achieve their results.

### 1.3.2 Ontology Alignment of Biological Identifiers in LinkHub

Data warehousing and federation, while important, deal with issues such as the physical location and organization of integrated data (i.e. centrally located versus distributed) and what gets translated (i.e. data or queries). However, a separate, more fundamental issue in

interoperation is determining what the relevant entities are and the attributes and properties they possess. This is precisely what formal ontologies are for, and ontologies are what the various languages and technologies of the semantic web, such as RDF, RDFS and OWL, are meant to allow one to create. An important remaining problem not solved directly by the formal ontology building standards of the semantic web is the problem of **ontology alignment** [28-30]: how to map the entities and concepts in one independent ontology to equivalent or similar entities and concepts in another independent ontology.

Ideally, the ontology alignment problem would not arise: everyone would agree on the relevant terms, relationships, and attributes and what they all mean and would use only them, so there would be no need for such mapping. However, practically the problem does arise in spite of the fact that ontology reuse is strongly encouraged. Because the semantic web deals with huge, distributed, independently maintained data, different groups and people will independently invent and use their own ontologies, and terms and relationships and properties for the same underlying essential concepts can be reinvented many times in different ways. It is also impractical to rely on a single, grand unifying ontology: it is very difficult to create, maintain, and use very large ontologies (although people have certainly tried, exerting much time and effort towards creating them, most notably in the CYC project [31]) and it is more practical to use smaller, partial ontologies that can deal with your particular problem. People also simply see the world differently, and will structure their data consistent with how they view the data's particular domain, which can often be different than how another person views the same domain. It is also arguable whether it even makes sense to try to "force" one particular

view or ontology on data. A recent trend on the web is for "organic organization" through tagging or *folksonomy* systems such as del.icio.us [32] and Connotea [33] where users independently assign multiple, free form textual tags to web pages and other items of interest, thus allowing multiple different views on the same data to emerge by the collaborative efforts of many users; in effect, the ontologies used to annotate web pages and other objects in such tagging systems are an emergent property of the independent actions of the many users, and one can look for interesting relationships among the terms (e.g. which terms co-occur often, etc.), in effect doing a kind of ontology mapping [6, 34]. In any case, there is a need for ontology alignment on the semantic web [29]

A commonly used basic technique for ontology alignment in general is to do some kind of string matching [35], e.g. if one dataset identifies some object as "056-94-8945" and another as "056948945" then we might infer they refer to the same object (i.e. the same person with the given social security number). Biology is blessed with a fundamental, commonly accepted principle around which data can be organized, namely biological sequences such as DNA, RNA, and protein, and various string matching techniques for biological sequences (see appendix A for some details) can solve a large part of the ontology alignment problem in biology. LinkHub uses such string matching of biological sequence for its ontology alignment problems. In particular, for LinkHub, how can we determine that two biological identifiers refer to the same entity (i.e. are synonyms for that entity) or that the identifiers are related and what the relationship type is? Again much biological data can be effectively organized around biological sequences and LinkHub takes advantage of this.

LinkHub incorporates the full UniProt database and uses it as "backbone" content. The large UniProt staff performs manual mapping (using biological sequence matching to automate or semi-automate much of the process) of biological identifiers for large, well-known databases, mapping UniProt protein sequence identifiers to other important databases such as PFAM [22] (protein domains or modules which occur in many species' proteins by evolution), GO [36] (standard taxonomy of terms for annotating protein and gene function), and PDB [37] (protein 3-D structure). LinkHub can directly take advantage of UniProt's many inter-database mappings by mapping the sequences for its own identifiers to UniProt's sequences (UniProt identifiers are keys to records which contain much information about the proteins, such as their sequence, sequence properties, and of course the cross-references to other identifiers in other biological databases). This also would allow indirect connections to be made to other small databases (or other LinkHub instantiations) which themselves linked their identifiers to UniProt's --- in fact this kind of indirect mapping is how global interoperation among biological databases can be achieved in a cooperative, loosely-coupled way. LinkHub focuses on exact sequence matching, which is conservative and guarantees that two identifiers referring to exactly matched sequences are referring to exactly the same entity. Non-exact matching sequences which nonetheless still share much sequence identity are handled by connections through PFAM; UniProt maps its identifiers to PFAM domain identifiers, and LinkHub proteins indirectly map to PFAM identifiers through their connections to UniProt (and thus two proteins which both map to the same PFAM identifier indirectly through connections to UniProt identifiers are evolutionarily related, i.e. share a common domain and are members of the same family). While UniProt is a primary source for

identifier mappings, when available LinkHub also similarly takes advantage of other identifier mappings that have been precomputed (e.g. mapping of pseudogenes to UniProt). Finally, for relationship types LinkHub tries to be flexible and, similar to folksonomy tagging systems, does not force the use of any given ontology but allows the relationships between identifiers to be specified with free text such as "family mapping" or "functional annotation" (although the free text could be used to specify formal terms in an ontology); LinkHub could also be said to favor and support *shallow ontologies* as opposed to *deep ontologies* [6]. More details of LinkHub's identifier mapping methods will be given in chapter 4.

## *1.4 LinkHub enables Enhanced Information Retrieval*

A key theme of this thesis is providing enhanced information retrieval to unstructured, free-text data (e.g. the scientific literature of journal and conference articles, web pages, etc.) using the information present in the graph of identifier relationships and identifier node-linked documents that are stored in LinkHub. The most common and prevalent access to free text documents is currently by search engines, i.e. where users enter words they want to search for and the search engine returns documents that contain those terms many times and in prominent locations or lexically close together, etc. Google [38] and Yahoo [39] are the most well-known general purpose search engines for web content, and PubMed [40] provides such keyword-based search access to the biomedical scientific literature (Medline). However, while such keyword-based search access to free text documents often works well and is sufficient to produce a user's desired information, there are other important cases where it does not. In fact, simple keyword-based search lacks precision and can return millions of documents with poor ranking, and this was a

big problem with early web search engines; it was Google's solution to this problem [41] of effective ranking of huge result sets, using hyperlinks as "votes" of importance for linked to pages, which made Google the de-facto only web search engine most web surfers use today.

Even if you can rank large result sets well there are certain queries that cannot be done by keyword-based search engines. For example, imagine the conceptual query "Return to me all documents containing information for proteins which are members of the Pfam Adenylate Kinase (ADK) family." You could not directly do this query with just keyword-based search access. You could imagine entering terms "Pfam", "adenylate", and "kinase" which might get you some of the relevant documents but in a jumbled order, and would also return unrelated or more generally related documents (e.g. a page generally describing what kinases are, what Pfam is, etc.) What the query is really asking for is to first determine the proteins which are members of the ADK family, and then find the relevant documents for each of these separately, and combine all these into a single result set; the initial part of this query, determining proteins in Pfam family ADK, requires access to relational information about family memberships of proteins and then finally a search for documents relevant to these proteins can be done. However, even when the individual proteins in the ADK family are known, keyword-based search will likely not be very effective at returning relevant documents for these proteins. Proteins, being important biological entities, are referred to by identifiers and because of conflated senses of the identifier text, identifier synonyms, and in general a need to consider and query for the key related concepts of the identifier, simple search for protein identifiers will likely not return good results (or, at the very least, suffer the standard problem of

returning millions of results with poor ranking).

This thesis shows how the relational data in the LinkHub graph of identifier relationships can be used for enhanced information retrieval. Recall that LinkHub stores a graph of identifiers and their relationships, where hyperlinks to web documents are linked to identifier nodes in the graph. The key point of the ADK family query above was that it required access to relational data about proteins' family memberships, in addition to keyword-based search techniques. LinkHub provides a special query access to the graph that allows one to flexibly retrieve useful subsets of web documents (which are linked to identifier nodes in the graph) based on the relational structure of the graph, and this supports queries such as the ADK example which are partly relational in nature.

The thesis also shows how enhanced, automated retrieval of documents from the web or scientific literature related to proteomics identifiers can be done using the LinkHub graph. The key idea is that the LinkHub subgraph emanating from a given identifier and the web pages (hyperlinks) linked to the identifier nodes in that subgraph provide copious and detailed information about the given central identifier that can be used to perform precise and accurate relevant document retrieval for it. The web pages linked to the identifiers' nodes in the subgraph are considered to be a "gold standard" for what additional relevant documents should be like. These identifier node-linked web pages are used as training sets to construct ranking functions, in particular a combined word weight vector of all of the web pages (where each of them is weighted to indicate how relevant they are to the central identifier). This combined word weight vector ranking function is used to score and rank additional documents (obtained from the web or scientific literature) for how well they match the training set (and hence the central

identifier).

The thesis also proposes a novel step in information retrieval, called the pre-inverse document frequency step (or just pre-IDF step), which is shown to greatly increase document relevance ranking accuracy (section 5.3 below covers inverse document frequency and other basic techniques of information retrieval). This step takes advantage of the fact that, for a given identifier for which we want to retrieve relevant documents, we not only know that identifier's subgraph and the documents linked to identifier nodes in that subgraph, but we also know the subgraphs and subgraph identifier node-linked documents for many other identifiers of the same type (e.g. we know the subgraphs and identifier node-linked documents for all the UniProt proteins). In essence, the pre-IDF step takes advantage of this "big picture" information to maximally separate the word weight vectors of all documents of the same type while at the same time making them as specifically relevant and discriminating as possible. Finally, linking automatically retrieved documents to the identifier nodes of the identifier for which they were retrieved can further enhance the utility of information retrieval queries that use the relational structure of the LinkHub graph (i.e., like the ADK example), since this makes a larger set of free-text documents available for retrieval (LinkHub initially only links a small number of hyperlinks to identifier nodes).

## *1.5 Organization of the Thesis*

Given the preceding high-level overview of this thesis, the remainder of the document is organized as follows. Chapter 2 will present a particular, exemplary proteomics data mining analysis, predicting the tractability of proteins for experimental three dimensional structure determination from known sequence-based features. This is given to

demonstrate the kinds of problems computational proteomics analysis can address and the data used for it; Appendix 1 contains some more detailed background information which can be read before chapter 2 if desired: an overview of proteomics and proteomics databases, and an overview of some representative and important computational proteomics problems. The presentation of this data mining analysis in chapter 2 motivates the problem of a need for better interoperation of proteomics data, since a large part of the analysis was spent simply manually assembling the dataset to perform the analysis, which is addressed by the YeastHub system described in chapter 3. In particular, chapter 3 will first discuss the challenges to data interoperation, general approaches to data interoperation, and the semantic web approach to data interoperation before finally describing the YeastHub system in detail. In building YeastHub there were still some important remaining issues that it didn't address that we identified, the most important one being a dearth of connections and relationships among the datasets integrated in YeastHub, and this served as a motivation for the LinkHub system which is the subject of chapter 4. Chapter 4 will cover the main idea behind LinkHub, the concept of biological identifiers and the relationships among them forming a large graph that is an important "scaffold" of biological knowledge and data. Chapter 4 will present the details of LinkHub, including its RDF and relational data models, how it handles ontology alignment (i.e. the relationship mapping) of biological identifiers, its web interactive and query interfaces, and its combination with YeastHub to better enable useful scientific exploratory data analysis of proteomics data (example queries of the combined YeastHub / LinkHub system will be demonstrated). Chapter 4 will also discuss how LinkHub enables novel information retrieval based on the LinkHub relational graph structure to

web documents linked to identifier nodes in the graph. Chapter 5 will cover the enhanced automated information retrieval aspects of LinkHub, in particular how LinkHub relational subgraphs can be used as training sets to construct ranking functions for document relevance ranking for identifier-related documents (appendix 2 lists the top 20 results of a LinkHub-based search of PubMed for documents related to a UniProt proteomics identifier). An overview of basic techniques in information retrieval, and the details of the pre-IDF step, will also be given. The results of experiments done to empirically measure the performance of the enhanced automated information retrieval for identifier-related documents through the use of a manually curated bibliography of yeast protein-specific literature citations will be presented. Finally, chapter 6 will conclude by summarizing the key points and contributions of the thesis in the context of related work, and give some possible future directions for research.

# Chapter 2 Structural Genomics Data mining as a Motivating Problem for Proteomics Data Interoperation

## *2.1 Structural Genomics Data mining: Predicting Tractability of Protein Targets for Experimental Structure Determination*

Appendix 1 contains an overview of proteomics and important proteomics databases, and covers a sample of important and representative problems in computational proteomics; appendix 1 can be read to obtain background knowledge for this chapter. This chapter will focus on a particular, exemplary proteomics data mining problem, predicting the tractability of proteins for experimental three dimensional structure determination from known sequence-based features. This is given to more fully present the kinds of problems computational proteomics analysis can address and the data used for it, and it motivates the problem of the need for better interoperation of proteomics data, which will be addressed by the YeastHub system in chapter 3, since a large part of the analysis was spent simply manually assembling the dataset to perform the analysis.

Structural genomics is a government sponsored initiative funded by the NIH to efficiently solve 3-D structures of representatives of protein families at high throughput using the experimental techniques of crystallography and nuclear magnetic resonance (these are two, somewhat complementary, experimental techniques for solving structures) [42-44]. The Gerstein Lab participates in one of the government-funded structural genomics centers, the Northeast Structural Genomics Consortium or NESG [45], and has the role of handling the information infrastructure and performing data management for it, most notably by maintaining the SPINE internal NESG target tracking database [46].

SPINE data is also fed into and is part of the public TargetDB structural genomics tracking database [47]. Another role of the Gerstein Lab in the NESG is to perform data mining of structural genomics data (from SPINE and/or TargetDB and PepcDB [48]) to understand and improve the process. In particular, solving a protein's structure experimentally takes much time and resources, and there is no guarantee of success (a large number of protein targets are abandoned after a number of steps of the experimental determination pipeline because they are found to be intractable for some reason). If we could better predict which protein targets are more likely to be successful ahead of time, then we could focus on these and there would be less waste of resources and time spent on "dead end" proteins; this was a key motivation for this structural genomics data mining analysis.

Proteins in structural genomics go through a lengthy pipeline of experimental stages towards 3-D structure determination; the standard pipeline stages as defined by TargetDB for crystallography are: **Selected → Cloned → Expressed → Soluble → Purified → Crystallized → Diffraction-quality Crystals → Diffraction → Crystal Structure → In PDB**. The pipeline for NMR based solution is the same except the 4 stages before **In PDB** are replaced by: **NMR Assigned → HSQC → NMR Structure**. Other statuses that can be assigned to TargetDB protein targets are **Work Stopped, Test Target, and Other**. TargetDB, then, provides status data (i.e. which stage in the pipeline a protein has reached, dates for reaching various stages and statuses, etc.) for all targets of the NIH-funded structural centers (this is a requirement of the centers' funding) and also from other US and international efforts who voluntarily provide the data to TargetDB for their targets. TargetDB was the primary database used in the data mining analysis, and

the basic idea of the analysis was to find protein sequence-based features (which could be determined or accurately predicted from only protein sequence data, which is thus known ahead of time before a protein enters the structure determination pipeline) that were predictive of success in TargetDB. I.e. which features differentiated proteins that were successful at having their structures solved versus those which were not, and for each pipeline stage which features were predictive of proteins reaching the next stage versus proteins which did not reach the next stage?

The protein sequence-based features which were considered include:

- The percentage composition of particular amino acids and groups of chemically similar amino acids (charged, hydrophobic, etc.)
- The length of the protein.
- Whether the protein is a member of a COG ("Cluster of Orthologous Groups" --- a separate database [49]).
- Whether a protein had or was predicted to have binding partners or be in a protein complex (i.e. was part of a protein-protein interaction). The MIPS complex catalog [50] and a number of other datasets were the source of the interaction information.
- The protein's isoelectric point (pI).
- Hydrophobicity scores as measured using the GES scale [51].
- Presence of motifs from the PROSITE database [52].

- Presence of nuclear localization signals and other signal sequence patterns (e.g. presence of charged amino acid in first 7 amino acids followed by 14 hydrophobic amino acids).
- Entropic low-complexity sequence scores calculated using the SEG program [53].
- Secondary structure: percentage of amino acids in helix, beta sheet, and coil regions.

We thus constructed a dataset of the above features and pipeline statuses for all the structural genomics targets in TargetDB and used machine learning to construct classifiers for predicting structural genomics tractability for proteins. In particular, we used decision trees as the classifier model because they give reasonable performance and, more importantly, they are easily interpretable unlike other machine learning models such as neural networks and support vector machines which are essentially black boxes; for decision trees, easily understandable "if-then" rules can be extracted and used. Interpretability of what were the important features was an important goal of the analysis, and decision trees fit this goal. The random forest algorithm is a decision tree based method based on bootstrap aggregating ("bagging") and random feature selection which is a robust way of finding the strongest, most predictive features, and we used it to perform feature selection. Decision trees were then constructed using the features selected by the random forest analysis.

The overall results of the analysis found the following features to be the most important predictors of successful structural characterization:

- **Conservation across organisms (i.e. presence in a COG).** This is likely because larger protein families are more studied. It also makes it more likely there will be an amenable bacterial representative in the family (the cloning and expression experimental stages are generally done in bacteria), or maybe simply that there will be multiple proteins from the family you can try until you eventually hit on one that happens to work.
- **Hydrophobicity**. This is likely because highly hydrophobic proteins are less soluble, and so are more likely to fail the solubility stage.
- **Presence of charged amino acids**. This is likely because more charge improves solubility.
- **Number of binding partners**. A protein that has binding partners likely requires them to be present in order to fold correctly and thus be crystallizable, and these partners would not be present in the structural genomics pipeline.

It was thought that the following features would be important, but in the end they were found not to be:

- **Low complexity sequences**. Low complexity sequences probably won't fold well and this would seem to be a predictor of intractability. However, it is likely low complexity sequences were filtered out during target selection.
- **Presence of nuclear localization signal motifs**. The presence of such signals indicates a secreted protein which structural genomics does not consider (they aren't targeted).

- **Percentage of lysine and arginine amino acid**. These are charged amino acids, so it was plausible to think they would aid solubility and be good predictors, but they did not turn out to be.

The full details of this structural genomics data mining analysis are in [54]. The important point for this thesis is that the analysis required extensive time and effort just to assemble the dataset, likely consuming a majority of the total time and effort spent. A number of different datasets had to be found, understood, and combined, and a number of external programs had to be run to calculate certain features. Bioinformatics researchers have to do this kind of time-consuming manual integration of data all the time. This was one of the motivations for the work in this thesis, to come up with practical and workable solutions for integrating biological data and enabling data interoperability to support efficient creation of datasets, so researchers can spend most of their time where it is most useful, actually doing their analysis. The next chapter will cover a system called YeastHub, a data warehouse for integrating and querying biological data based on semantic web technologies, which attempts to address such issues.

# Chapter 3 YeastHub

This chapter describes the challenges involved in the integration of databases storing diverse but related types of life sciences data. A major challenge in this regard is the heterogeneity of life sciences databases. There is a strong need for standardizing representations of life sciences data, both syntactically (i.e. encouraging the use of standard data formats such as XML) and semantically (i.e. precisely defining the meaning of terms and their relationships, and encouraging their widespread use). This thesis addresses the need for such standardization by using the emerging semantic web technologies based on the Resource Description Framework (RDF) standard. This chapter presents a system called YeastHub [26] which demonstrates how to use the latest RDF database technology to build a data warehouse that facilitates integration of life sciences data.

## *3.1 Background*

With the popularity and ubiquity of the World Wide Web a large quantity of biological data has been made available to the scientific community through the Internet. A multitude of web accessible biological databases have emerged. These databases differ in the types of biological data they provide, ranging from sequence databases (e.g., NCBI's GenBank [21]), microarray gene expression databases (e.g., SMD [55] and GEO [56]), pathway databases (e.g., BIND [57], HPRD [58], and Reactome [59]), and proteomic databases (e.g., UPD [60] and PeptideAtlas [61]). While some of these databases are organism-specific (e.g., SGD [23] and MGD [62]), others like (e.g., Gene Ontology [36] and UniProt [20]) are relevant irrespective of taxonomic origin. In addition to data

diversity, databases vary in scale ranging from large global databases (e.g., UniProt), medium boutique databases (e.g., Pfam [22]) to small local databases (e.g., PhenoDB [63]). Some of these databases (especially the local databases) may be network-inaccessible and may involve proprietary data formats.

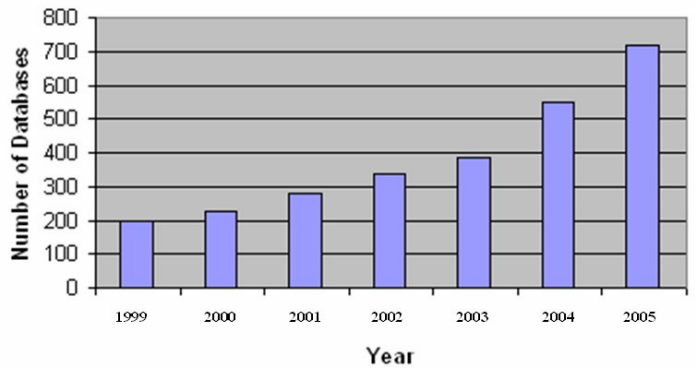



**Fig. 1. Number of databases published in the NAR Database Issues between 1999 and 2005.**

Fig. 1 indicates the rate of growth in the number of web-accessible molecular biology databases which were published in the annual Database Issue of Nucleic Acids Research (NAR) between 1999 and 2005. These databases only represent a small portion of all biological databases in existence today. With the sustained increase in the number of biological databases, the desire for integrating and querying combined databases grows. Information needed for analysis and interpretation of experimental results is frequently scattered over multiple databases. For example, some microarray gene expression studies may require integrating different databases to biologically validate or interpret gene clusters generated by cluster analysis [64].

- For validation, the gene identifiers within a cluster may be used to retrieve sequence information (e.g., from GenBank) and functional information (e.g., from Gene

Ontology) to determine whether the clustered genes share the same motif patterns or biological functions,

- For interpretation, such gene expression data may be integrated with pathway data provided by different pathway databases to elucidate relationships between gene expression and pathway control or regulation.

## *3.2 Data Interoperation Challenges*

Below the challenges faced when integrating information from multiple databases are reviewed:

- **Locating Resources.** Automated identification of websites that contain relevant and interoperable data poses a challenge. There is a lack of widely-accepted standards for describing websites and their contents. Although the HTML meta tag (http://www.htmlhelp.com/reference/html40/head/meta.html) can be used to annotate a web page through the use of keywords, such tags are problematic in terms of sensitivity and specificity. Furthermore, these approaches are neither supported nor used widely by existing web search engines: most web search engines rely on using their own algorithms to index individual websites based on their contents.

- **Data Formats.** Different web resources provide their data in heterogeneous formats. For example, while some data are represented in the HTML format, interpretable by the web browser, other data formats including the text format (e.g., delimited text files) and binary format (e.g., images) are commonplace.

- **Synonyms**. There are many synonyms for the same underlying biological entity as a consequence of researchers independently naming entities for use in their own datasets or because of legacy common names (such as the famous "sonic hedgehog" gene name) arbitrarily given to biological entities before large-scale databases were created. Such names have managed to remain in common use by researchers. An example of this synonym problem is the many synonymous lab-specific protein identifiers used to identify proteins (e.g. the structural genomics centers all give their own names to proteins). There can also be lexical variants of the same underlying identifier (e.g., GO:0008150 vs. GO0008150 vs. GO-8150).

- **Ambiguity.** Besides synonyms, the same term (e.g., insulin) can be used to represent different concepts (e.g., gene, protein, drug, etc). This problem can also occur at the level of data modeling. For example, the concept 'experiment' in one microarray database (e.g., SMD) may refer to a series of samples (corresponding to different experimental conditions) hybridized to different arrays. In another microarray database (e.g., RAD [65]), an experiment may refer to a single hybridization.

- **Relations**. There are many kinds of relationships between database entries, some one-to-one like synonyms but also one-to-many relationships are prevalent. For example, a single Gene Ontology identifier can be related with many UniProt identifiers (i.e. they all share the same functional annotation). An important structuring principle for genes and proteins, which leads to one-to-many relationships, is the notion of families

based on evolutionary origin. A given protein or gene can be composed of one or more family specific units, called domains. For example, a UniProt entity may be composed of two different Pfam domains. In general a given Pfam domain will be related to many UniProt proteins by this family association, and the UniProt proteins can in turn be related to other entities through various kinds of relationships (and similarly for Gene Ontology). A transitive closure in such a relationship graph, even a few levels deep, can identify indirect relationships with a great number of other entities. It is important to note, however, that there are certain relationship types for which following them in the wrong way can lead to incorrect inferences, with the family relationship being a key one. For example, if protein A and B share Pfam domain X, and protein B and C share Pfam domain Y then there is a relationship path from A to C but it is through 2 different Pfam domains, and thus A and C are not really related. One cannot blindly follow relationship links to infer indirect relationships, but must take account of the link types. It is important but a challenge to uncover all the important relationship types among biological entities and be able to correctly reason with them.

## *3.3 General Approaches to Database Interoperation*

Chapter 1 gave a preview of the two general approaches to database integration, namely, the data warehouse approach and the federated database approach, and this section gives some more details. The data warehouse approach emphasizes data translation, whereas the federated approach emphasizes query translation. The warehouse approach involves translating data from different sources into a local data warehouse in common format

under a unified schema and executing all queries on the warehouse rather than on the distributed sources of that data. This approach eliminates various problems including network bottlenecks, slow response times, and the occasional unavailability of sources. In addition, creating a warehouse allows for an improved query efficiency or optimization since it can be performed locally [66]. Another benefit of this approach is that it allows values (e.g., filtering, validation, correction, and annotation) to be added to the data collected from individual sources; this is possible because the data is copied into the data warehouse and can be modified, whereas the source databases themselves are generally read-only. This is a desirable feature in the domain of biosciences. The data warehouse approach, however, suffers from the maintenance problem in light of evolution of the source databases (both in structure and content). The warehouse needs to be periodically updated to reflect the modifications of the source databases and, based on how many databases are present in the warehouse and the complexity of the modifications, there can be considerable lag time in updates preventing users from seeing and being able to use the most up-to-date data. Some representative examples of biological data warehouses include SRS [67], BioWarehouse [68], Biozon [69], and DataFoundry [70].

The federated database approach concentrates on query translation [71]. It involves a mediator, which is a middleware responsible for translating, at runtime, a query composed by a user on a single federated schema into separate queries on the local schemas of the underlying data sources, executing all the separate queries, and combining all the results into a single unified result. A mapping is required between the federated schema and the source schemas to allow query translation between the federated schema and the source schemas. While the federated database approach ensures data is

concurrent / synchronized and it is easier to maintain (when new databases are added), it generally has a poorer query performance than the warehouse integration approach. Some representative examples of the federated database approach include BioKleisli [72], Discoverylink [73], and QIS [74]. A methodological overview and comparison of these two major database integration approaches was discussed in the biomedical context [1].

## *3.4 Semantic Web Approach to Data Interoperation*

The semantic web addresses interoperation by seeking methods to facilitate machine-based identification and semantic interoperability of web resources. Crucial to the semantic web approach is the design and development of ontologies (semantic part) that are represented in computer-readable formats (syntactic part). While the HyperText Markup Language (HTML) is used for providing a human-friendly data display in web browsers, it is not machine-friendly. In other words, computer applications do not know the meaning of the data when parsing the HTML tags, since they only indicate how data should be displayed. To address this problem, the eXtensible Markup Language (XML) was introduced, to allow meaningful tags to be defined and associated with data values. In addition, a hierarchical (element/sub-element) structure can be created using these tags. With such descriptive and hierarchically-structured labels, computer applications are given better semantic information to parse data in a meaningful way.

While XML has become a standard syntax for data exchange between applications it does not adequately address semantics. In particular, XML lacks expressivity for formal knowledge representation and inference. Despite its machine readability, as indicated by [75]., the nature of XML is syntactic and document-centric. This limits its ability to achieve the level of semantic interoperability required by highly

dynamic and integrated bioinformatics applications. In addition, there is a problem with the proliferation of and redundancy of XML formats in the life sciences domain; overlapping XML formats (e.g., SBML [76] and PSI MI [77]) have been developed to represent the same type of biological data (e.g., pathway data).

The introduction of the semantic web has taken the usage of XML to a new level of ontology-based standardization. In the semantic web realm, XML is used as an ontological language to implement machine-readable ontologies in conjunction with standard knowledge representation techniques. The Resource Description Framework (RDF) (http://www.w3.org/RDF/) is an important first step in this direction. It offers a simple but useful semantic model based on a directed graph structure. In essence, RDF is a modeling language for defining statements about objects and the relationships among them. Such objects and relationships are uniquely named and identified using the system of Uniform Resource Identifiers (URIs). Each RDF statement is a triplet with a **subject**, **property** (or **predicate**), and **property value** (or **object**). For example,

*<"http://en.wikipedia.org/wiki/Protein#",*
*"http://en.wikipedia.org/wiki/Name#",*
*"http://en.wikipedia.org/wiki/P53#">*

is a triple statement expressing that the subject *Protein* has *P53* as the value of its *Name* property. An object may be related to many other objects through many different relationship types, and a large group of inter-related objects thus form a directed graph structure where the nodes are objects and the directed edges are the relationships. RDF

also provides a means of defining classes which can be used for both objects and properties/predicates. These classes are used to build statements that assert facts about objects and properties/predicates. RDF uses its own syntax (RDF Schema or RDFS) for writing class schemas. RDFS is more expressive than RDF and it includes subclass / superclass relationships as well as constraints on the statements that can be made in a document conforming to the schema.

Some biomedical datasets such as the Gene Ontology [36], UniProt (RDF available at http://expasy3.isb-sib.ch/~ejain//rdf/), and the NCI thesaurus [78] have been made available in RDF format. The Semantic Web Health Care and Life Sciences Special Interest Group (SW HCLSIG) [79] has been formed as a community effort within the World Wide Web Consortium to promote and develop semantic web use cases in the healthcare and life science domains.

Extensions of RDF, such as the Ontology Web Language or OWL [16] based on description logics [80], exist and provide a richer framework for knowledge representation and inferencing. However, RDF has the advantage that it is relatively simple to understand and use but much can practically be done with it. Its simplicity and usefulness make it likely more people will use it and it will spread more, better supporting widespread loosely-coupled collaborative data integration; i.e. RDF is arguably the "most bang for your buck" or "best value" semantic web technology, and has the best "simplicity versus expressiveness" tradeoff; i.e. RDF exemplifies the principle of least power discussed earlier [17]. This thesis thus focuses on practical uses of RDF for data interoperation.

## *3.5 YeastHub*

YeastHub demonstrates how to use the RDF approach to integrate heterogeneous genomic data, focusing on yeast data. YeastHub involves using a native RDF database system called Sesame [13] to implement a warehouse or hub for integrating diverse types of genomic/proteomic data. Sesame allows users to choose whether their RDF repository will be stored in main memory, in an underlying relational database (MySQL, http://www.mysql.com), or in native RDF files (based on DBM files). For small or moderate size datasets, the main memory approach provides the fastest query speed. For large amounts of data, Sesame utilizes the efficient data storage and indexing facilities provided by the relational database engine. Finally, the native file-based approach eliminates the need of using a database and its associated overhead at the cost of some performance if the data files involved are large. The YeastHub system consists of three key components: registration, data conversion, and data integration.

### 3.5.1 Registration

This component allows the user to register a web-accessible dataset so that it can be used by YeastHub. During the registration process, the user needs to enter information (metadata) describing the dataset such as the web location (URL), owner, and data type. The dataset description uses standard terminology from the Dublin Core metadata standard [81] and to encode the metadata in a standard format, the Rich Site Summary (RSS) format was used. RSS is a lightweight application of RDF, as the amount of metadata involved is typically small or moderate. The RSS-encoded description of an individual dataset is called an “RSS feed”. Many RSS-aware tools (e.g., RSS readers and aggregators) are available in the public domain, which allow automatic processing of

RSS feeds. Among the different versions of RSS, RSS 1.0 was chosen because it supports RDF Schema. This allows ontologies to be incorporated into the modeling and representation of metadata. Another advantage of using RSS 1.0 is that allows reuse of standard/existing modules as well as creation of new custom modules. The custom modules can be used to expand the RSS metadata structure and contents to meet specific user needs.

### 3.5.2 Data Conversion

As the registered datasets are provided by different sources in different formats, we need to convert these formats into the RDF format. A variety of technologies can be used to perform this data conversion. XSLT is used to convert XML datasets into the RDF format. For data stored in relational databases, D2RQ [12] is used to map the source relational structure to the target RDF structure. In addition, YeastHub provides a converter for translating tabular datasets into the RDF format. The translation operates on the assumption that each dataset belongs to a particular data type or class (e.g. gene, protein, or pathway) where each row represents a separate record (e.g. gene, protein, or pathway record), and one of the data columns is chosen by the user to hold the unique identifier for records. Each identifier identifies an RDF subject. The rest of the data columns become RDF properties of the subject. The user can choose to use the header column values as the default property names or enter his / her own property names. The system allows some basic filtering or transformation of string values (e.g., string substitution) when generating the property values. Once a dataset is converted into the RDF format, it can be loaded into the RDF repository for storage and queries. Also, it can be accessed by other applications through an API.

### 3.5.3 Data Integration

Once multiple datasets have been registered and loaded into YeastHub's RDF repository, integrated RDF queries can be composed to retrieve related data across the multiple datasets. YeastHub offers two kinds of query interface.

1. **Ad hoc queries.** This allows the user to textually compose RDF-based query statements and issue them directly to the data repository. Currently, it allows the user to use the following query languages: RQL, SeRQL, and RDQL. This requires the user to be familiar with at least one of these query syntaxes as well as the structure of the RDF datasets to be queried. SQL users typically find it easy to learn RDF query languages.

2. **Form-based queries**. While ad hoc RDF queries are flexible, users who do not know RDF query languages or who want simpler means of executing basic queries can use YeastHub's guided query builder interface to pose queries. YeastHub allows users to query the repository through web query forms, although they are not as flexible as the ad hoc query approach. To create a query form, YeastHub provides a query template generator. First, the user selects the datasets and the properties of interest. Second, the user needs to indicate which properties are to be used for the query output (select clause), the search Boolean criteria (where clause), and the join criteria (property values that link the records of the multiple datasets, e.g. gene identifier). In addition, the user is given the option to create a textfield, pulldown menu, or select list (in which multiple items can be selected) for each search property. Once all the

information has been entered, the user can go ahead to generate the query form by saving it with a name. The user can then use the generated query form to perform Boolean queries on the datasets associated with the form.

### 3.5.4 Example YeastHub Query

Fig. 2 shows an RDF query statement written in SeRQL (Sesame implementation of RQL), which simultaneously queries the following yeast resources: a) essential gene list obtained from MIPS, b) essential gene list obtained from YGDP, c) protein-protein interaction data [82], d) gene and GO ID association obtained from SGD, e) GO annotation and, f) gene expression data obtained from TRIPLES [24]. Datasets (a)- (d) are distributed in tab-delimited format. They were converted into our RDF format. The GO dataset is in an RDF-like XML format (we made some slight modification to it to make it RDF-compliant). TRIPLES is an Oracle database. We used D2RQ to dynamically map a subset of the gene expression data stored in TRIPLES to RDF format.

```
SELECT DISTINCT ns0orf,ns0connectivity,ns4accession,ns4name,ns5growth_condition,
ns5clone_id, ns5expression_level
FROM
{source58640} ns1:orf {ns1orf},
{source58639} ns2:orf {ns2orf},
{source58638} ns3:DB_Object_Synonym {ns3DB_Object_Synonym},
{source58638} ns3:GO_ID {ns3GO_ID},
{source58636} ns4:name {ns4name},
{source58636} ns4:accession {ns4accession},
{source55396} ns5:orf {ns5orf},
{source55396} ns5:growth_condition {ns5growth_condition},
{source55396} ns5:expression_level {ns5expression_level},
{source55396} ns5:clone_id {ns5clone_id},
{source58642} ns0:connectivity {ns0connectivity},
{source58642} ns0:orf {ns0orf}
WHERE
ns0connectivity="80"
AND ns5expression_level="1"^^<http://www.w3.org/2001/XMLSchema#longInteger>
AND ns5clone_id="V182B10"^^<http://www.w3.org/2001/XMLSchema#string>
AND ns5growth_condition="vegetative"^^<http://www.w3.org/2001/XMLSchema#string>
AND ns0orf=ns1orf
AND ns1orf=ns2orf
AND ns2orf=ns3DB_Object_Synonym
AND ns3DB_Object_Synonym=ns5orf
AND ns3GO_ID=ns4accession
USING NAMESPACE
ns2=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58639.rdf> ,
ns3=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58638.rdf> ,
ns1=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58640.rdf> ,
ns0=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58642.rdf> ,
ns5=<http://mcdb750.med.yale.edu/yeasthub/schema/schema_triples.rdf#> ,
ns4=<http://139.91.183.30:9090/RDF/VRP/Examples/schema_go.rdf>
```

| ns0orf | ns0connectivity | ns4accession | ns4name | ns5growth_condition | ns5clone_id | ns5expression_level |
|---|---|---|---|---|---|---|
| YBL092W | 80 | GO:0005842 | cytosolic large ribosomal subunit (sensu Eukaryota) | vegetative | V182B10 | 1 |
| YBL092W | 80 | GO:0003735 | structural constituent of ribosome | vegetative | V182B10 | 1 |
| YBL092W | 80 | GO:0006412 | protein biosynthesis | vegetative | V182B10 | 1 |

**Fig. 2. SeRQL query statement correlating between gene essentiality and connectivity.**

The example query demonstrates how to correlate between gene essentiality and connectivity derived from the interaction data. The hypothesis is that the higher the connectivity of a gene, the more likely that it is essential. This hypothesis has been investigated in other work [83, 84]. The example query includes the following Boolean condition: *connectivity = 80*, *expression_level = 1*, *growth_condition = vegetative*, and *clone_id = V182B10*. Such Boolean query joins across six resources based on common gene names and GO IDs. Fig. 2 (at bottom) shows the query output, which indicates that

the essential gene (YBL092W) has a connectivity equal to 80. This gene is found in both the MIPS and YGDP essential gene lists, thus giving a higher confidence of true essentiality (i.e. the two resources might have used different methods and sources to identify their essential genes, and their concordance could indicate higher likelihood of true essentiality). The query output displays GO annotation (molecular function, biological process, and cellular component) and TRIPLES gene expression.

# Chapter 4 LinkHub

An important problem encountered in building YeastHub was that, although we were able to co-integrate many disparate datasets, the integration was thin; the key to effective and flexible use of YeastHub was numerous and varied connections among the integrated datasets and these were limited in the original YeastHub system. This problem was an important motivation for the LinkHub system described in this chapter. Rather than integrate all types of data as YeastHub aims to do, LinkHub focuses on an important and more manageable high-level structuring principal for biological data, namely biological identifiers and the relationships (and types of relationships) among them. LinkHub is thus useful and complementary to YeastHub as a "connecting glue" among datasets in that it makes and stores these cross-references (i.e. performs ontology alignment of biological identifiers) and enables more complete integrated access to the YeastHub data.

LinkHub is a semantic-web RDF-based system that manages complex graphs of proteomics identifier relationships and allows exploration with web interactive and query interfaces. For efficiency and robustness, relational-database access and translation between the relational and RDF versions is also provided. LinkHub is practically useful in creating small, local hubs on common topics and then connecting these to major portals in a federated architecture; LinkHub has been used to establish such a relationship between UniProt and the North East Structural Genomics Consortium. LinkHub can thus help support loosely coupled, collaborative data integration without requiring explicit coordination or centralization. In its role as "connecting glue", LinkHub also facilitates queries and access to information spread across multiple databases between different identifier spaces. Example queries of the combined YeastHub and LinkHub are given

discovering “interologs” of yeast protein interactions in the worm and exploring the relationship between gene essentiality and pseudogene content, and also showing how “protein family based” retrieval of documents can be achieved. LinkHub is accessible by either http://hub.gersteinlab.org or http://hub.nesg.org.

## *4.1 Background*

A key abstraction or “scaffold” for representing biological data is the notion of unique identifiers for biological entities and relationships among them. For example, each protein sequence in the UniProt database is given a unique accession by the UniProt curators, e.g. Q60996; this accession uniquely identifies its associated protein sequence and can be used as a key to access its sequence record in UniProt. And UniProt sequence records contain cross-references to related information in other genomics databases, e.g. Q60996 is cross-linked in UniProt to Gene Ontology identifier GO:0005634 and Pfam identifier PF01603 (although the kinds of relationships, which would here be “functional annotation” and “family membership” respectively, are not specified in UniProt). Two identifiers such as Q60996 and GO:0005634 and the cross-reference between them can be viewed as a single edge between two nodes in a graph. Conceptually, then, a large, important part of biological knowledge can be viewed as a massive graph whose nodes are biological entities such as proteins, genes, etc. represented by identifiers and the links in the graph are typed and are the specific relationships among the biological entities. Figure 3 is a conceptual illustration of the graph of relationships among biological identifiers, with the boxes representing biological identifiers (originating database names given inside) and different edge types representing different kinds of relationships. The problem is that this graph of biological knowledge does not explicitly exist. Parts of it are

in existence piecemeal, e.g. UniProt's cross references to other databases, while other parts do not exist, e.g. the connections between structural genomics targets and UniProt identifiers.

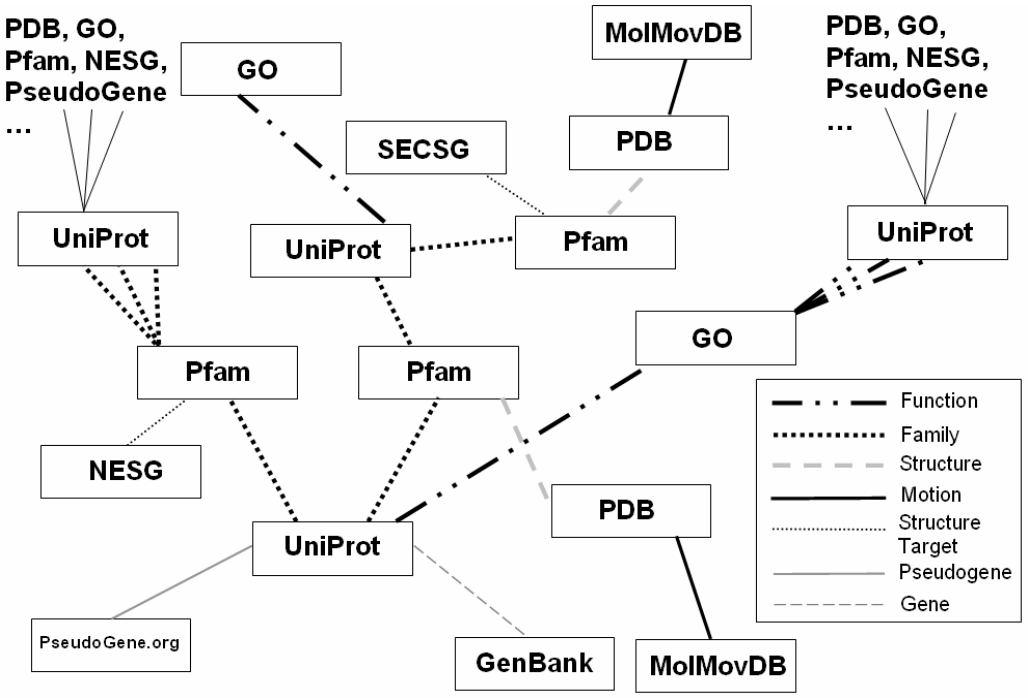



**Fig. 3. A conceptualization of the semantic graph of interrelationships among biological identifiers.**

A basic problem preventing this graph of relationships from being more fully realized is the problem of nomenclature. Often, there are many synonyms for the same underlying entity caused by people independently naming them for use in their own datasets or leftover common names that people loosely gave to biological entities early on before large-scale databases were created and which have managed to stick as names used by researchers. An example of this synonym problem is that there are many synonymous protein identifiers, as different laboratories (for example, the structural genomics centers) have assigned their own lab-specific identifiers to the proteins they are

working on. There can also be lexical variants of the same underlying identifier (e.g. GO:0008150 vs. GO0008150 vs. GO-8150). Synonyms are a small part of the overall problem, however, and more generally there are many kinds of relationships, some one-to-one but also one-to-many relationships are quite prevalent. For example, a single Gene Ontology identifier can be related with many UniProt identifiers (i.e. they all share the same functional annotation).

An important structuring principle for genes and proteins, which also leads to one to many relationships, is the notion of families based on evolution. A given protein or gene can be composed of one or more family units, called domains. For example the two Uniprot entities in figure 3 are both composed of two different Pfam domains. In general a given Pfam domain will be related to many UniProt proteins by this family type link, and these UniProt proteins can be related further still to other entities through various kinds of relationships (and similarly for GO). Thus, doing a transitive closure even a few levels deep in this relationship graph can lead to indirect relationships with a great number of other entities, and being able to store, manage, and work with this graph of entities and relationships can lead to many opportunities for interesting exploratory analysis. It is important to note, however, that there are certain relationship types for which following them in the wrong way can lead to incorrect inferences, with the family relationship being a key one. For example, starting from some protein in the graph you might reach another protein that shares a common family domain; if this other protein has additional different domains then it is generally not valid to propagate inferences made via these additional domains to the original protein. If you don't have a concept of relationship types and what they mean then you might just blindly assign features you

find in following links of relationships and this could potentially lead to serious inaccuracies in your analysis.

## *4.2 Implementation*

### 4.2.1 LinkHub: a system for loosely coupled, collaborative integration of proteomics identifier relationships

The semantic web is increasingly gaining traction as the key standards-based platform for biological data integration [75, 79, 85, 86]. LinkHub is designed based on a semantic graph model, which captures the graph of relationships among biological entities discussed above. LinkHub is thus a good fit to semantic web technologies, in particular RDF because RDF precisely models such graph data. To provide a scalable implementation while exploring the semantic web database technologies, LinkHub was implemented in both a MySQL (http://www.mysql.com) database and in a Resource Description Framework or RDF database. LinkHub provides interfaces to interact with this graph in various ways such as a web frontend for viewing and traversing the graph as a dynamic expandable / collapsible HTML list and a mechanism for viewing particular path types in the graph, as well as via RDF query languages.

Another motivation for LinkHub is that centralized data integration to an extent does make sense, e.g. a single lab or organization might want to interrelate its various resources to one another and to larger, well-known resources such as UniProt or GenBank, i.e. create a local central hub of interconnections among its individual data resources; but it does not want to have to explicitly connect its data resources up to everything in existence, which is impossible. The key idea is that if groups independently

maintaining data resources each connect their resources up to some other resource X, then any of them can reach any other through these connections to X, and we can collectively achieve incremental global integration of genomics data in this way. LinkHub is a software architecture and system which aims to help realize this goal by enabling one to create such local minor hubs of data interconnections and connecting them to major hubs of data such as UniProt or GenBank in a federated "hub of hubs" framework and this is illustrated in figure 4.

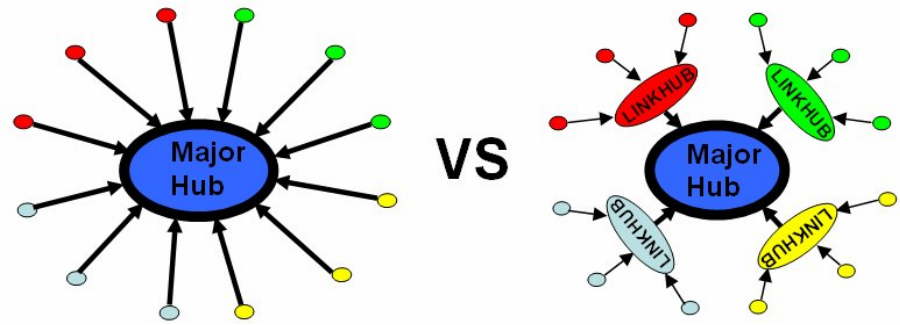



**Fig. 4. LinkHub as an enabler of an efficient "hub of hubs" organization of biological data.** The different colors represent different labs, organizations, or logical groupings of data resources.

### 4.2.2 Mapping Biological Identifiers and Obtaining LinkHub Data

As discussed in chapter 1, biology has a fundamental, commonly accepted principle around which data can be organized, namely biological sequences such as DNA, RNA, and protein, and various string matching techniques for biological sequences can solve a large part of the ontology alignment problem in biology. LinkHub thus takes advantage of biological sequence matching, in particular conservative, exact sequence matching, to cross-reference or align biological identifiers. LinkHub also takes advantage of available sources of pre-computed identifier mappings, with the most important one being UniProt which is arguably the most important major proteomics resource and serves as LinkHub's

backbone content (i.e. most relationships between identifiers in LinkHub are indirect through UniProt). The general strategy for mapping identifiers in LinkHub is to first take advantage of known and trusted pre-computed identifier mappings; if such pre-computed mappings are unavailable, an attempt is made to map identifiers based on exact sequence matches of their underlying sequences to UniProt and other sources of sequence data whose identifiers are stored in LinkHub.

Efficient, exact sequence matching programs were developed and used to do quick inter-database cross-referencing or alignment based on exact sequence matches (e.g. to cross-reference TargetDB to UniProt, see below). A custom Perl module was developed and used to index UniProt (and in general sequence databases in FASTA format [87]) to support this fast exact sequence matching. Specialized Perl web crawlers and other scripts were written to fetch and extract data from different sources in different formats; identifiers, identifier relationships, and other related information were extracted from the sources and inserted into the LinkHub MySQL database (which is also converted to RDF and inserted into the RDF version of LinkHub; see below). A running instantiation of the LinkHub system is at http://hub.gersteinlab.org and http://hub.nesg.org, and it is actively used and populated with data from the Gerstein Lab (http://www.gersteinlab.org) and related to the lab's research interests. Thus while the ideas of LinkHub are applicable in general to biological data, the concrete instantiation of LinkHub focuses heavily on proteomics data, as that is a key research initiative of the Gerstein Lab. The "hub of hubs" relationship described above has already been established between UniProt and LinkHub (i.e. UniProt hyperlinks to the LinkHub instantiation and cross-references to it in its DR lines). In addition, LinkHub cross-

reference the proteins which are targets of the structural genomics initiative (obtained from the TargetDB resource [47]) to UniProt and the LinkHub instantiation serves as a "related links" and "family viewer" (more below) gateway for the Northeast Structural Genomics Consortium (NESG) [45] with which the Gerstein Lab is affiliated. Additional focuses of the LinkHub instantiation are yeast resources, macromolecular motions [88], and pseudogenes [89].

### 4.2.3 LinkHub Database Models

LinkHub is conceptually based on the semantic web (graph) model and is thus represented and stored in Resource Description Framework or RDF [10]. However, experience using RDF database technology has found it to be currently lacking in performance and scalability [26]. In fact, this is likely an important impediment to more active and widespread use of the semantic web, and the creation of high-performance, robust RDF databases should be a research priority of the semantic web community. Thus, to support LinkHub's practical daily use, LinkHub is also modeled and stored using relational database technology (MySQL) for efficiency and robustness. RDF can be easily stored in relational databases, usually by having a "triples" table which stores the RDF graph's edges, and some RDF databases (e.g. 3Store [90]) use relational databases such as MySQL as their underlying data store. However, the drawback of storing RDF in relational databases through approaches such as the "triples" table is that it is not a natural or efficient representation of the graph data. In particular, it is difficult to use declarative relational queries (SQL) to perform certain types of graph operations such as recursive traversal of links and retrieval of sub-graphs; such graph operations could be impossible or at least very inefficient, e.g. requiring the "triples" table to be self-joined

many times. Specialized procedural codes are thus needed to implement such graph operations (such as were required for the "path type" view described below). Although relational databases such as Oracle support hierarchical queries, these query languages are vendor-specific, which are not compliant with the SQL standard. Thus, while ideally LinkHub could be stored and managed using only an RDF database, the lack of high-performance, robust RDF database technology required modeling and storing LinkHub also in the more tried and tested relational database technology for practical daily use.

The relational structure of LinkHub (see figure 5a) reflects how the graph of biological identifier relationships and associated data, such as URLs of identifier-specific web pages, are managed and stored. Biological identifiers are stored in the identifier table and are typed, where the identifier_types table gives the type. Thus, for example, two different identifiers in separate databases which happen to have the same identifier text can nevertheless be distinguished by differing identifier types (based on the databases they come from). The mappings table is used to store the relationships between identifiers, with the "type" attribute giving the description or meaning of the relationship. The identifier table thus gives the nodes and the mappings table the edges of the graph of biological identifier relationships. The resource, resource_accepts, and link_exceptions tables together manage and store URLs for identifier-specific web pages (e.g. the web page at UniProt giving specific information particular to some UniProt identifier). The basic idea is that web resources such as UniProt have template URLs which can be interpolated with particular identifiers to generate identifier-specific URLs. The resource table contains a short name, longer description, and the template URL of web resources such as UniProt. The resource_accepts table lists the particular types of identifiers that

can be interpolated into a resource's template URL, as well as an exception type except_type. The exception type is to handle cases where not all identifiers of an accepted type are legal, i.e. some of the identifiers cannot be interpolated into the template URL to generate a good URL. If except_type is NONE then there are no exceptions and all identifiers of the type are accepted. Otherwise except_type has value NACC or ACC. If except_type is NACC, then the exceptions are explicitly given in the link_exceptions table (i.e. the identifiers in the link_exceptions table of the given type for the resource are the ones that cannot be interpolated into the template URL, and all other identifiers of the type CAN be interpolated). If except_type is ACC then the behavior is the opposite: the identifiers NOT listed in the link_exceptions table are the exceptions and the ones explicitly listed are the only ones that can be interpolated into the resource's template URL. NACC and ACC exception types are both supported to allow the most efficient handling of exceptions, i.e. whichever is smaller between the set of accepted identifiers and the set of exception identifiers can be listed in link_exceptions thus minimizing the amount of space necessary for storing exceptions. The resource_group table supports grouping of web resources, e.g. all web resources maintained by the Gerstein Lab or relating to protein structure. Finally, the resource_attribute table allows free text attributes to be associated with web resource, however it is not currently used.

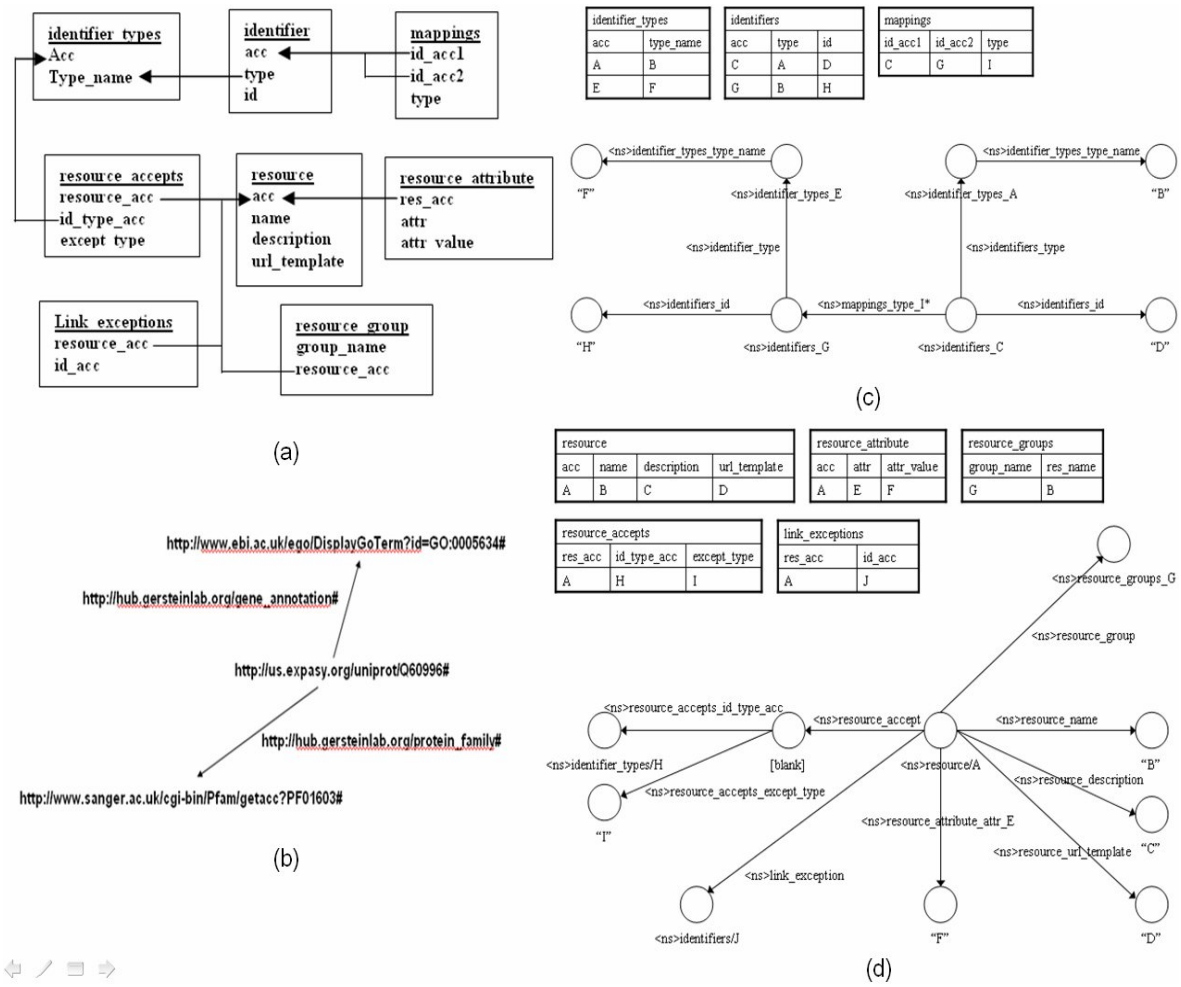



**Fig. 5. LinkHub Relational and RDF Data Models.** (a) LinkHub relational model (b) An example RDF graph (c) and (d) How the relational model maps to the RDF structure.

### 4.2.4 RDF Data Model

RDF is a popular data model (or ontological language) for the semantic web and is designed to provide a natural representation of a directed labeled graph. In addition, it comes with query languages (e.g., RDQL [91]) to allow the user to pose semantic queries against graph data. While there are more advanced ontological languages such as the Web Ontology Language or OWL [16] that support data reasoning based on Description Logics or DL (http://dl.kr.org/), RDF is a good start and much can be effectively modeled

with it. For example, the benefits of representing proteomics data in RDF were discussed in [75]. In addition, UniProt data has recently been made available in RDF format (http://www.isb-sib.ch/~ejain/rdf/data/).

RDF is based on the idea of identifying things using web identifiers called Uniform Resource Identifiers or URI's [11], and describing resources in terms of properties and property values. This enables RDF to represent simple statements (in the form of RDF triplets) about resources as a graph of nodes and arcs. Each statement is represented as an RDF triplet: (*subject*, *property*, *property value*), where *property value* can be a literal or a pointer to another *subject*. Such a collection of statements represent the resources, and their properties and values. Figure 5b gives an example RDF graph of two statements or triplets: (Q60996, gene_annotation, GO:0005634) and (Q60996, protein family, PF01603), which describe that there is a protein (Q60996) whose gene annotation is identified by GO: 0005634 and protein_family by PF01603. This also exemplifies using an RDF graph to connect multiple resources. Here, it connects UniProt, Gene Ontology, and Pfam. The detailed description associated with each identifier can be provided by the corresponding resource (the URL or URI can provide access to such detailed descriptions).

It is straightforward mapping between the relational and RDF versions of LinkHub and Java code is used to do this. Figures 5c and 5d illustrate how the relational tables are mapped to the corresponding RDF structure. 5c illustrates how the key LinkHub relational tables identifier_types, identifiers, and mappings (reproduced above the RDF structure) are mapped to the corresponding RDF structure. The resulting RDF graph captures different types of object identifiers stored in different databases and the

relations (or mappings) between these object identifiers. The mapping types are explicitly represented as RDF properties. 5d shows how the rest of the LinkHub relational tables (reproduced above the RDF structure) map to the RDF structure. The resulting RDF graph captures the different web resources (which can be grouped) accessible by LinkHub. In addition, the graph captures information about which web resources accept which types of object identifiers, as well as exceptions. Appendix 2 contains the RDF schema for the RDF structure of LinkHub.

### 4.2.5 LinkHub Web Interfaces

The primary interactive interface to the LinkHub database is a web-based interface (implemented using the so-called AJAX technologies, see http://en.wikipedia.org/wiki/AJAX, i.e. DHTML, JavaScript, DOM, CSS, etc.) which presents subsets of the graph of relationships in a dynamic expandable / collapsible list view. This interface allows viewing and exploring of the transitive closure of the relationships stemming from a given identifier interactively one layer at a time: direct edges from the given identifier are initially shown and the user may then selectively expand fringe nodes an additional layer at a time to explore further relationships (computing the full transitive closure is prohibitive, and could also cause the user to "drown" in the data, and we thus limit it initially, and in each subsequent expansion, to anything one edge away, with the user then guiding further extensions based on which relationships he would like to explore).

Figure 6 is a screenshot of the interface. Here, the data and relationships for UniProt identifier P26364 are presented. P26364 is presented at the root of the list, and lower levels contain information on additional related identifiers. Each identifier has two

subsections: Links which gives a list of hyperlinks to web documents directly relevant to the identifier; and Equivalent or Related Ids which contains a list of additional identifiers related to the first identifier (the relationship type if it exists is given in parentheses; a synonym relationship is assumed if no relationship is given). The identifiers in the Equivalent and Related Ids section may themselves be further related to other identifiers which will have their own Links and Equivalent or Related Ids sections, ad nauseum. The initial display shows the transitive closure of the root identifier one level deep, and dynamic callbacks to the server retrieve additional data when the user clicks on identifiers whose subsections have not yet been loaded; in this way, the user can explore the relationship paths he desires without performance penalties (of loading the whole graph) or 'information overload'. The interface is dynamic, and a '+' list icon can be expanded to view the hidden underlying content, and a '-' list icon can be clicked to hide the content.

The second interface presents results the same as the first interface (i.e. dynamic expandable / collapsible list view) but allows viewing of particular path types in the graph. For example, one might want to view all proteins in some database D in the same Pfam family as a given protein; in LinkHub Pfam relationships are stored for UniProt proteins, so one could view the fellow family members of the given protein by viewing all identifier relationship paths (starting from the given protein) matching:

> Given protein in database D → equivalent UniProt protein → Pfam family → UniProt proteins → other equivalent proteins in database D

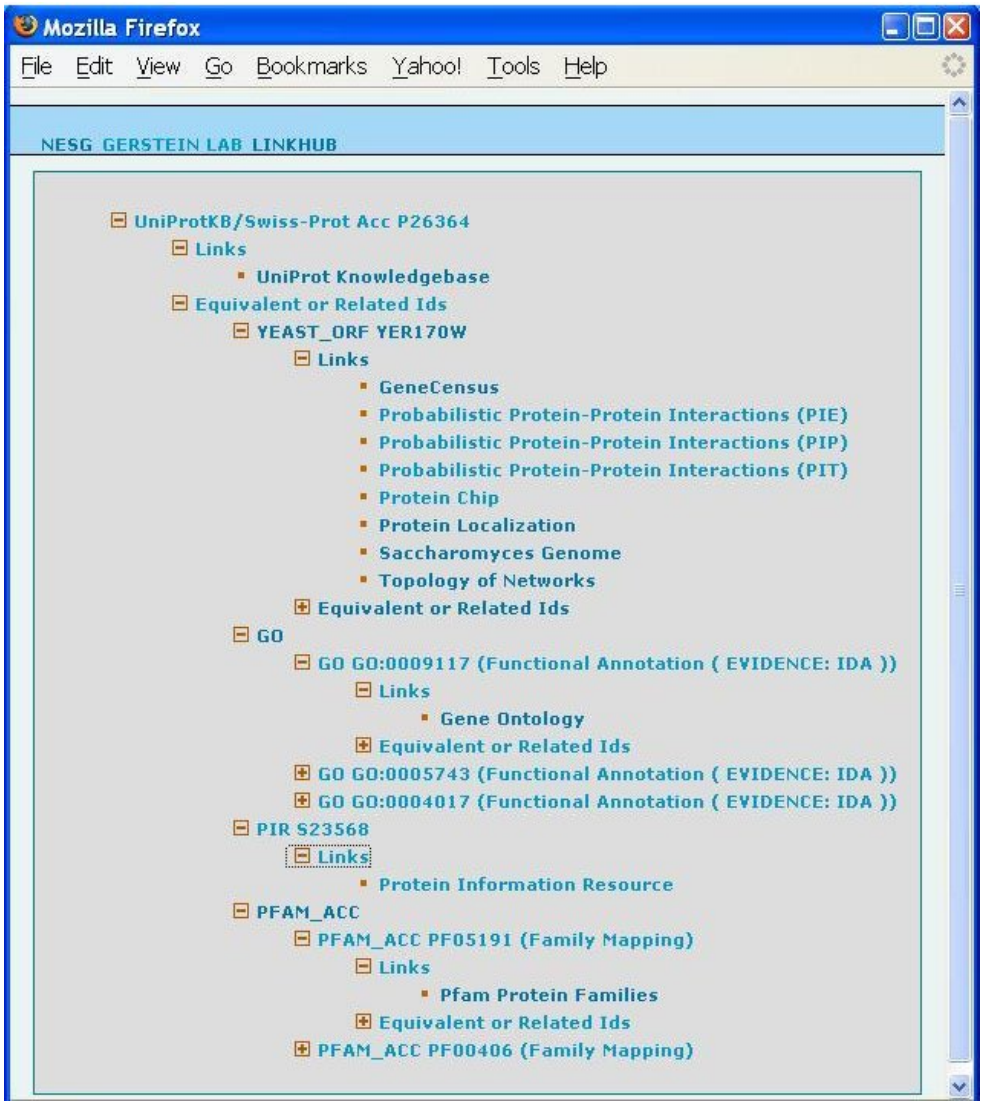



**Fig. 6.The basic DHTML list interface to LinkHub.**

## *4.3 Results*

### 4.3.1 Novel Information Retrieval based on LinkHub Relational Graph Structure

The "path type" interface to LinkHub allows one to flexibly retrieve useful subsets of the web documents linked to identifier nodes in the graph based on the relational structure of the graph. As discussed in chapter 1, normal search engines (which generally just rely on users entering keywords) could not provide such access, and thus LinkHub enables novel information retrieval to the web documents it knows about (i.e. linked to its identifier nodes). LinkHub has limited web document hyperlinks initially linked to its nodes, and if this could be increased the utility of this novel information retrieval would be enhanced. In fact, the next chapter shows how the information present in the LinkHub graph can be used to construct ranking functions for document relevance ranking, and this can be used to automatically retrieve documents from the web or the scientific literature (from PubMed) relevant to identifiers in the LinkHub graph.

An important and practical use of this “path type” interface is as a secondary, orthogonal interface to other biological databases in order to provide different views of their underlying data. For example, the molecular motions database MolMovDB [88] provides movie clips of likely 3D motions of proteins, and one can access it by PDB (http://www.pdb.org) identifiers. However, an alternative useful interface would be a “family view” where one queries with a PDB identifier and wants to see all available motion pages for proteins that are in the same family as the query PDB identifier, and LinkHub provides this interface for MolMovDB. LinkHub also provides a similar “family view” interface to structural genomics data, e.g. see the NESG’s SPINE [46]

target pages such as http://spine.nesg.org/target.pl?id=WR4 for the "NESG Family Viewer" links. One can easily imagine other similar applications, e.g. a "functional view" where all pages for proteins that have the same Gene Ontology function as a given protein are shown or a "pseudogene family view" where all pages for pseudogenes of proteins in the same family are shown. While the "path type" interface is a simple way of providing novel, relational access to LinkHub identifier node-linked documents, RDF query language access to the LinkHub relational graph would allow the most flexible novel information retrieval.

### 4.3.2 RDF Queries

To demonstrate the data interaction and exploration capabilities engendered by the RDF version of LinkHub, the RDF-formatted LinkHub dataset is loaded into the YeastHub system described above which uses Sesame [13] as the native RDF repository. Two demonstration queries below written in SeRQL (Sesame implementation of RQL) demonstrate one can efficiently do the kinds of interesting preliminary scientific investigation and exploratory analysis commonly done at the beginning of research initiatives (e.g. to see whether they are worth pursuing further). These queries make use of information present in both YeastHub and LinkHub (and thus could not be done without joining the two systems), and LinkHub is used as 'glue' to provide connections (both direct and indirect) between different genomics identifiers. It is noteworthy that these queries can be formulated and run in relatively very little time (a few hours at most) and they roughly duplicate some results from published papers. In effect, LinkHub does the up-front time-consuming manual work of integrating multiple datasets, and this integrated data is generally useful for efficient formulation and execution of queries,

which is in contrast to the papers which likely required extensive "one-off" effort to combine the necessary data to achieve their results.

**Query 1: Finding Worm 'Interologs' of Yeast Protein Interactions**

Proteins rarely act in isolation and often interact with one another and other molecules to perform necessary cellular actions. Experimental determinations of protein interactions are expensive and computational methods can leverage them for further interaction predictions. With this query we want to consider all the protein interactions in yeast (S. cervisiae) and see how many and which of them are present as evolutionarily related *homologs* in worm (C. elegans), also known as *interologs* [92], i.e. protein pairs in worm corresponding to evolutionarily related known interacting pairs in yeast. We thus start with a dataset containing known and predicted yeast protein interactions which is already loaded into YeastHub [93]; here the interactions are expressed between yeast gene names. Part of the SeRQL statement for this query together with a portion of its corresponding output can be seen in figure 7; appendix 3 contains the full text of the SeRQL query statement. However, abstractly, the query is doing the following. For each yeast gene name in the interaction set we can use LinkHub's data as 'glue' to determine its homologs (via Pfam) in worm by traversing paths in the LinkHub relationship graph of type:

yeast gene name → UniProt Accession → Pfam accession → UniProt Accession → WormBase ID.

Then, for each pair in the yeast protein interaction dataset, we determine if both of its yeast gene names lead to WormBase IDs [94] in this way and print those WormBase IDs as possible protein interactions if so.

**Query 2: Exploring Pseudogene Content versus Gene Essentiality in Yeast and Humans**

Pseudogenes are genomic DNA sequences similar to normal genes (and usually derived from them**)** but are not expressed into functional proteins; they are regarded as defunct relatives of functional genes [95, 96]. In the queries here we explore the relationship between gene essentiality (a measure of how important a gene is to survival of an organism) and the number of pseudogenes in an organism. We might hypothesize that more essential genes might have larger numbers of pseudogenes, and we explore this idea with queries of the joined YeastHub and LinkHub data. First, YeastHub has the MIPS [97] essential genes dataset, and we use this as our data on gene essentiality; LinkHub contains a small dataset of yeast pseudogenes [98].

Abstractly, for each yeast gene name in the list of essential genes we determine its pseudogenes by traversing paths in the relationship graph of type:

yeast gene name → UniProt Accession → yeast pseudogene.

For each essential yeast gene we then determine how many pseudogenes it has. We can then inspect the list of essential genes to see if there is a relationship between essentiality and number of pseudogenes. Humans have a large number of known pseudogenes [99]

but gene essentiality is difficult to characterize in humans (with many tissue types and developmental states complicating the issue). Since essentiality is well studied in yeast, one thing we can do is determine the human homologs of yeast essential genes, which would perhaps likely be "more important" in a survival sense, and examine them for patterns associated with essentiality. For each yeast gene name in the list of essential genes, we can find the homologous pseudogenes in human by traversing paths in the LinkHub relationship graph of type:

yeast gene name → UniProt Accession → Pfam accession → human UniProt Id → UniProt Accession → Pseudogene LSID.

Part of the SeRQL for the first query (for yeast pseudogenes) and results from both can be seen in figure 7 (appendix 3 contains the full text of the SeRQL query statements for both queries), and they show that few yeast essential genes are associated with pseudogenes whereas this is not the case with human. This may reflect the difference in processes of creation of the predominate numbers of yeast and human pseudogenes (duplication vs retrotransposition, see [95, 96]).

```
SELECT DISTINCT Yeast_Protein_1, Yeast_Protein_2, Worm_Protein_1, Worm_Protein_2
FROM
{ppi}     it:Protein1                {Yeast_Protein_1},
{lhYO1}   lh:identifiers_id          {Yeast_Protein_1},
{lhYO1}   lh:identifiers_type        {lhYOType},
{lhYO1}   lh:mappings_type_synonym   {lhUP1a},
{lhUP1a}  lh:identifiers_type        {lhUPType},
{lhUP1a}  lh:mappings_type_Family_Mapping {lhPFAM1},
{lhPFAM1} lh:identifiers_type        {lhPFType},
{lhPFAM1} lh:mappings_type_Family_Mapping {lhUP1b},
...
WHERE
Yeast_Protein_1 = "YAL005C" AND
Yeast_Protein_2 = "YLR310C" AND
YEAST_ORF   = "YEAST_ORF" AND
(UNIPROT_KB = "UniProtKB/Swiss-Prot Acc" OR
 UNIPROT_KB = "UniProtKB/TrEMBL Acc") AND
PFAM_ACC    = "PFAM_ACC" AND
WORMBASE    = "WORMBASE"
USING NAMESPACE
it=<http://yeasthub2.gersteinlab.org/yeasthub/schema/the_platinum_standard_for_ppi20060224234451_schema.rdf>,
lh=<http://yeasthub2.gersteinlab.org/yeasthub/datasets/manual_upload/linkhub_schema.rdf#>
```

(a)

| Yeast_Protein_1 | Yeast_Protein_2 | Worm_Protein_1 | Worm_Protein_2 |
|---|---|---|---|
| YAL005C | YLR310C | CE00103 | CE01784 |
| YAL005C | YLR310C | CE00103 | CE16278 |
| YAL005C | YLR310C | CE00103 | CE19874 |
| YAL005C | YLR310C | CE00103 | CE28290 |
| YAL005C | YLR310C | CE00103 | CE31570 |
| YAL005C | YLR310C | CE00103 | CE31571 |

(b)

```
SELECT DISTINCT Yeast_ORF, Pseudogene
FROM
{gene}     mips:ORF                  {Yeast_ORF},
{lhYO}     lh:identifiers_id         {Yeast_ORF},
{lhYO}     lh:identifiers_type       {lhYOType},
{lhYOType} lh:identifier_types_type_name {YEAST_ORF2},
{lhYO}     lh:mappings_type_synonym  {lhUP},
{lhUP}     lh:identifiers_type       {lhUPType},
{lhUPType} lh:identifier_types_type_name {UNIPROT_KB},
...
WHERE
YEAST_ORF2   = "YEAST_ORF" AND
(UNIPROT_KB = "UniProtKB/Swiss-Prot Acc" OR
 UNIPROT_KB = "UniProtKB/TrEMBL Acc") AND
YEAST_PGENE  = "YEAST_PGENE"
USING NAMESPACE
mips=<http://yeasthub2.gersteinlab.org/yeasthub/schema/mips_lethal_genes20050608191535_schema.rdf>,
lh=<http://yeasthub2.gersteinlab.org/yeasthub/datasets/manual_upload/linkhub_schema.rdf#>
```

(c)

| Yeast_ORF | Pseudogene |
|---|---|
| YDR037W | 448_chrii |

(d)

| Yeast_ORF | Human_gene | Pseudogene |
|---|---|---|
| YAL003W | EF1B_HUMAN | urn:lsid:pseudogene.org:9606.Pseudogene:72051 |
| YAL003W | EF1B_HUMAN | urn:lsid:pseudogene.org:9606.Pseudogene:72052 |
| YAL003W | EF1B_HUMAN | urn:lsid:pseudogene.org:9606.Pseudogene:72053 |
| YAL003W | EF1B_HUMAN | urn:lsid:pseudogene.org:9606.Pseudogene:1934 |
| YAL003W | EF1B_HUMAN | urn:lsid:pseudogene.org:9606.Pseudogene:54119 |

(e)

**Fig. 7. Example RDF queries.** (a) shows a part of the SeRQL query that finds pairs of worm (C. elegans) proteins homologous to pairs of interacting proteins in yeast (C. cervisiae), i.e. “interologs”. b) shows part of the corresponding query results. (c) shows part of the SeRQL query that explores the relationship between gene essentiality and the level of pseudogene content in yeast, which is one feature that might be hypothesized to be associated with essentiality, with queries of the joined YeastHub and LinkHub data. (d) shows the yeast pseudogenes found, interestingly only one. (e) shows part of the list of pseudogenes found in human homologs for a similar query; the full list is long, around 20000, consistent with there being many known pseudogenes in humans.

# Chapter 5 Automated Information Retrieval for Biological Identifier-related Documents using LinkHub Subgraphs

This chapter describes how the information present in the LinkHub relational graph can be used for enhanced automated information retrieval access to documents from the web or the scientific literature (e.g. Medline) not explicitly linked to identifier nodes in the LinkHub graph. For example, an interesting and practical problem would be to retrieve documents highly relevant to a UniProt identifier from the web or Medline. The key idea is that the LinkHub subgraph emanating from a given identifier and the web pages (hyperlinks) linked to the identifier nodes in the subgraph provide copious and detailed information about the given central identifier that can be used to perform precise and accurate relevant document retrieval for it. The web pages linked to the identifier nodes in the subgraph are considered to be a "gold standard" for what the additional relevant documents should be like, and they are used as training sets to construct a ranking function used to score and rank additional documents (obtained from the web or scientific literature) for how well they match the training set. This chapter will first describe the basic procedure for using the LinkHub relational graph to perform automated information retrieval for identifier-related documents. Example uses of the procedure to retrieve web documents and scientific literature articles (from PubMed) relevant to a UniProt identifier will then be given to demonstrate the procedure and show that it gives reasonable results. Then, empirical results of the performance of the procedure for a curated bibliography of yeast protein-related documents will be given, followed by a discussion.

## *5.1 Deficiencies of Standard Search Engines*

The simplest approach to find relevant documents for a biological identifier would be to simply use a search engine and do a search using the identifier itself as the search term, however this likely will not give good results. First, the identifier text might be used in many different contexts (e.g. product identifier in a shopping catalog, or generally some identifier in another non-biological setting); thus the result set from the search engine will be a conflation of all these contexts when what is really wanted is to limit it to only documents of the correct, biological context (i.e. for which occurrences of the identifier text in the page actually refer to the searched for identifier). Second, many relevant documents might not refer to the underlying biological entity (e.g. protein or gene) directly by the given identifier, but will use other synonyms to refer to it -- the LinkHub graph should contain these synonyms and this additional information should be made use of to expand the search. Finally, and most importantly, many important relevant documents might not directly refer to the given identifier (or its synonyms) at all; for example, an identifier might represent a key protein in an important cancer pathway and we would thus like to pull in additional documents about cancer and cancer pathways that do not necessarily directly refer to the given identifier. Note that abstracts searchable at PubMed (the primary access point to the biomedical scientific literature) do not usually contain identifiers, so searching PubMed by related concepts is necessary. Even if identifiers are referred to, if we have information about closely related terms (as we do in LinkHub) we should not waste this information.

## 5.2 Traversing the Graph to get Weights

Define a *relational subgraph* (or just *subgraph*) of a node in the LinkHub graph to be all nodes reachable from that node by traversing relationship links from that node (i.e. the node's transitive closure of relationships); similarly, define the *N-subgraph* of a node to be the subset of its relational subgraph where all nodes in the subset are reachable by traversing at most N relationship links. The key idea is that the LinkHub subgraph for a given identifier and the web pages (hyperlinks) linked to the identifier nodes in the subgraph is concrete, accurate, extra information about the given identifier that can be used to improve document retrieval for the given central identifier. The documents linked to the identifier nodes in the subgraph are considered to be a "gold standard" for what the additional relevant documents should be like, and are used as a training set to construct a ranking function (in the form of a combined word weight vector of all the subgraph documents; see below) used to score and rank additional documents for how well they match the training set.

However, we do not consider all the web pages linked to identifier nodes in the subgraph as equally important. First, the web pages linked directly to the central identifier's node are given the highest importance (weight 1.0), while web pages linked to identifiers' nodes further from the central identifier are scaled down in importance based on how many relationship links away they are and the types of those relationship links (there is no downscaling for synonym links, and so web pages linked to synonym nodes are also given the highest weight 1.0). We give numerical weights to particular relationship types or related identifiers of particular types. These weights are a kind of "degree of relationship" and are meant to indicate how much the pointed to identifier type

expresses or elucidates the semantics of the identifier type pointing to it. For example in the examples of sections 5.5 and 5.6 weights of 0.75 and 0.5 are assigned respectively to identifier relationships of types "UniProt → PFAM" and "UniProt → GO". Thus, if the central identifier of interest were a UniProt protein, its pointed to PFAM and GO pages would be considered 75% and 50% as important respectively in expressing what the UniProt protein is about compared to the pages linked directly to the UniProt protein node (and synonym nodes).

An example makes this clear and is shown in figure 8. The weight of a node is determined by summing all the weights of incoming nodes, each multiplied by the downscaling parameter of its relationship type. This weight calculation starts at the originating node (A in fig. 8) and propagates out to its connected nodes, then their connected nodes, etc. This figure represents a worked out example, and the final weights of the nodes are written below them. Here, let A be the identifier about which additional relevant documents are desired. Edges without number labels are considered synonym links and they do not incur downscaling (they are implicitly labeled with 1.0); the other number labeled edges represent non-synonym relationships for which downscaling is employed. The web pages linked to A are most important and given weight of 1.0. B is a synonym for A and so its linked web pages are not downscaled and are given full weight. Identifier E is not a synonym and its linked web pages are scaled down by 0.5, giving them an importance weighting of 0.5. Identifier C has two links coming in from B and E, and neither of the links is a synonym; so the weights at B and E are downscaled by .5 and 0.7 respectively and then added to give the weight for C, i.e. 0.7*0.5 + 0.5 = 0.85. Identifier F is a synonym for E and shares its weight of 0.5; identifier G downscales F's

weight by 0.8 to give it a weight of 0.4. Finally, identifier D downscales C's weight by 0.33 and adds in its synonym F's weight to give a weight for D of 0.33*0.85 + 0.5 = 0.78. LinkHub has default scaling parameters for its relationship link types such as "family mapping" and "functional annotation" which can also be set as parameters.

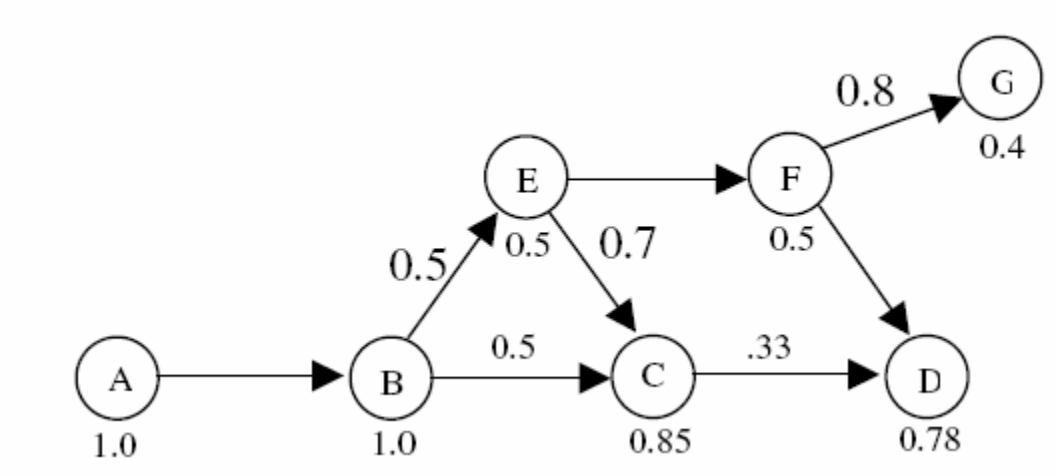



**Fig. 8. Computing importance weights for nodes (and their associated documents) in a LinkHub relational subgraph.**

## *5.3 Basic Information Retrieval and Text Categorization*

The procedure for document relevance ranking described below uses basic techniques from information retrieval [100-102] and text categorization [103-105] and this section gives an overview of these. Information retrieval and text categorization solve different, but related, problems and use similar basic techniques. Essentially, documents and queries are modeled the same in each, and the problem of text categorization is to predict which among a discrete number of categories a query document is a member of (e.g. is a query news article about sports, business, or the weather?) while the problem of information retrieval is, given a query (e.g. a list of words like in a web search engine or

even a whole document), score other documents for how well they match the query and rank them in decreasing order of relevance to the query. In information retrieval and text categorization documents are generally considered to be "bags of words", i.e. the ordering of words and relationships among them, and the syntactic structure and semantics of documents, are not considered but rather it is simply the presence or absence of words that is important. While this might seem to be too simplistic and ignoring too much important information, in fact it works well in practice. Concretely then, documents are represented as vectors of word weights, where the weights are meant to be proportional to how important the corresponding words are to discriminating the fine-grain meaning of the document. This way of representing documents is called the **vector space model**. For example, in a document about how some protein catalyzes a reaction the following might be part of the vector of word weights for the document:

<*protein* → 5.0, *reaction* → 4.0, *catalyze* → 4.0, *the* → 0.0, *a* → 0.0,
*science* → 1.5, *biology* → 1.0 …>

Note that what are intuitively the most discriminating terms, *protein*, *reaction*, and *catalyze*, are given the highest weights; the more general but still relevant terms *science* and *biology* are given non-negligible but considerably less weight. Finally, terms such as *a* and *the*, which are general terms appearing in practically all documents, provide no discriminating power and are given no weight. In fact, such general, non-discriminating terms are filtered out altogether before weighting is even done; such words are called **stopwords** and a list of such terms to filter is called a **stoplist** and usually consists of

around 100 of the most frequently occurring words.

To generate such a word weight vector from a raw document, a number of steps are undertaken. First, if the raw document is not in hand but rather just a hyperlink pointing to the document, the raw document text must be fetched over the internet using a web client application. Once the raw document text has been obtained it can be processed. If the document is not simply raw English text, in particular if it has markup such as HTML or XML, then the document must be filtered to raw English text; for example, XML and HTML tags need to be stripped (and comment and script sections removed). Rather than simply stripping the markup, the markup could be made use of for weighting words; for example, words appearing inside the <title> html tag might be considered more prominent and important and up-weighted, although there are no hard and fast rules for how to modify word weights based on markup and location in the document (it is application specific and somewhat ad-hoc). This thesis does not attempt to use such markup or location information to modify word weightings. Once the document is filtered to raw English text, the document must be **tokenized**, i.e. broken up into the individual terms or words that make up the document. This might seem straightforward, i.e. simply break text on whitespace (i.e. all sequences of space, tab, and newline characters), and this in fact works correctly in most cases. However, in addition to whitespace, there are other characters which are sometimes token delimiters and sometimes not, such as periods, colons, question marks, etc. and these complicate matters; also, how to handle numbers and numbers appended or pre-pended to words? In addition to issues relating to what characters constitute token delimiters (and in what context), more advanced lexical analysis can be used to attempt to extract meaningful

multi-token elements such as noun phrases, proper names, etc. Depending on the needs of the application, one can choose to use very simple or very complex tokenization procedures. In general, more complex tokenization procedures are more appropriate to fine-grain information extraction applications (such as attempting to extract gene and protein names from text) while simpler tokenization procedures suffice for applications where documents are considered only to be "bags of words" such as information retrieval and text categorization. Thus, this thesis uses a fairly basic tokenization procedure which in Perl code is as follows:

```
sub tokenize {

  my @tokens;
  while ($_[0] =~ /([-\w]+)/g) {
    my $word = lc $1;
    next unless $word =~ /[a-z]/;
    $word =~ s/^[^a-z]+//;  # Trim leading non-alpha characters (helps with ordinals)
    push @tokens, $word;
  }
  return \@tokens;
}
```

Note that all words are lowercased; word capitalization effectively does not matter and in fact can only hurt, e.g. two words which are really the same might not be so recognized because of different capitalization. This tokenization procedure and some other code for basic information retrieval and text categorization tasks used in the thesis was taken from the `AI::Categorizer` Perl module (http://search.cpan.org/~kwilliams/AI-Categorizer-0.07/).

After tokenization, the next step is called **stemming** or **lemmatization**, but it is an optional step. The basic idea is to transform words to their common base forms, for example *runs*, *running*, *runner*, and *run* would all be transformed to the common stem form *run*. The idea is that all these terms really signify the same, underlying concept and this should be reflected in the vector of word/term weights --- i.e. a document that

contained these terms is really referring to the concept *run* with high frequency and not to many different terms each with low frequency, and thus the word weight vector should have a single large weight for *run* and not several small separate weights for *runs*, *running*, *runner*, and *run*. The effect of stemming, then, is to reduce the number of distinct words in a document and to increase the frequency of occurrence of some words (i.e. words that would be considered separate without stemming, but collapse to a single word with stemming). While stemming procedures are not perfect (i.e. some words will not be transformed to their correct base form, or will be stemmed when they shouldn't be or not stemmed when they should be) they work reasonably well and it is generally accepted that stemming can provide a small positive benefit to information retrieval and text categorization applications, since the weights of words are based partly on their frequency in the document (see below), and this thesis thus uses stemming. In particular, a standard stemming algorithm that is used in many applications is the so called **Porter Stemmer** [106] (also see http://www.tartarus.org/martin/PorterStemmer/), and this thesis uses a Perl implementation of the Porter Stemmer in the `Lingua::Stem` module (http://search.cpan.org/dist/Lingua-Stem/). Note that stopwords can be removed either before or after stemming (if after, the stopwords need to be stemmed too); this thesis removes stopwords after stemming (although unlikely to make a big difference either way, this is the default behavior in the `AI::Categorizer` Perl module and seemed to produce the best results in most cases the author of that module tried, so it is adopted also for this thesis).

Finally, after tokenization and stemming we have the list of terms that are present in a document and we need to weight these terms to create the document's word/term

weight vector. The basic technique for doing this is called **Term Frequency-Inverse Document Frequency** or simply its acronym **TF-IDF**. The intuition is that, first, a word that occurs frequently in a document is more likely to be central to the meaning of the document, i.e. be relevant to what the document is about as a whole, and should thus be up-weighted. Second, the less frequent a word is in the corpus or collection of documents being queried the more discriminating that word is and that word should thus be up-weighted; similarly, the more frequent a word is in the corpus the less discriminating it is and it should thus be down-weighted (the extreme examples of this are in fact the stopwords, which occur in all or a large majority of documents; they have no discriminative value and are thus given weight 0, i.e. filtered out). In other words, the importance of a word increases proportionally to the number of times that word appears in the document but is offset by how common the word is in all of the documents in the corpus from which documents are being retrieved.

There are different ways of computing term frequency-inverse document frequency that are consistent with the above intuition. The following are some common ways of computing term frequency *TF* for a term T:

- Raw term frequency, i.e. a simple count of the number of times the term occurs in the document; call this *RTF*.
- $\sqrt{RTF}$
- Log(*RTF*)
- Binary: 1 if term present, 0 otherwise.

- Normalized: $\frac{n_i}{\sum_k n_k}$ where $n_i$ is the number of occurrences of the term, and the sum in the denominator is the total number of terms in the document.

The following are the common ways of computing inverse document frequency *IDF* for term T:

- Standard. Let *N* be the total number of documents in the corpus and *D* be the document frequency of T in the corpus (i.e. the number of documents in the corpus in which T appears at least once). Then, the standard inverse document frequency for T is $\text{Log}\left(\frac{N}{D}\right)$
- Probabilistic. Let N and D be the same as for standard. Then, the probabilistic inverse document frequency for T is $\text{Log}\left(\frac{N-D}{D}\right)$

Finally, once the ways to compute *TF* and *IDF* are decided, the weight for term T is then simply: $\text{weight}(T) = TF * IDF$. In this thesis, raw term frequency and standard inverse document frequency are used; for more details on TF-IDF see the `AI::Categorizer` Perl module documentation and Salton et al 1987.

Finally, since documents are modeled as vectors a natural way of computing similarity between such word weight vectors is by the cosine angle between them. The smaller the angle, the more similar two word weight vectors (and hence their original documents) are. This way of measuring similarity between documents is called the **cosine**

**similarity measure** or **cosim** for short, and for word weight vectors A and B is defined as:

$$\text{Cosim}(A,B) = \frac{A \bullet B}{|A| \quad |B|}$$

i.e. the dot product of the vectors divided by the product of their magnitudes. Note also that the cosim implicitly normalizes the vectors, as it is equivalent to simply the dot product of the normalized A and B. A value of 1 means the angle is 0 and hence signifies the maximum similarity; similarly, a value of 0 signifies the maximum angle and the least similarity. Information retrieval, then, uses the cosim to measure the similarity between a query (either an entire document or a list of terms entered by a user) and documents in a corpus being searched --- the result is a list of the corpus documents sorted descending by cosim value. In fact, this is essentially how some of the early web search engines worked, but it did not work particularly well because a huge number of documents will have about the same similarity value when the query consists of only a few terms; in fact, it was Google's solution to this problem [41] of improved relevance ranking by using the link structure of documents as "votes" of importance which was so successful and made them the most used search engine. The vector space model, TF-IDF term weighting, and cosim generally work well for ranking relevance in small document collections, or for larger collections (like the web) when the query is large and precise; in fact, the queries considered in this thesis are large and precise (multiple, whole documents added together; see chapter 5) and the ranking works well as will be shown below.

## *5.4 Building a Combined Word Weight Vector for Document Relevance Ranking*

The techniques described above such as tokenization, stopword filtering, stemming, term weighting, and cosine similarity measure are used in the procedure. We first extract the N-subgraph of the identifier which is the object of the query from the LinkHub graph for some value of N (e.g. 1 is used in the examples in proceeding sections below). We then obtain all documents linked to identifier nodes in this N-subgraph and consider them weighted by their node's weight (which was determined as described in section 5.2 above) --- this using and weighting multiple documents based on their relationships is a novel aspect of the procedure. We then turn each of these documents into word weight vectors as described above, except at the end we multiply all the term weights by the weight of their originating document as a whole (gotten from their node's weight). Next, all of these word weight vectors are added together to form the combined word weight vector. The combined word weight vector is then sorted by term weight and some percentage of the smallest weighted terms are eliminated (they presumably do not significantly add to discriminatory power or are effectively uninformative, noise terms and can thus be eliminated); for example, the bottom weighted 80% of terms can be eliminated and the top weighted 20% kept. To compute relevance for a new document we use the standard cosine similarity measure between the word weight vector representation of the new document and the N-subgraph documents' combined word weight vector.

Finally, it is necessary to obtain a set of documents that are potentially relevant and related to the given central identifier which is the object of the query (i.e. something to actually score and rank with the combined word weight vector). We only need to

consider documents which have terms in common with the combined word weight vector and a basic search engine can be used essentially as a keyword-to-document hash (implemented in the internals of the search engine as a so called inverted index [107]) to efficiently obtain such documents. Multiple searches are performed using as search terms all the identifiers in the N-subgraph, as well as some percentage of the top weighted terms from the combined word weight vector (translated to their unstemmed, original form), and the top results for each search are retrieved (e.g. top 50 or 100). Finally, all the result sets of all the searches are combined and we compute the cosine similarity value between the combined word weight vector and each element of this combined result set and rank (sort) descending based on the cosim value.

The following then is a restatement giving the detailed steps of the procedure for retrieving documents relevant to an identifier whose node is present in the LinkHub relational graph:

1. Extract the identifier's N-subgraph for some value of N. Larger values of N can potentially provide greater information, but past 2 could be negligible or too indirect and unreliable.
2. Assign the identifier's node (and synonym nodes) weight 1.0 and then propagate weights throughout the rest of the N-subgraph's nodes by downscaling based on relationship types and adding all in-link weights at each node (as explained above in section 5.2). Thus, each node in the N-subgraph obtains a weight in this way.
3. Determine all the documents linked to identifier nodes in the N-subgraph and assign them the same weight as the node they are linked to.

4. Form a word weight vector for each identifier node-linked document as described in section 5.3 scaled (multiplied) by their document's weight and add all of these to form the combined word weight vector. Eliminate some percentage of the smallest weighted terms. The combined word weight vector is used for ranking documents by the cosine similarity measure.
5. Use some percentage of the top-weighted terms (in their original, unstemmed form) from the combined word weight vector, along with all the identifiers of nodes in the N-subgraph, as "base searches", i.e. perform individual searches for each of them against the desired document collection (e.g. the web or PubMed). Combine the results of all the base searches together in one long list and score each document in this list for its cosine similarity value against the combined word weight vector; order descending by cosine similarity values. The end result is the ranked list of documents, where the higher a document is ranked the more specific and relevant it should be to the identifier which was the object of the query.

Essentially, this procedure can be viewed as systematically exploring the "concept space" around a given biological identifier and the display of results can present this as follows. For each ranked document, all the base searches which independently retrieved it can be listed for it ordered by relevance as determined by the search engine. The base searches that retrieved on average the most relevant documents can also be presented separately, ordered by average cosine similarity value of all documents they returned. These base searches returning on average the most relevant documents are key concepts

related to the given identifier and are of interest in their own right as succinct snippets of what the identifier is "about"; they could be called "semantic signatures". See figure 10 and appendix 4 for examples of this way of displaying results. The next two sections give examples of this procedure, first against the web and then against the biomedical scientific literature, to make it more clear and demonstrate that it gives reasonable, good results.

## *5.5 Improved Search for Web Documents Related to a Proteomics Identifier*

The basic procedure presented in section 5.4 above was implemented to retrieve documents from the web relevant to a proteomics identifier. The Yahoo search engine, via its web API, is used to perform the base searches to retrieve the documents to rank. Other search engines, such as Google, could also be used via their web API's but Yahoo was chosen because of its generous daily search limits, flexibility, and ease-of-use (Yahoo is also generally respected as a search engine and returns good results). As an example, figure 9 gives the LinkHub view for UniProt P26356, which shows its 1-subgraph and identifier node-linked documents (and their weights) which together form the training set; figure 10 shows the results of this enhanced web search for it. P26364 is a yeast mitochondrial protein named 'Adenylate kinase 2'. At top left in figure 10 are small screenshots of the web pages of the results and the spawned base searches that retrieved them (sorted descending by the average cosine similarity value of the documents they retrieved), reproduced in tabular format at bottom and right respectively for clarity. The two center columns in the table at bottom in figure 10 compare the overall rank given by the combined word weight vector with Yahoo's rank (out of 40 and for the

search term given in the rightmost column); the leftmost column is the page title as returned by Yahoo. While some of the results are more generally relevant, all of them are clearly relevant (i.e. documents of a biological nature and in the right sub-area of yeast genomics and proteomics, not unrelated documents of conflated senses of the identifier such as product catalog codes, etc.), and some, such as the ones with 'ADK' and 'ADK_lid' in their titles, are very closely related and relevant. Considering the top 20 spawning searches, again, while some are more generally relevant all are clearly biologically relevant and many of them, such as "kinase", "mitochondrial", "adenylate", "adk2", "saccharomyces", and the searches which are biological identifiers (PFAM, GO, etc.) are very closely related and relevant and give a very succinct overview of key aspects of the protein.

Interestingly, the direct Yahoo search using the UniProt identifier itself, i.e. P26364, was one of the lowest ranking searches (not even shown in the figure). In a manual inspection of the Yahoo search results for P26364 (on 2/27/2006), almost half (17 / 40) of the first 40 results clearly had absolutely nothing to do with the yeast protein P26364 but were related to other non-biological senses of the text P26364. In contrast, in the LinkHub-derived results for P26356 all of the first 40 were clearly related and the first unrelated result isn't found until position 72. Thus, it seems safe to conclude that the LinkHub-derived results for P26364 are superior to the Yahoo results.

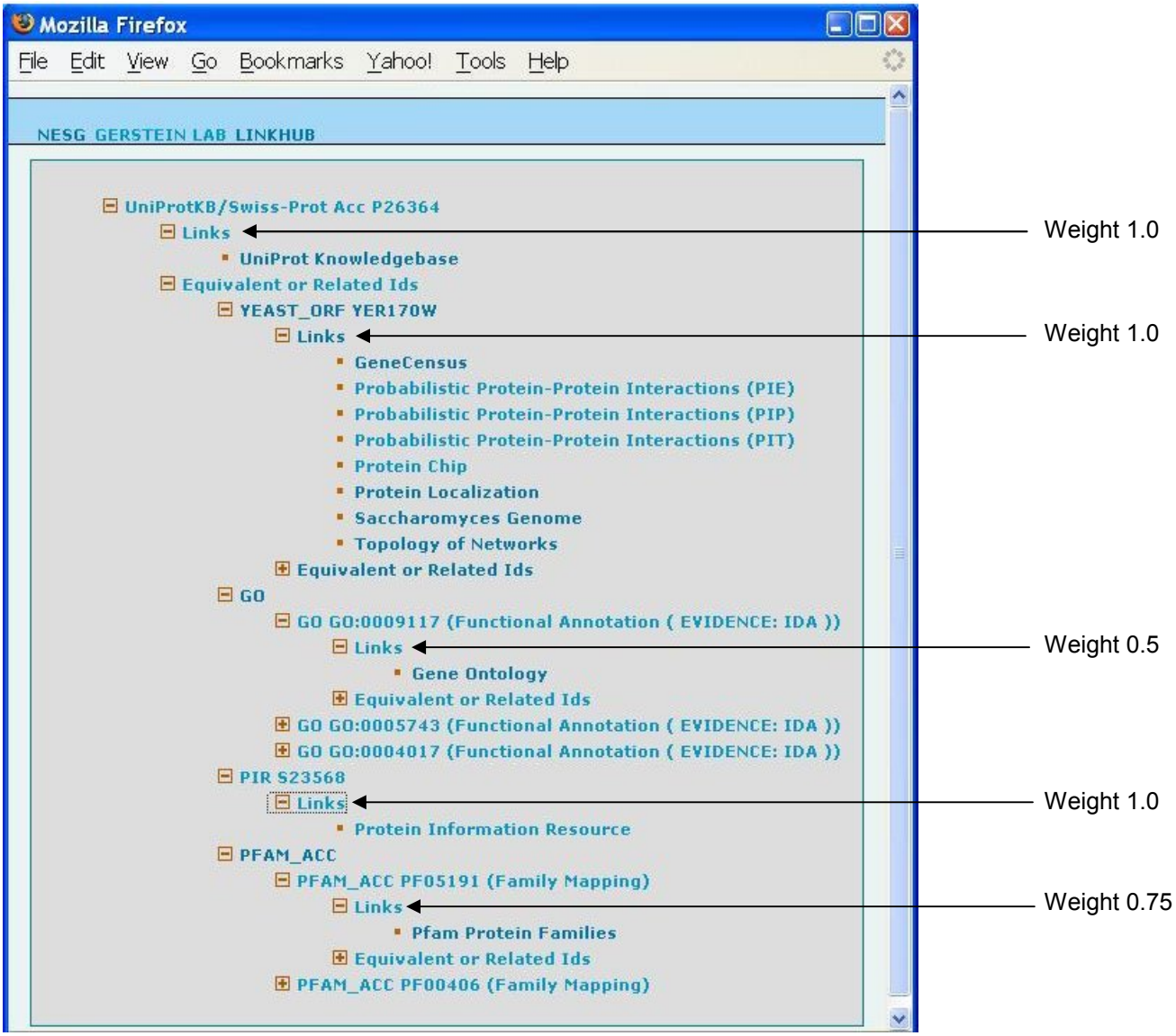
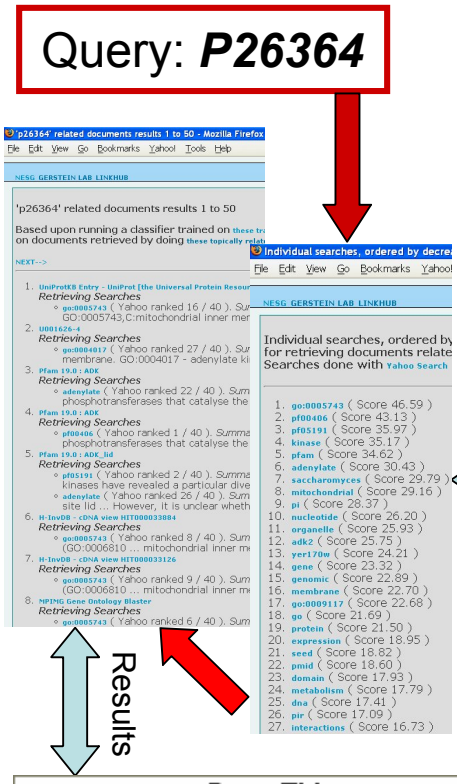



**Fig. 9. The LinkHub web interface view for UniProt protein P26364 (same as in figure 6) with arrows pointing to the identifier node-linked documents (hyperlinks) and giving their weights.**




| Search Term | Avg Norm Relevance Score |
|---|---|
| go:0005743 | 46.59 |
| pf00406 | 43.13 |
| pf05191 | 35.97 |
| kinase | 35.17 |
| pfam | 34.62 |
| adenylate | 30.43 |
| saccharomyces | 29.79 |
| mitochondrial | 29.16 |
| pi | 28.37 |
| nucleotide | 26.2 |
| organelle | 25.93 |
| adk2 | 25.75 |
| yer170w | 24.21 |
| gene | 23.32 |
| genomic | 22.89 |
| membrane | 22.7 |
| go:0009117 | 22.68 |
| go | 21.69 |
| protein | 21.5 |

| Page Title | Classifier Rank | Yahoo Rank | Retrieving Search Terms |
|---|---|---|---|
| UniProtKB Entry - UniProt [the Universal... | 1 | 16 | go:0005743 |
| U001626-4 | 2 | 27 | go:0004017 |
| Pfam 19.0 : ADK | 3 | 22 | adenylate |
| Pfam 19.0 : ADK | 4 | 1 | pf00406 |
| Pfam 19.0 : ADK_lid | 5 | 2 | pf05191 |
| Pfam 19.0 : ADK_lid | 5 | 26 | adenylate |
| H-InvDB - cDNA view HIT000033884 | 6 | 8 | go:0005743 |
| H-InvDB - cDNA view HIT000033126 | 7 | 9 | go:0005743 |
| MPIMG Gene Ontology Blaster | 8 | 6 | go:0005743 |
| Detailed annotation info for ACL00000176 | 9 | 7 | pf05191 |
| Detailed annotation info for ACL00000176 | 9 | 21 | pf00406 |
| Detailed annotation info for ACL00000176 | 9 | 33 | go:0004017 |
| Entry Page | 10 | 32 | go:0009117 |
| http://www.charite.de/ch/medgen/ontolo... | 11 | 34 | go:0005743 |
| Microarray QiagenArray | 12 | 14 | go:0009117 |
| http://legr.liv.ac.uk/carpbase/carpbase_1... | 13 | 15 | go:0005743 |

**Fig. 10. Related documents retrieved for UniProt P26364 using the procedure of section 5.4 and top searches retrieving them.**

## *5.6 Improved Search of the Biomedical Scientific Literature for Documents Related to a Proteomics Identifier*

While enhanced web search for proteomics identifier-related documents is important, another important and relevant corpus to provide improved access to is the biomedical scientific literature. PubMed [40] is a free search engine to the biomedical literature offered by the United States National Library of Medicine as part of the Entrez information retrieval system. PubMed indexes and provides search access to, among a few other sources, the MEDLINE (citations from the mid 1960s to present), and OLDMEDLINE (pre mid 1960s citations) databases as its core content. PubMed is the primary place where researchers search and access the biomedical literature. In PubMed, articles are indexed and searchable by words in the title, abstract, author names, journal name, date of publication, etc. (the full text of articles is not searchable, however). In addition, articles are annotated with terms from the Medical Subject Headings (MeSH) standardized biomedical vocabulary [108], and search may also be done by MeSH terms. Some biological database identifiers are attached to some PubMed citations (citations where the database identifier was referenced somewhere in the full text of the article), but these are manually added and are limited in number. In particular, SwissProt (curated, smaller portion of UniProt) identifiers are attached to some citations (however not all SwissProt identifiers are) but TrEMBL (automatically generated, much larger portion of UniProt) identifiers are not; in general, SwissProt database entries list a small number of related PubMed citations and the PubMed records for these citations are the ones with attached SwissProt identifiers. There exists no access to PubMed, however, which allows high-quality automated retrieval of relevant citations for proteomics identifiers such as

SwissProt or TrEMBL, and such access would be practically useful if available. Interestingly, the highly touted Google Scholar [109] (Google's search engine of the academic literature) does return some results for searches of biological identifiers such as P26364, but the results it returns are negligible (often no results at all) and unsatisfactory. This thesis demonstrates how such access can be provided, and empirically measures its performance using a curated, "gold-standard" bibliography of PubMed citations related to particular yeast proteins.

The PubMed database can be acquired for free from the National Library of Medicine. It consists of more than 500 XML files which together contain over 15,000,000 literature citations from over 4800 biomedical journals and consume several hundred gigabytes of disk space uncompressed; updates are periodically released and can be downloaded to maintain timeliness, with full releases done yearly. For this thesis, the PubMed release covering literature citations up to the end of 2005 was obtained and used for all applications and experiments described below. The open source program Swish-e [110, 111] was used to index the PubMed XML and provide basic keyword-based search access to it (i.e. Swish-e plays the same role Yahoo did in the web search application described in section 5.5 above). Using Swish-e to execute base searches of PubMed then, it was straightforward to implement the basic procedure (described in section 5.4 above) to retrieve PubMed citations relevant to a proteomics identifier and this was done. As an example, a text-formatted version of the top 20 results (and the base searches pulling in the most relevant documents on average) of a search for relevant citations for UniProt P26364 (the same identifier whose web search results were shown in figure 10) is given in appendix 4. In manually perusing the results they seem reasonably good --- they are all

about yeast and/or protein kinases, which is what one would intuitively expect since P26364 is a yeast adenylate kinase protein. Again, it is not possible to do an automated search of PubMed for documents related to P26364, so we cannot directly make a comparison of the LinkHub-derived results (shown in appendix 4) and the results of a simple search for 'P26364' as was done for the Yahoo web search application in section 5.5 above. However, we can examine the citations which have been manually attached to P26364 and they are shown in figure 11.

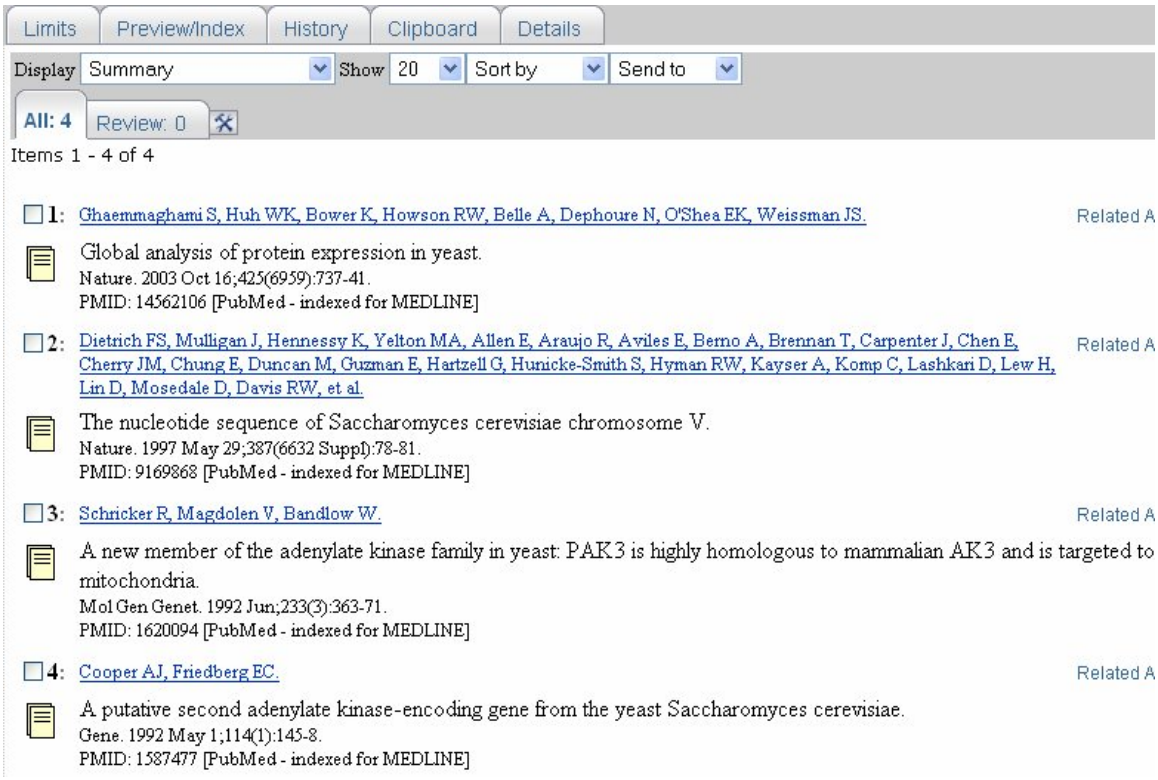



**Fig. 11. Manually annotated PubMed citations for UniProt P26364 on 7/27/2006.**

There are only 4 PubMed citations for P26364, and in fact citations 1 and 2 are really more generally relevant (they are about the yeast genome/proteome as a whole or in large, not specific to P26364), so there are really only 2 specifically relevant citations,

items 3 and 4. Also note that citations 3 and 4 correspond to results 13 and 7 respectively in the LinkHub-derived results given in appendix 4. So using LinkHub we were able to both find the manually annotated relevant citations at a very high ranking, but also many other specifically relevant citations. Again, it can be argued that the LinkHub-derived results are superior.

## *5.7 Empirical Performance Results of LinkHub-based Retrieval of Biomedical Scientific Literature Documents Related to a Proteomics Identifier*

While the LinkHub-derived search results from the web and PubMed in the examples above are typical and seemed to provide good results, ideally we would like empirical measures of how well information retrieval based on word weight vectors derived from the LinkHub relational graph works. Fine-grained judgments about the relative relevance rankings of documents are difficult, e.g. in appendix 4 is result 7 really more relevant than result 13, or vice versa --- a difficult, subjective call to make. It is more reasonable and straightforward, however, to accurately make coarse-grained judgments about relative relevance. For example, while it is difficult to say which of results 7 or 13 should really be ranked higher, we can confidently say that they are both more relevant to UniProt P26364 than articles about, say, earthquakes, global warming, or even more related areas such as medical ethics or surgical procedures, and should thus be ranked higher than articles on such topics. The goal will be to form separate groups of documents where the relative ordering of the groups is clear. In other words, if group A contained appendix 4 results 7, 13 and other similar citations while group B contained

documents about earthquakes, then we can correctly say that all of the group A documents should be ranked above all of the group B documents; we cannot, however, say anything about the relative rankings of documents *within* group A or group B.

Such fine grained distinctions among documents' relevance are also not really necessary or useful to users. Practically, a user, depending on how committed they are to rooting out relevant results, will only peruse a small portion of the results at the top of the ranked list of results. A casual user might only look at the first results page (maybe 25 documents), while progressively more intent users might look at 50, 100, or 200 results. A very committed user, say someone preparing to write a literature review on some topic, might look at 500 to 1000 results, but likely not more. Also, while it is ideally better to have the most relevant results ranked highest, practically, whether the most relevant results are ranked at the top or at the bottom *of the subset of results that the user looks at* does not matter --- getting the relevant results somewhere in the top 50, 100, or 200 (or however many results the user is willing to look at) is what really counts.

Guided by these ideas then, we test the performance of the LinkHub-based information retrieval of PubMed as follows. There exist manually curated bibliographies of PubMed citations relevant to particular genes or proteins. UniProt itself provides citations in its protein entries, but there are relatively few and also it is better to use a UniProt-independent bibliography (since web pages of UniProt entries are used in constructing the word weight vectors). A better bibliography is available from the SGD (yeast genome database) website and provides a large number of relevant citations for yeast proteins; the file is named gene_literature.tab and is available from the SGD ftp directories (linked to from the SGD web site) [112]. The performance test, then, is based

on gene_literature.tab. In addition to gene_literature.tab, we also form a group of PubMed citations which are likely to be completely irrelevant to proteomics (this group is called the *out* group). The *out* group is formed by sampling random citations from PubMed which do not appear in gene_literature.tab or as a citation in any UniProt (SwissProt or TrEMBL) entry. For a given yeast protein, its associated citations from gene_literature.tab are called its *in* group, and citations associated with other yeast proteins are called its *mid* group. It is reasonable to assert, then, that for any given yeast protein the correct relevance ranking of its *in*, *mid*, and *out* group is: *in – mid – out*. Again, we cannot assert anything about the relative citation rankings within the *in*, *mid*, or *out* groups, but we can confidently assume that all *in* group citations should be more relevant than all *mid* group citations, which should themselves all be more relevant than all the *out* group citations. To measure performance, then, we construct the word weight vector for a given yeast protein and use it to score and rank all the *in*, *mid*, and *out* group citations; the deviation of the word weight vector's ranking from the assumed correct ordering of *in – mid – out* is then the measure of performance (the closer to the assumed correct ordering, the better). Practically, however, note that the most important thing is that the *in* group be correctly ranked ahead of the *mid* and *out* groups, and the correct relative ordering of the mid and out groups is of secondary importance; we thus focus on assessing the ranking of the *in* group documents.

### 5.7.1 Concrete Measures of Performance

To concretely measure the deviation of a yeast protein's word weight vector ranking of citations from the assumed correct ranking we use the area under the **receiver operating curve** (abbreviated **ROC**); this area is itself abbreviated **AUC** [113-115]. The ROC

originated in signal detection where it was used to characterize the tradeoff between hit rate and false alarm rate over a noisy channel. To generate a ROC for a binary classifier over a set of test instances, one first uses the classifier to make predictions for all the members of the test set and then rank order these instances descending by the scores given by the classifier (which might, e.g., be probabilities but in our case are cosim values); the higher an instance is ranked the more likely it is that it is truly a positive instance. One generates an ROC from this ranked list of test instances, then, by successively forming larger subsets of the top ranked instances (starting from the subset containing the single highest ranked instance) and then plotting the fraction of all the true positives (TP) in the subset versus the fraction of all the true negatives (TN) in the subset. The ROC thus shows you the tradeoff of true positive rate (good) versus false positive rate (bad). Equivalently, the ROC shows the tradeoff in *sensitivity* of a binary classifier for varying values of 1 – *specificity*. The ROC is a good measure in our case because it can measure performance of a classifier without regard to class distribution or error costs, neither of which we know; i.e. for a given protein we don't know the ratio of truly relevant citations to not relevant ones, nor do we know the relative costs of false/true positives and false/true negatives. The ROC can be summarized in a single measure by taking the area under it, i.e. AUC; AUC can vary between 0 and 1, where larger is better (i.e. 1 means perfect predictive accuracy, 0.5 is worst and represents random chance, while 0 means perfect inverse prediction accuracy). The AUC can be interpreted as the probability that a randomly chosen positive instance will be ranked higher than a randomly chosen negative instance by the classifier. Figure 12 above illustrates these ideas through example ROCs.

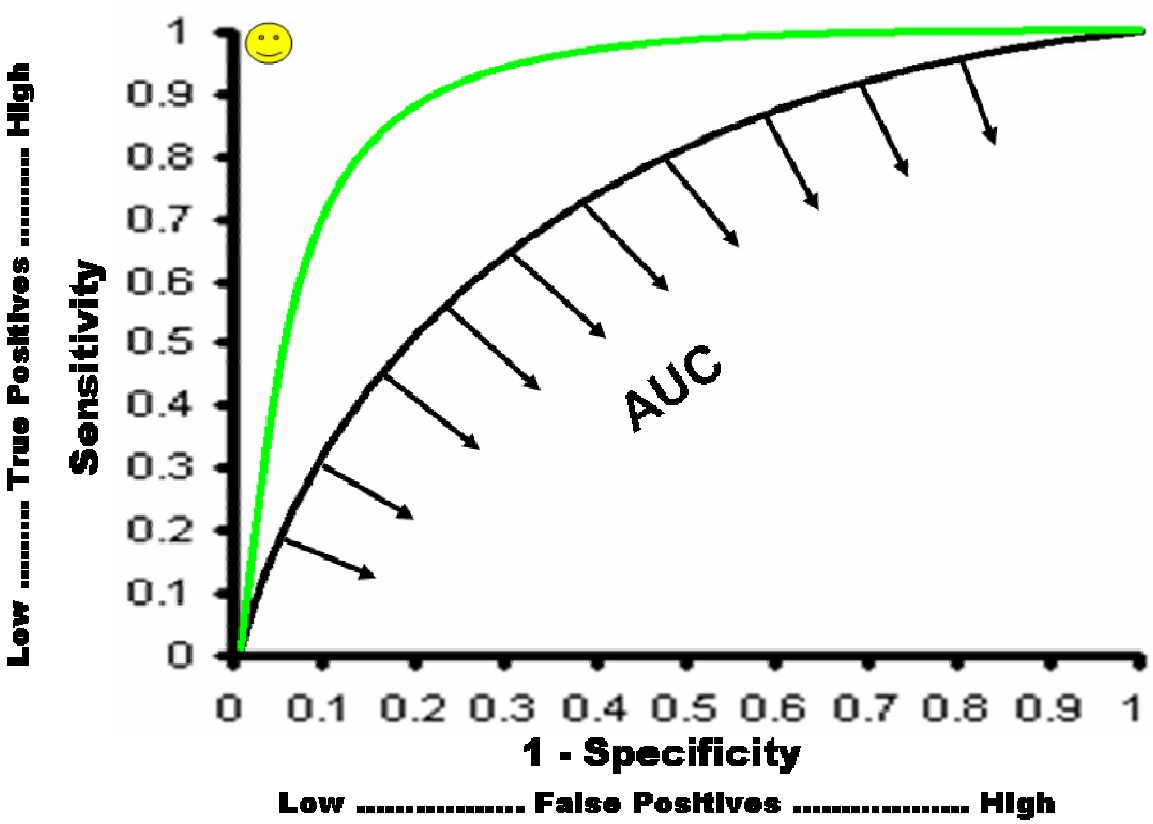



**Fig. 12. Example ROC curves.** The green curve indicates better classifier performance than the black curve. The arrows indicate how the AUC is determined. The yellow "smiley face" in the upper left corner represents an ideal classifier, one which predicts all true positives and no false positives.

Midway between the full ROC and the single number summary of it, AUC, one can consider the area under different parts of the ROC. For example, the 0.1 AUC would measure the partial area under the ROC up to a false positive rate of .1 (i.e. up to an X axis value of .1). In general, the $N/100$ ROC measures the area under the ROC curve up to false positive rate $N/100$, and the value can range from 0 (worst) to $N/100$ (best).

Since the performance of search results in the top of the rankings is so important, measuring $N/100$ AUC for small values of N can better illuminate the performance in

the top of the rankings. We thus measure $N/100$ AUC for values of N in the range 1 to 100 (the value for 100 is simply the normal, full AUC) for the *in* group, e.g. for the *in* group $N/100$ AUC calculation the *in* group citations are like a classifier's 1 values while the *mid* and *out* group citations are combined to be like a classifier's 0 values. We will specifically discuss the results of the .05 (i.e. $5/100$) and 1.0 (i.e. normal, full) AUC values, which are representative respectively of the performance at the important top of the results and of the overall performance.

In addition to simply measuring AUC values we would also like to know if tried optimizations, such as adding in weighted PFAM and GO pages, lead to statistically significant improvements in performance (as measured by AUC values). For example, even if some improvement is observed when PFAM and GO pages are added in addition to just the base UniProt page in forming the combined word weight vector, is this improvement statistically significant or instead likely due to chance? As the section below on experimental protocol describes, we will be running tests for many UniProt proteins, each multiple times under different conditions (e.g. different optimizations). Essentially, what we want to know is if the mean AUC values from the sample of UniProt proteins under some condition A (e.g. word weight vector from base UniProt page plus weighted GO and PFAM pages) is statistically significantly different from that sample of UniProt proteins under some other condition B (e.g. word weight vector from only the base UniProt page). This problem is exactly the problem solved by the paired Student's t-test [116] which will thus be used for statistical evaluation of the results. The AUC results and statistical tests of significance on them are given below in section 5.7.5.

### 5.7.2 Goals of the Experiments

The experiments with gene_literature.tab are being conducted to answer the following questions:

- To get hard data on how well the word weight vectors perform on average at retrieving citations from the scientific literature relevant to UniProt proteomics identifiers. How close do they come to achieving perfect classification accuracy, i.e. a value of 1.0 for the full AUC and 0.05 for the 0.05 AUC?
- The basic procedure given in section 5.4 described how the documents in an identifier's N-subgraph are weighted and added together to create the combined word weight vector used for scoring and ranking documents. An important question is whether adding in multiple, related documents actually helps. PFAM and GO are key related concepts for proteins, and as a practical proof-of-concept we explore them. In particular, can you improve retrieval by adding in related, but not synonymous, documents such as GO and PFAM or is it better simply to just use the direct pages of an identifier (e.g. only the UniProt entry page) and its synonyms. If adding in non-synonymous but related pages does help, then what are the optimal weights for this; in particular, the values of .75 for PFAM and .5 for GO used in sections 5.5 and 5.6, while chosen somewhat arbitrarily, gave pleasing results but what are the optimal values to weight PFAM and GO pages?
- To test whether certain enhancements to the basic procedure can improve performance. In particular, one enhancement that dramatically enhanced performance as will be shown below was a **pre-IDF step** of LinkHub relational

subgraph identifier node-linked documents against document frequency statistics computed for the databases they came from.

### 5.7.3 Pre-IDF Step

The Pre-IDF step (or pre-inverse document frequency step) enhancement to the basic procedure of section 5.4 is one novel contribution of this thesis and takes account of the fact that, for a given proteomics identifier we not only know its subgraph and the documents linked to identifier nodes in it, but we also know the subgraphs and identifier node-linked documents for all other proteomics identifiers and we can leverage this "big picture" information to improve ranking accuracy for all proteomics identifiers. In particular, we have information about the subgraphs and identifier node-linked documents for all UniProt identifiers. For a given proteomics identifier it is the relevant documents for all the other proteomics identifiers that will be most similar to the relevant documents of the given proteomics identifier and thus the most difficult to correctly discriminate. In essence, we want to create word weight vectors for all proteomics identifiers that are maximally different from one another while at the same time each being correctly as specifically relevant and discriminating as possible.

Section 5.3 above described formally IDF re-weighting of word weights, and the idea here is simply to do such IDF re-weighting twice, ultimately as usual against the corpus (e.g. the web or PubMed) you are interested in searching but also first against document frequencies computed for all (or a reasonable sample of) web pages of the same type as a given web page. Thus, for example, UniProt (or PFAM, GO, etc.) has many individual, identifier-specific pages which can together be considered a corpus, and we can compute document frequency statistics for them and use these statistics to do IDF

re-weighting of individual UniProt (or PFAM, GO, etc.) web pages. In step 4 of the basic procedure given in section 5.4, before adding together the separate word weight vectors for web pages linked to identifiers in the N-subgraph, for each such separate word weight vector perform an IDF re-weighting of its word weights against the document frequencies of its corresponding type (if available). For example, if a word weight vector was generated from a Pfam page, then re-weight its words by an IDF step against document frequencies computed for all Pfam pages. This has the effect of down-weighting words which occur frequently in pages of a type (e.g. *protein* or *sequence* in UniProt pages) and are thus less discriminating while up-weighting words that occur infrequently (which are intuitively the most distinguishing words). For completeness, step 4 of section 5.4 becomes as follows:

4 Form a word weight vector for each identifier node-linked document as described in section 5.3 scaled (multiplied) by their document's weight. If document frequency statistics are available for a document's type, perform IDF re-weighting of the document's word weight vector against those document frequency statistics (pre-IDF step). Add all documents' word weight vectors to form the combined word weight vector. Eliminate some percentage of the smallest weighted terms. The combined word weight vector is used for ranking documents by the cosine similarity measure.

Note that the pre-IDF step also has the added benefit of effectively filtering out words which appear as part of the "template pattern" of pages of a type and are thus not

real information-bearing terms. For example, all UniProt pages follow a common template, e.g. have the text "UniProtKB Entry" and "NiceProt view" at the top, etc., and words which are constants in the template are effectively filtered out by the pre-IDF step since they occur in every page and thus $\mathrm{Log}\left(\frac{N}{D}\right)$ is 0 because D = N. Thus, there is no need to write special, type-specific HTML parsers to eliminate such constant template content, as it is handled simply as a by-product of the pre-IDF step. In general, web pages can be difficult to deal with for information retrieval, as they have many more extraneous, unrelated terms than well-focused English text, e.g. scientific papers, newspaper and magazine articles, etc., and the pre-IDF step can thus be especially useful for information retrieval using web documents as queries. For example, all UniProt entry pages contain cross-references to other databases, and the name of the database that is being cross-referenced to is given in the page (and often there are multiple cross-references to the same database, so that database's name will appear multiple times). Without the pre-IDF step, these database names bubble to the top as the most highly weighted words (because they occur with such high frequency in the UniProt pages and are also relatively rare in PubMed, against which the usual IDF step is done, because they are relatively modern terms). This is clearly incorrect and the pre-IDF step severely down-weights (or eliminates altogether) these database names.

### 5.7.4 Experimental protocol

We perform experiments on random samples of UniProt identifiers. UniProt consists of two parts, Swiss-Prot and TrEMBL (which stands for "translated embl" where embl is a DNA sequence database). Swiss-Prot is the much smaller part, but is of higher quality,

having been manually curated. TrEMBL is much larger than Swiss-Prot and is of lower quality due to its being generated by automated processes. We do separate experiments for SwissProt and TrEMBL. For SwissProt, we pick a random set of 200 identifiers which have GO and PFAM annotations and at least 20 citations in gene_literature.tab; for TrEMBL we do the same but only pick 100 random TrEMBL identifiers (since TrEMBL is lower quality I was afraid it might not be possible to find a full 200, so I settled for 100 which is still a good sized sample). We perform a grid search optimization procedure, at a granularity of 0.1 for all variables being optimized, to find the optimal values for PFAM documents' weight, GO documents' weight, and the percentage of features (i.e. words) to keep in the word weight vectors (see step 4 of the basic procedure given in section 5.4); UniProt entry pages are always weighted 1.0. We might hypothesize that PFAM and GO pages would be more likely to improve information retrieval performance for TrEMBL pages, given that TrEMBL pages have less information since they are not manually curated; also, PFAM and GO pages might not improve (or improve less) performance for SwissProt pages since they are high quality, manually curated pages and thus are likely to be fairly complete statements about the proteins they describe (and so PFAM and GO pages might, in effect, be redundant and not add information). The experimental results, described next, answer these questions. The code to do these experiments was written in Perl, and borrowed code from the `AI::Categorizer` Perl module (http://search.cpan.org/~kwilliams/AI-Categorizer-0.07/) and the `Lingua::Stem` Perl module (http://search.cpan.org/dist/Lingua-Stem/).

### 5.7.5 Results

***TrEMBL***

Appendix 5 contains 2 tables giving respectively the average .05 and 1.0 AUC values (sorted descending by AUC value, then by percentage features kept) for the 100 randomly sampled TrEMBL proteins and for different trials with different values for the four parameters *GO Wt, PFAM Wt, Perc Features Kept,* and *pre-IDF applied*. .05 AUC is indicative of performance at the important top of the results ranking, while 1.0 AUC is simply the normal, full AUC value and indicates overall performance. Figure 13 below reproduces the important result rows from appendix 5, and these and appendix 5 are the basis of the discussion that follows below.

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | .05 AUC |
|---|---|---|---|---|
| 0 | 0.2 | 0.5 | 1 | 0.032269328 |
| 0 | 0 | 0.5 | 1 | 0.03131128 |
| 0 | 0.6 | 0.5 | 0 | 0.020527959 |
| 0 | 0 | 0.5 | 0 | 0.017914374 |

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | 1.0 AUC |
|---|---|---|---|---|
| 0 | 0.1 | 0.5 | 1 | 0.927418459 |
| 0 | 0 | 0.5 | 1 | 0.920262344 |
| 0 | 0.4 | 0.5 | 0 | 0.860134119 |
| 0 | 0 | 0.5 | 0 | 0.849313752 |

**Fig. 13. Important .05 and 1.0 AUC results for 100 randomly sampled TrEMBL proteins.** The overall optimal .05 AUC is achieved by use of the pre-IDF step and with a PFAM weight of 0.2 and a GO weight of 0.0. The overall optimal 1.0 AUC is achieved by use of the pre-IDF step and with a PFAM weight of 0.1 and a GO weight of 0.0.

The first thing to note is how the pre-IDF step separates the trials into two large groupings; for both .05 and 1.0 AUC, any trial, regardless of the values of the other parameters, that used the pre-IDF step gave a better result than any trial that did not

(again, regardless of any other parameters), so the pre-IDF step seems to be clearly advantageous for performance. Also, the AUC results seem robust to changes in the percentage of features kept, i.e. for a given PFAM and GO weighting the AUC values for all the values of percentage features kept tried (1.0 down to 0.1) do not change much and most give the same value. There is a trend, however, for larger values of percentage features kept to give slightly better AUC values, but generally 1.0 down to 0.5 or 0.4 for percentage features kept result in the same AUC values; very small values for percentage features kept of 0.1 or 0.2 do show some drop-off but it seems marginal. Keeping more features requires more computation time since the computation time for the cosine similarity measure will increase linearly with the number of features, i.e. length of the word weight vectors. Thus, the tables in appendix 5 indicate that only a small percentage of the features need to be used to achieve good performance; 0.5 or 0.4 probably represent the best tradeoff between computation time and performance. The results that follow will be based on using a value for percentage of features kept of 0.5.

To quantify the performance increase obtained by the pre-IDF step, let us look at the average .05 and 1.0 AUC values for the base case when only a protein's UniProt TrEMBL page is used to construct the word weight vector (i.e. PFAM Wt and GO Wt are 0). Note that the top possible scores for the .05 and 1.0 AUC values are respectively .05 and 1.0. From figure 13 (and the tables in appendix 5), for this case the average .05 and 1.0 AUC values without the pre-IDF step are respectively 0.01791437 and 0.849313752 and with the pre-IDF step are respectively 0.03131128 and 0.920262344. Thus, the pre-IDF step increases the .05 AUC value 75% and the 1.0 AUC value over 8%. The percentage increases in AUC are greater the farther you go to the left (i.e. as false

positive rate decreases) in the ROC; for example, the .01 AUC (not shown) increases over 92%. The AUC increase is thus concentrated in the left portion of the ROC, which is what is desired.

While the above showed that the pre-IDF step does improve performance, let us now look at the addition of GO and PFAM pages and see if they improve performance. Looking at figure 13 and the tables in appendix 5 we can see that GO pages do not help and in fact their addition slightly decreases performance (for both .05 AUC and 1.0 AUC and whether or not the pre-IDF step was done). On the contrary, the addition of PFAM pages does boost performance, albeit not as much as the pre-IDF step. For the .05 AUC and with the pre-IDF step, a weight of 0.2 for PFAM pages was optimal and this gave a .05 AUC value of 0.03226933; the pre-IDF step plus addition of PFAM pages weighted 0.2 thus increased the .05 AUC value 80%, so the addition of the PFAM pages increased the .05 AUC value an additional 5% over the baseline value. Similarly, for the 1.0 AUC and with the pre-IDF step, a weight of 0.1 for PFAM pages was optimal and this gave a 1.0 AUC value of 0.927418459; the pre-IDF step plus addition of PFAM pages weighted 0.1 thus increased the 1.0 AUC value 9.2%, so the addition of the PFAM pages increased the 1.0 AUC value a modest additional 1.2% over the baseline value.

Without the pre-IDF step, the addition of PFAM pages is optimal at larger weight and gives a larger performance increase over baseline performance of only using the UniProt TrEMBL page (i.e. no pre-IDF or addition of GO or PFAM pages). Figure 13 and the tables in appendix 5 show the .05 AUC and 1.0 AUC scores to be 0.01791437 and 0.849313752 respectively for when GO or PFAM pages are not added and no pre-IDF step is done. For .05 AUC and no pre-IDF step, adding PFAM at weight 0.6 (the

highest weight tried in the experiment, so possibly going higher would have further increased performance) gave the best result of 0.02052796 for the .05 AUC, an increase of almost 15%. Similarly, for 1.0 AUC and no pre-IDF step, adding PFAM at weight 0.4 gave the best result of 0.860134119, an increase of 1.3%. Again note that it is the performance at the top of the results, corresponding to the left region of the ROC (which the .05 AUC measures), which is most important. Thus, if data were not available to perform the pre-IDF step, the addition of PFAM pages can be a useful alternative to increasing performance (albeit by not as much).

To statistically validate the conclusions reached above which looked at average AUC values, we use the Student's paired t-test over the 100 randomly sampled TrEMBL proteins; we use the one tail P-value because our hypothesis is that one of the two test conditions (e.g. pre-IDF step versus no pre-IDF step) will result in an improved AUC value. Appendix 6 contains the results of the t-tests. All tests returned highly significant P-values, with the largest being 0.002952176 which is still much smaller than the commonly accepted .05 level of significance. Thus, we can conclude that the addition of PFAM pages appropriately weighted and the use of the pre-IDF step both significantly increase information retrieval performance as measured by AUC values for UniProt TrEMBL identifiers.

***Swiss-Prot***

Appendix 7 contains 2 tables giving respectively the average .05 and 1.0 AUC values (sorted descending by AUC value) for the 200 randomly sampled Swiss-Prot proteins and for different trials with different values for the three parameters *GO Wt, PFAM Wt, pre-*

*IDF applied*. *Perc Features Kept* was set at the constant 0.5 since the performance doesn't vary significantly when it is changed and 0.5 was used as the basis of the comparisons for TrEMBL above (and is thus used for all comparisons here too). Figure 14 below reproduces the important result rows from appendix 7, and these and appendix 7 are the basis of the discussion to follow.

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | .05 AUC |
|---|---|---|---|---|
| 0 | 0.1 | 0.5 | 1 | 0.035710069 |
| 0 | 0 | 0.5 | 1 | 0.035667273 |
| 0 | 0.2 | 0.5 | 0 | 0.025253218 |
| 0 | 0 | 0.5 | 0 | 0.024920735 |

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | 1.0 AUC |
|---|---|---|---|---|
| 0 | 0.1 | 0.5 | 1 | 0.950538504 |
| 0 | 0 | 0.5 | 1 | 0.950253129 |
| 0 | 0.1 | 0.5 | 0 | 0.897383042 |
| 0 | 0 | 0.5 | 0 | 0.897098083 |

**Fig. 14. Important .05 and 1.0 AUC results for 200 randomly sampled Swiss-Prot proteins.** The overall optimal .05 AUC is achieved by use of the pre-IDF step and with a PFAM weight of 0.1 and a GO weight of 0.0. The overall optimal 1.0 AUC is achieved by use of the pre-IDF step and with a PFAM weight of 0.1 and a GO weight of 0.0.

The same as for TrEMBL the pre-IDF step separates the trials into two large groupings; for both .05 and 1.0 AUC, any trial, regardless of the values of the other parameters, that used the pre-IDF step gave a better result than any trial that did not (again, regardless of any other parameters), so the pre-IDF step again seems to be clearly advantageous for performance.

To quantify the performance increase obtained by the pre-IDF step for Swiss-Prot proteins, let us look at the average .05 and 1.0 AUC values for the base case when only a protein's UniProt Swiss-Prot page is used to construct the word weight vector (i.e. PFAM Wt and GO Wt are 0). From figure 14 and the tables in appendix 7, for this case the

average .05 and 1.0 AUC values without the pre-IDF step are respectively 0.024920735 and 0.897098083 and with the pre-IDF step are respectively 0.035667273 and 0.950253129. Thus, the pre-IDF step increases the .05 AUC value 43% and the 1.0 AUC value 6%. Again, the percentage increases in AUC are greater the farther you go to the left (i.e. as false positive rate decreases) in the ROC. The AUC increase is thus concentrated in the left portion of the ROC, which is what is desired. The percentage increases for Swiss-Prot, while still substantial, are not as large as for TrEMBL and this is consistent with the fact that Swiss-Prot, being manually curated, is of higher quality and thus there is less need or room for improvement compared to TrEMBL However, TrEMBL is much larger than Swiss-Prot and generated by automated processes, and it is thus practically very useful that TrEMBL can be improved more, reaching close to parity with Swiss-Prot. The improvements in both mean .05 and 1.0 AUC from the pre-IDF step are also both statistically significant as is shown in appendix 8.

As was conjectured above, the addition of PFAM and GO pages does not help improve performance for Swiss-Prot as much as for TrEMBL. First, the same as for TrEMBL, the addition of GO pages does not help and in fact decreases performance slightly. The addition of PFAM weights at small weight very slightly increases average .05 and 1.0 AUC for all cases compared, however all but one of these increases are not statistically significant (see appendix 8 for details). Thus, the addition of PFAM pages cannot be said to significantly improve performance (although they don't decrease it either).

### 5.7.6 Discussion

Overall, for a small PFAM page weight (and no GO pages added) and performing the pre-IDF step we achieve .05 AUC and 1.0 AUC scores of 0.03226933 and 0.927225042 respectively for TrEMBL and 0.035710069 and 0.950538504 respectively for Swiss-Prot. A very recent study [117] (covered in more detail in the next chapter) constructed classifiers for predicting relevance of PubMed documents for various clinical medicine themes. This study used state of the art support vector machine classifiers trained on relatively large, manually curated and respected bibliographies of articles in various clinical medicine disciplines, and used text from the article title, abstract, journal name, and MeSH terms for features. In contrast, the experiments in this thesis only used the abstract text for features, used only relatively small training sets (i.e. the single UniProt page plus a few GO and PFAM pages), and, in comparison to support vector machines, used only a fairly basic classifier model (i.e. word weight vectors compared with the cosine similarity measure). In addition, it should be pointed out that a small number of the citations in the yeast gene_literature.tab bibliography are really not directly related to the yeast proteins they were assigned to, but are more generally related; for example, the first citation in figure 11 ("Global Analysis of Protein Expression in Yeast") is assigned to several proteins in gene_literature.tab. Thus, these more general citations really should be ranked lower even though the assumption of the experiments is that all gene_literature.tab citations should be ranked highest; the AUC values obtained for these experiments on the LinkHub-based information retrieval thus understate slightly the true performance. In spite of these seeming deficiencies, it is notable that the procedure of this thesis achieved average 1.0 AUC scores of .927 and 0.951 for TrEMBL and Swiss-Prot respectively in

ranking PubMed documents for their relevance to UniProt proteins which is better than or negligibly smaller than the Aphinyanaphongs et al 2006 study which achieved a 1.0 AUC score of 0.893 on one of the clinical medicine bibliographies and 0.932 and 0.966 on two others[2].

While it was shown that PFAM pages improve performance, it was initially expected that GO pages would also and this turned out not to be the case. Qualitatively, when you look at typical examples of PFAM and GO pages the PFAM pages seem more information rich; they generally have several fairly detailed and long but concise paragraphs describing important functional and other information about their PFAM protein domain, and have less extraneous unrelated terms. GO pages, on the contrary, have shorter and less detailed textual descriptions and more extraneous terms (e.g. all the higher level terms in the GO hierarchy are given). Thus, in retrospect it makes sense that PFAM pages would be better at improving performance and this is borne out by the results presented here. In general, UniProt, GO, and PFAM web pages (and for the most part web pages in general, which often have e.g. advertisements, navigation text, etc.) have many more extraneous, unrelated terms than well-focused English text (e.g. paper abstracts, introductions, etc.) and it is noteworthy the reasonably good performance achieved in spite of this.

The pre-IDF step was shown to be the most effective optimization for increasing performance. In fact, this idea is generally useful, as many pages on the web are generated by templates and are on common topics. For example, the CNET web site has

[2] Note that this isn't an exact "apples to apples" comparison but it is still a reasonable comparison to make. Both the study in this thesis and the Aphinyanaphongs et al 2006 study used the same objective measure of performance, namely the AUC, and both achieved comparable, nearly perfect results (and thus neither leaves much room for significant improvement) on the same kind of classification task (i.e. ranking PubMed citations for relevance), although not on the exact same tasks.

in-depth product reviews and information for many types of electronics products at http://reviews.cnet.com/. Thus, for example, there are separate CNET pages for each different digital camera or mp3 player, and they all follow the common CNET template. If we wanted to search the web for pages relevant to a particular digital camera (or mp3 player, etc.), we could compute document frequencies for all CNET digital camera (or mp3 player, etc.) pages and do a pre-IDF step against it for the word weight vector formed from the CNET page for a single, particular digital camera (or mp3 player, etc.) If and when the semantic web becomes widespread, and semantic web metadata becomes commonly associated with standard web pages and other documents, these ideas could be even easier to use and widely applicable in an automated way. Even now, there exist "tagging" systems where people can attach short text description "tags" to web documents, with a prominent site being Connotea [33] which is run by the journal Nature and focuses on science and technology. All of the ideas of this thesis could be applied in a fairly straightforward way to build effective word weight vectors for the tags of Connotea and other tagging sites, to be used for fine-grain retrieval of documents relevant to the underlying concept of the tags.

The pre-IDF step idea could also be used as part of a coarse-to-fine cascade of classifiers for possibly better performance. For example, to find pages relevant to a given UniProt protein, we could first construct a combined word weight vector from all (or a sample of) UniProt entry pages. This combined UniProt word weight vector would presumably rank pages that are in the domain of proteomics higher than non-proteomics pages. The results of this initial UniProt word weight vector could then be filtered below some threshold. The remaining, non-filtered ranked results could then be ranked again by

the word weight vector (modified by the pre-IDF step against UniProt document frequencies) specific to the given UniProt protein's word weight vector. Higher-level word weight vectors could also be used, e.g. constructed from all of PubMed and ranking highly documents that have a biomedical theme.

We could also likely do something like the pre-IDF step at the higher levels, e.g. to make a whole UniProt word weight vector as different as possible from one for all of PubMed and as specific as possible for proteomics. Here, we might do it differently since the situation is not a single page of a given type but rather two different large sets of documents where effectively one (all the UniProt pages) represents a subtopic, i.e. proteomics as a subtopic of biomedicine, of the other (all of PubMed). It might make sense here to consider relative document frequencies between the two document sets, e.g. the weight of word $w_i$ in the whole UniProt word weight vector could be reweighted by multiplying it by $\text{Log}\left(DF_U^i / DF_P^i\right)$ where $DF_U^i$ is the percentage of documents in UniProt containing $w_i$ and $DF_P^i$ is the percentage of documents in PubMed containing $w_i$. This will have the effect of upweighting words that occur relatively more frequently in UniProt versus PubMed (and thus are more specific and relevant to it) and downweighting words that occur at the same or lower frequency (which are thus not specific or relevant to it). This thesis does not empirically explore this idea of a coarse-to-fine classifier cascade or this different way of doing a pre-IDF step, but they are intriguing and are given as possible future directions to explore.

# Chapter 6 Related Work, Contributions, and Conclusions

## *6.1 Data Interoperation: LinkHub and YeastHub*

While the contributions of this thesis specific to data interoperation are more subjective and possibly limited compared to the information retrieval aspects of the thesis, they are still noteworthy and this section will cover them. Section 3.3 gave an overview of the two main approaches to data interoperation, data warehousing and federation, and referred to some related, representative systems for biological data interoperation based on these two approaches. In fact, there is no need to rely solely on one of these two techniques and there are advantages and efficiencies to be gained by combining them. This thesis thus espouses a hybrid approach between these two extremes and LinkHub enables this: individual LinkHub instantiations (such as at hub.gersteinlab.org) are a kind of mini, local data warehouse of commonly grouped data, and individual LinkHub instantiations can then be connected to larger major hubs such as UniProt (as is done with hub.gersteinlab.org) in a federated fashion; efficiency is gained by obviating the need for all the individual source datasets to be connected directly to each other or individually to the major hubs. As a practical by-product, LinkHub also allows the creation of a single point-of-entry to the many separate web resources within a lab or organization; the LinkHub web GUI interface presents a simple and intuitive way to navigate these many web resources (and their relationships through biological identifiers) and is essentially a "Links Portal" to them.

There is a need for standards and tools enabling the genomics and proteomics communities to harness their own collective efforts towards connecting their vast amount

of data in a loosely coupled collaborative manner, without requiring explicit coordination or centralization. The position of this thesis is that the semantic web is the logical basis for such standards, and LinkHub and YeastHub are such tools. Chapter 1 described how the semantic web allows what could be termed incremental data warehousing, allowing one to make partial, incremental progress in data interoperation without requiring the complete problem to be solved. The YeastHub and LinkHub systems described in this thesis were based on semantic web technologies in order to achieve such practical partial progress in biological data interoperation, the complete solution of which is impractical due to the large and changing size, and independent distribution, of biological data. The semantic web is still in an early stage and not so widespread, and YeastHub and LinkHub are noteworthy as early explorations and testbeds for biological data interoperation through the semantic web.

The YeastHub work identified important issues that would need to be addressed to enable interoperation of biological data, such as syntactic conversion to a common format, and built a practical system addressing many of them. The LinkHub system addressed an important remaining problem, the need to find, store, and work with identifiers and the relationships among them (i.e. ontology alignment or mapping of biological identifiers). This is a high-level, important structuring principle for biological knowledge, and LinkHub complemented YeastHub by providing such connections. Several example queries were given of the LinkHub and YeastHub RDF data to show that even the relatively basic data integration provided by these two systems could enable quite useful, interesting, and non-trivial exploratory data analysis with RDF query languages across many disparate datasets and roughly duplicate the results of some

published works; this demonstrates the value of the semantic web for allowing practical, incremental progress towards integrated biological data analysis to be made.

An important future direction for LinkHub and YeastHub would be to explore how other relevant semantic web-related technologies could be effectively used with them, in particular named graphs [118] and Life Science IDentifier or LSID [119]. Named graphs allow RDF graphs to be named by URI, allowing them to be described by RDF statements; named graphs could be used to provide additional information (metadata) about identifier mappings, such as source, version, and quality information. LSID is a standard object naming and distributed lookup mechanism being promoted for use on the semantic web, with emphasis on life sciences applications. An LSID names and refers to one unchanging data object, and allows versioning to handle updates. The LSID lookup system is in essence like what Domain Name Service (DNS) does for converting named internet locations to IP numbers. While LinkHub does store some identifiers named as LSID's (e.g. from the resource pseudogene.org) it would be interesting to explore more fully using LSID for naming objects in LinkHub and incorporating LSID lookup functionality. Finally, like software such as Napster and Gnutella did for online file sharing, one could explore enhancing LinkHub to enable multiple distributed LinkHub instantiations to interact in peer-to-peer networks for dynamic biological data sharing, possibly using web services technologies such as Web Services Description Language or WSDL (http://www.w3.org/TR/wsdl) and Universal Description, Discovery and Integration or UDDI (http://www.uddi.org/) for dynamic service discovery, and available peer-to-peer toolkits.

Two other recent related systems include Tabulator and BioGuide. Tabulator

[120] is an attempt at a semantic web browser developed by Tim Berners Lee, the inventor of the web and head of W3C. It is early stage and has limitations, but it is similar to the LinkHub system in that it uses essentially the same basic interface, namely a dynamic expandable/collapsible list, in presenting an integrated view of RDF data fetched from multiple sites. Another related system BioGuide [121] uses RDF, but it is limited in that it focuses on abstract conceptual modeling of resources and their interconnections rather than on more practical instance data as in LinkHub and YeastHub; also its interface presents the data using Graph drawing software with Java, whereas LinkHub is more lightweight and relies only on the web browser with JavaScript. Finally, there have been a number of graph database systems and graph query languages developed through the years but they suffer from being proprietary and none have developed into widely used standard systems. However, it should be pointed out that some of these systems support advanced graph data mining and analysis operations not supported by RDF query languages and these features might be necessary for effective analysis of biological data represented in RDF [122]; nevertheless, the simplicity of RDF and its query languages are more conducive to widespread acceptance and use (and the relative complexity of the graph database systems might have been their downfall for this).

A key contribution of the data interoperation techniques described and employed in this thesis is simply that they enabled the concrete creation of the LinkHub system which served as the basic, background system supporting the novel information retrieval aspects of the thesis. LinkHub joins relational data with free text data by linking free text web documents to the identifier nodes in the LinkHub relational graph. The "path type"

interface to LinkHub allows one to flexibly retrieve useful subsets of these web documents based on querying the relational structure of the graph, enabling a general kind of combined relational and keyword-based access to documents that normal search engines do not provide. Finally, chapter 5 described how the LinkHub relational graph could be used for enhanced automated information retrieval of documents related to proteomics identifiers; the next section covers the related work, contributions, and conclusions of this very important aspect of the thesis.

## *6.2 Automated Information Retrieval*

This section gives an overview of some important related work to the automated information retrieval aspects of this thesis (chapter 5) and highlights the contributions of the thesis in this area. First, information retrieval is generally considered to be a subarea of *text mining* [123, 124], with the other key area being *information extraction*. Information extraction differs from information retrieval in that it attempts to perform fine-grain analysis of individual documents, e.g. shallow or full parsing of text, or pattern matching, to break it up into different lexical types, named and typed entities, etc., to pull out as much concrete information from the text as possible. Its aims include summarizing text and extracting factual information and individual entities from text, which can then be assigned types and/or meaning and represented in some structured form such as relational tables or even RDF; various kinds of logical inference or other analysis can then be done on the extracted data. For example, a typical problem for an information extraction system would be to take news articles about committed crimes and to extract such information as the location of the crime, when it occurred, who was the victim, etc. For evaluation of information extraction systems, the Message Understanding Conference

(http://www-nlpir.nist.gov/related_projects/muc/index.html) was held throughout the 1990s as a kind of competition among information extraction systems and a showcase of progress in the field. In biomedical text mining research information extraction seems to be the more studied area. Biomedical information extraction systems attempt to extract various kinds of structured information, such as gene and protein names and even entire biological networks or pathways (e.g. see [125]), from the biomedical literature. As described in section 5.3, information retrieval does not attempt to do such fine-grain analysis of text but considers it simply as a "bag of words"; the goal is to return relevant documents for a query, i.e. documents which would, if read, ideally satisfy the information requirements inherent in the query. Biomedical information retrieval is a very important problem and information retrieval systems to the biomedical literature are the most practical text mining tool generally used by researchers (e.g. NCBI's PubMed search engine).The focus of this thesis is on methods for enhanced information retrieval of the scientific literature and the web and not on information extraction.

Information retrieval research has a long history and the pioneer in the field was Gerald Salton. He and his research group developed the basic ideas of information retrieval (which are covered in section 5.3) and built the first information retrieval system based on these ideas, called the SMART information retrieval system [126], during the 1960s. Salton's group also investigated *relevance feedback* [127] where the idea is to take the results that are initially returned from some query and to use information about whether or not those results are relevant to perform a new and ostensibly better query (relevance feedback is in essence for information retrieval what PSI-BLAST [128] is to sequence database search). SMART was a pioneering system and worked well for its

time, but its success was partly based on the primitive computing technology available at the time and the limited amount of text available to search in digital form. In other words, the basic ideas of information retrieval work well for small document collections, which were all that were available computationally at the time, but they do not scale well to large document collections such as the World Wide Web. With the growth of the internet, there came a need to be able to search for information spread across it. Wide Area Information Servers or WAIS [129] was an early and noteworthy system for distributed searching of text, and it was often used as the full text search engine for internet Gopher servers; Gopher [130] is a distributed internet document search and retrieval network protocol whose goal was similar to the Web but has been almost completely displaced by the Web (kind of a proto-web or pre-web web).

The real, practical need for effective information retrieval of large document collections became apparent with the birth and growth of the World Wide Web around the mid 1990s. Early web search engines based on basic information retrieval ideas (or simpler) were painfully inadequate. The problem was that, given the enormous size of the web, there can be millions of web documents that score about the same for a given query of a small number of terms, and useful documents could be anywhere within these vast result sets; as discussed above, the most relevant documents must be among at most about the top 1000 documents or else they are effectively inaccessible to the user. It was the Google search engine's solution [41] to this problem in the late 1990s that made it famous and still the most widely used and respected search engine today. The essence of Google is that it uses what could be construed as primitive semantics in the current web to greatly improve search results ranking, i.e. the hyperlinks (and associated link text)

that web page creators add to their pages. A web hyperlink is a "vote" for the importance of the web document being linked to, and the link text ideally is a short, succinct text snippet telling what the linked to page is about or why it is relevant given the linking page's context. Intuitively Google ranks highest documents that have the most "votes" and this proved to be effective (more specifically votes from more important pages are weighted higher, and Google's famous PageRank algorithm iterates multiple times over the web's hyperlink graph to solve for consistent scores for pages). Nevertheless, Google has a very simple interface and relies on users entering only a few key words to search for, and without Google's effective ranking this could result in millions of potential matches which would be impossible for the user to sift through. Thus, techniques such as Google's are necessary for coarse, topical queries where only a few search terms are entered.

An alternative approach to the problem of too many documents being returned is to try to increase the precision of the initial search, e.g. many terms instead of only a few, so that only a relatively small number of documents meeting the stricter query requirements are returned; this is the approach taken by this thesis. A problem with trying to increase the precision of the initial search is that it will generally require many more terms to be entered, and users likely will not have the desire or time to enter directly themselves a large amount of information for a query. A key idea of this thesis is that this problem can be solved by the semantic web, where the much greater information (compared to a few manually entered words) in the known semantic relationships among identifiers and the textual content of their associated documents can be used to greatly increase the precision of searches. In a vision of how the semantic web could aid search

of the standard web in the future, users would only need to point to nodes in the semantic web (there will have to be search access to the semantic web itself to aid in finding these) and then click “find more like this node” to specify and execute such high-precision searches and the ideas of this thesis could help enable this.

Other work has increased the precision of queries by considering entire documents to be queries, e.g. notably PubMed’s “Related Articles” links [131] to find other articles similar to a given article. This thesis extends this idea to using multiple, weighted documents combined (i.e. from the LinkHub relational subgraph), where some of the combined documents are only indirectly related to the query concept such as PFAM and GO pages to UniProt identifiers, and demonstrates empirically that this approach can improve information retrieval accuracy over just using single documents as queries. This thesis also demonstrates how the accuracy of such queries can be dramatically improved by the use of the pre-IDF step, which can be especially important when web pages are the source query documents since they tend to be less concise and contain more extraneous, non-informative terms than straight English text. Finally, the thesis makes a contribution simply by empirically quantifying how the useful, practical (but currently unmet) need of automatically retrieving related documents to proteomics identifiers can be achieved with high accuracy.

Other research has explored ways of using ontologies to aid information retrieval and some representative recent examples are as follows. [132] considers the problem of ontology grounding, or making mappings between concepts in formal ontologies and terms in text documents which are instances of those concepts, and how this problem can be solved automatically using machine learning support vector machine classifiers

(SVMs). They demonstrate how their technique can be used to assign free text questions on legal topics to appropriate subtopics in a taxonomy of legal topics. [133] follows a similar procedure, where they generate word weight vectors from text (which, interestingly, was originally spoken speech and translated to text; their application focus is what they call "mobile audio-based knowledge management") but only consider and weight terms which are found to be concepts from a task-oriented ontology of common problems in computing (e.g. printer paper jam, UPS connection problem, etc.) LinkHub-based information retrieval is more flexible and easily-deployed; it differentiates itself by focusing on lightweight relations among instance-level data, and does not require a large, complex background ontology (class-level data) be present for which it can be assumed all text documents considered will be related to this ontology (and where only terms which in fact are determined to be related to the ontology are considered).

Most recently is research showing the high performance of SVM classifiers trained on gold standard, manually curated bibliographies for specialized information retrieval tasks [117] (hereafter referred to as Aphinyanaphongs et al 2006). This work notes the need for specialized filters for finding relevant documents in the huge and ever expanding scientific literature:

> "The growth of publication volume in the majority of fields of biomedicine is rapidly becoming intractable. Modern approaches to biomedical information retrieval are seeking to alleviate the problem by developing specialized filters that find documents that satisfy special content or methodological criteria. Such filters have been developed, for example, to identify randomized controlled trials or to

> select documents that focus on prognosis and satisfy rigorous criteria of statistical design and analysis, etc. This Focused Filter paradigm is implemented either via automated methods based on machine learning or on manual and semi-manual construction of search queries tailored to the criteria of interest."

A prominent example of such manually constructed queries is the PubMed Clinical Queries (http://www.ncbi.nlm.nih.gov/entrez/query/static/clinical.shtml), which were painstakingly constructed to try to optimize sensitivity and specificity (for each query, there are separate versions which optimize sensitivity and specificity) for particular information retrieval tasks in clinical medicine. Such a manual approach does not scale well, and in fact Aphinyanaphongs et al 2006 demonstrates that superior performance can be achieved automatically by machine learning SVM classifiers.

The basic difference between Aphinyanaphongs et al 2006 and the work in this thesis is that this thesis addresses information retrieval for proteomics identifiers while Aphinyanaphongs et al 2006 addresses information retrieval for specialized medical contexts. The key difference, however, is that this thesis demonstrates how specialized filters can be constructed automatically and easily, through a kind of symbiosis with the semantic web, at very large scale (i.e. for the millions of proteomics identifiers present in the LinkHub relational graph) using only a relatively small amount of information (i.e. the small number of web pages linked to relational subgraphs' identifier nodes versus a relatively large number of documents in manually created medical bibliographies) and relatively simple and computationally efficient methods (pre-IDF plus combined word weight vectors, which to create will have linear time complexity in the number of terms /

words, versus SVMs which are more computationally intensive and require the solution of quadratic programming optimization problems) while still achieving high performance. Nevertheless, the Aphinyanaphongs et al 2006 work is consistent with and supports the general approach taken in this thesis of creating specialized filters (in the form of word weight vectors) for retrieval of documents specific to particular proteomics identifiers, and demonstrates its effectiveness.

Another noteworthy result of the Aphinyanaphongs et al 2006 work is its evaluation of relevance metrics based on citations, such as citation count, journal impact factor, and Google's PageRank algorithm. As noted above, Google's great success was due in large part, at least initially, to its PageRank algorithm which provided an effective solution to the problem of relevance ranking of huge result sets. It would thus seem reasonable to expect that algorithms based on citation information such as PageRank would also prove very effective for information retrieval of the scientific literature, but the Aphinyanaphongs et al 2006 work finds this to not be the case. They conclude:

> "These experiments provide evidence that when building information retrieval filters focused on a retrieval task and corresponding gold standard, the filter models have to be built specifically for this task and gold standard. Under those conditions, machine learning filters outperform standard citation metrics. Furthermore, citation counts and impact factors add marginal value to discriminatory performance [*when used as additional features by the machine learning classifiers*]" (italicized text added for clarification).

In fact, the Aphinyanaphongs et al 2006 work did not directly compare to PageRank, but they cite a previous study [134] that showed citation count superior to PageRank, and since their study directly showed machine learning classifiers to outperform citation count they conclude by transitivity that machine learning classifiers are very likely superior to PageRank.

Given Google's great success with PageRank for ranking web documents, this result might seem counterintuitive. Aphinyanaphongs et al 2006 provides some intuition:

> "An article may cite another article for a variety of reasons: to acknowledge prior work, identify methodology, provide background reading, correct or criticize, substantiate claims, alert readers to forthcoming work, authenticate data, identify original publication of a term or concept, disclaim work of others, or dispute priority claims. In addition, the citing paper may be a comprehensive review that attempts to cite most recent papers on the topic, the reviewers may have recommended that a citation needs be included, the cited article may be a highly controversial or fashionable one, etc. An article citation thus may or may not endorse a cited article. The lack of an unambiguous connection between citation, context of use, manner of use, and/or endorsement prevents citation count from being a single effective measure of inclusion in an "importance" bibliography. More generally stated, the conceivable reasons for citation are so numerous that it is unrealistic to believe that citation conveys just one semantic interpretation. Instead citation metrics are a superimposition of a vast array of semantically distinct reasons to acknowledge an existing article. It follows that any specific set

of criteria cannot be captured by a few general citation metrics and only focused filtering mechanisms, if attainable, would be able to identify articles satisfying the specific criteria in question.

Another limitation of citation metrics is that they assume that the frequency of citations is uniform across all topics. This assumption is clearly not true across all topics in biomedicine. For example, the total number of citations using the query "breast cancer" in Pubmed returns 141,704 citations whereas the query "osteosarcomas" returns 15,904 articles (executed on 11/15/2005). Thus even the highest ranking article in osteosarcomas by citation count may not rank comparably to articles at lower ranks within breast cancer.

We also note that citation metrics are not only limited by their lack of focus, but, in general, they are not available until several years have passed. This reduces the usefulness of citation-based metrics for assessing cutting-edge articles. Since predicting future citation count is an open and unsolved problem in pattern recognition so far [note: *this work attempted such prediction unsuccessfully*], it follows that citation metrics are not only highly non-specific but also unavailable when needed the most (i.e., for articles published in recent years)."

In spite of this result, citation information still seems intuitively likely to be an important, useful source of information for relevance ranking. An important part of PageRank is that the link text is used as a concise snippet summary of the linked to page,

and citations in scientific papers generally do not provide such link text (or make it easily apparent what it would be); this is important, because the link text is propagated as highly weighted text to the linked to page, and this thus wouldn't be available in uses of PageRank on scientific literature. Also, at the extremes citation metrics would seem to be highly predictive of relevance, e.g. an article not cited at all (or very few times) is very likely less relevant than one cited a large number of times; thus, citation information might work better at a coarse-grained level or in discretized form. The observation that there is a large time-lag until citation information is available is valid, however possibly this could be taken advantage of: the temporal information about when and at what rate citations become available for an article might be useful for predicting relevance. In other words, for articles with the same number of citations, the temporal patterns of when those citations were received might be predictive (e.g. maybe an article that receives a flurry of citations in a short period in a hot area would be more relevant than one that received its citations at a steady rate). Next, consider that citations are just one way that documents can be linked together into a large graph. PubNet [135] allows one to do PubMed queries and view the results as a network, where the returned citations can be specified to be linked by co-authorship, common MeSH terms, shared location, or databank identifiers (PDB, GenBank, SwissProt). In the same way, one could run PageRank or consider other link-based metrics on such differently constructed literature networks; for example, PageRank on a co-authorship graph seems likely to be effective for predicting relevance (i.e. who we associate and work with influences the quality of our work). Finally, note that the issue about the opacity of semantic interpretations of citations is something that the semantic web could directly address. Given that the constructs for specifying links in

the semantic web are much richer and unambiguous, application of PageRank or other citation metrics to relevance ranking in the current web or scientific literature enhanced with semantic web links or citations (expressed in RDF or OWL format) would not encounter the problem of ambiguous semantic interpretation of links. Thus, while the results of Aphinyanaphongs et al 2006 regarding citation metrics are intriguing, it seems there is still room for fruitful explorations about how citation and linking information among documents could be used for improved relevance ranking.

# Appendix 1 Proteomics Overview, Proteomics Databases, and Representative Problems in Computational Proteomics

## *A1.1 Overview of Proteomics and Related Areas; or a (very) Crash Course in Modern Biology*

**Genomics** can be defined as the science of sequencing or spelling out each letter of the DNA molecule in an organism's genome, and identifying its genes (http://www.med.umich.edu/genetics/glossary/#g). The Human Genome Project [136] is likely the most well known large-scale biological research project within the last two decades to the general public. It was a large, US government funded scientific initiative coordinated by the National Institutes of Health and the Dept. of Energy involving many different researchers and labs whose aim was to determine the complete DNA base sequence of the human genome, which consists of 23 chromosome pairs and about 3 billion DNA bases in total, and to identify all the genes within it. The initial draft of the human genome was released and published in the journals Nature [137] and Science [138] in Feb 2001 (the Science paper actually described the sequence as determined by a private, competing project run by the company Celera Genomics, although it used data from the publicly funded project), and subsequent work has been ongoing to improve the accuracy of the human genome sequence. The human genome project has been covered in the mainstream news often and there is a popular belief that the human genome project was somehow the end-point and culmination of biological knowledge about humans. In

fact, while it is clearly a monumental achievement, it is really only the beginning of unraveling the complete biological workings of human beings (and similarly for other organisms whose genomes have been sequenced).

**Molecular biology** is the study of the structure and function of biological molecules, and the so-called **central dogma of molecular biology** was first enunciated by Francis Crick in 1958 and later restated more formally in a 1970 Nature paper [139]. It describes the basic flow of information in cells: DNA → RNA → protein. In other words, DNA makes RNA which makes proteins, which in turn facilitate the previous two steps as well as the replication of DNA. Research in the years since the central dogma was first put forth has complicated this simple picture, for example recent discoveries point to a much more complex and active role for RNA and splicing and post-translational modifications greatly increase the variety of protein products. Nevertheless, the central dogma is still essentially correct and provides the basic framework for understanding how cells work. The important point is that DNA is just the static, information encoding template for the dynamic operation of cells. For the most part, DNA is not an active molecule that actually carries out the necessary life supporting function of cells, such as moving molecules between subcellular compartments, catalyzing the reactions which create, store, and use energy, etc.; it is the proteins (and also somewhat RNA) which are the prime active agents of cell operation, and just knowing the genome sequence and genes does not help you much towards elucidating the dynamics of cells. Thus, to more fully understand how cells work we need to study the prime active agents, the proteins, and this is what proteomics does.

A good, concise definition of proteomics is as follows

(http://en.wikipedia.org/wiki/Proteomics):

> **Proteomics** is the large-scale study of proteins, particularly their structures and functions. This term was coined to make an analogy with genomics, and while it is often viewed as the "next step", proteomics is much more complicated than genomics. Most importantly, while the genome is a rather constant entity, the proteome differs from cell to cell and is constantly changing through its biochemical interactions with the genome and the environment. One organism has radically different protein expression in different parts of its body, in different stages of its life cycle and in different environmental conditions.
>
> The entirety of proteins in existence in an organism throughout its life cycle, or on a smaller scale the entirety of proteins found in a particular cell type under a particular type of stimulation, are referred to as the *proteome* of the organism or cell type respectively.

Proteomics is a key component of **systems biology**, whose goal is a holistic, systems-level understanding of biological systems that takes into account complex interactions of gene, protein, and cell elements (www.genpromag.com/Glossary~LETTER~S.html). Proteomics and systems biology, because they are trying to achieve a global understanding of the interactions among a huge number of interacting elements (genes, proteins, etc.), of necessity rely heavily on computational analysis using techniques from **bioinformatics**. Bioinformatics deals with the collection, organization and analysis of large amounts of biological data using networks of computers, databases and techniques from computer science, mathematics, and statistics

(http://www.abc.net.au/science/slab/genome2001/glossary.htm). Much data used in bioinformatics is the result of large-scale, high-throughput experiments which obtain large amounts of aggregate data about cell dynamics and function. For example, one commonly used type of such large-scale high-throughput data is gene expression data gotten from a **DNA microarray**, which is a collection of microscopic DNA spots attached to a solid surface, such as glass, plastic or silicon chip forming an array for the purpose of expression profiling, i.e. monitoring expression levels for thousands of genes simultaneously (http://en.wikipedia.org/wiki/DNA_microarray). Other commonly used high-throughput experimental techniques used in proteomics include **gel electrophoresis** (to separate a mixture of proteins and determine their relative masses and isoelectric points), **X-ray crystallography** and **nuclear magnetic resonance** (different experimental methods for determining the three dimensional structures of proteins), **mass spectrometry** (to find the composition of an unknown physical sample), and **Two-hybrid screening** (for experimentally determining potential protein-protein interactions), among others. Such high-throughput experimental techniques are generating a vast amount of biological data, most of which is publicly available.

## *A1.2 Proteomics Related Databases*

In addition to data from large-scale, high-throughput experiments a number of other kinds of protein data and information are collected in various web-accessible (and also often downloadable) databases. The following are some of the important large and well-known sources of proteomics data (almost all of these are present in LinkHub):

- **UniProt** (http://www.uniprot.org). UniProt is the ***uni***versal ***prot***ein database, a central repository of protein data that is the world's most comprehensive resource on protein information. It contains entries for most known proteins, and each entry has information such as originating gene names, source organism of the protein, literature citations relating to the protein, comments giving functional information and subcellular localization information (if known), the protein sequence and associated biochemical information, and, importantly, a large number of cross-references to other biological databases (in particular, almost all the others described next). UniProt, in fact, serves as core backbone content for the LinkHub system described in this thesis.
- **Pfam** (http://pfam.wustl.edu/). Protein families or domains are an important structuring principle for proteins; they are common sub-modules of protein sequences which are shared (in slightly modified form, but usually preserving some basic function) across many proteins in many organisms based on common evolution. Pfam is a main database for such protein families, and the Pfam website describes the Pfam database as follows: “Pfam is a database of multiple alignments of protein domains or conserved protein regions. The alignments represent some evolutionary conserved structure which has implications for the protein's function. Profile hidden Markov models (profile HMMs) built from the Pfam alignments can be very useful for automatically recognizing that a new protein belongs to an existing protein family, even if the homology is weak.” Pfam is another important piece of backbone content in the LinkHub system.

- **Gene Ontology or "GO"** (http://www.geneontology.org). GO is a standardized vocabulary in the form of a directed acyclic graph of terms (in order of more general to more specific) for describing 3 important properties of gene and protein function: the **molecular function** of gene products, their role in multi-step **biological processes**, and their localization to **cellular components**. GO is the most widely used standard for annotating gene and protein function, and many protein databases use GO terms to describe the functions of proteins (e.g. UniProt cross-references to GO terms in its protein entries).
- **Protein Data Bank** or "**PDB**" (http://www.pdb.org). The PDB is a repository for 3-D structural data of proteins and nucleic acids. This data, typically obtained by X-ray crystallography or NMR spectroscopy, is submitted by biologists and biochemists from around the world, is released into the public domain, and can be accessed for free. The database is the central repository for biological structural data (http://en.wikipedia.org/wiki/Protein_Data_Bank).
- **Structural Genomics Target Tracking Databases: TargetDB and PepcDB** (http://targetdb.pdb.org and http://pepcdb.pdb.org/). Protein 3-D structure is highly conserved through evolution, i.e. evolutionarily related proteins in distant species with mostly divergent sequences nevertheless often share very similar 3-D structures, and family members with even only a small amount of sequence similarity to another protein family member of known structure (down to around 20%) will have structures very close to that known protein's structure (and a technique called *homology modeling* is used to model them). Protein structure is the key indicator of protein function, so knowing the 3-D structures of proteins is

important. Predicting protein 3-D structure from only protein sequence data is currently intractable (and is a large open problem) so experimental techniques must be used. Structural Genomics is another government funded program involving a number of labs throughout the United States and also internationally. The main goal is to solve representative 3-D structures of all known protein families in a high-throughput, efficient manner. Protein targets of structural genomics go through an experimental pipeline of a number of stages en route to structure determination, and the target tracking databases provide status information (e.g. where in the pipeline a protein is, dates when it reached stages, etc.) on all targets; the endpoint of the pipeline is deposition of the protein's 3-D coordinates in the PDB. These target tracking databases are used to inform the public about structural genomics progress, as the basis of data mining and statistical analysis (an example is described below), and, importantly, to avoid wasting effort and resources on redundantly solving multiple structures from the same family when only one is necessary.

- **MolMovDB** (http://www.molmovdb.org). MolMovDB is the database of macromolecular motions. While the PDB contains protein 3-D structures, they are static structures. Proteins and other molecules in the cell are in fact flexible and have motions which are important for carrying out their functions. Molecular dynamics simulation based on a known structure is one technique that can explore likely motions of molecules, but it is very computationally intensive. MolMovDB takes proteins for which more than one structure is available, in different conformations, and calculates the most likely (least energy) trajectory between the

different structures; this motion will likely be close to the motion the protein actually takes in the cell.

- **Species specific databases: SGD (yeast), RGD (rat), MGD (mouse), WormBase (nematode worm)**. These are databases that provide much detailed information and data, including but not limited to proteomics, but are specific to important model organisms such as SGD for yeast (*Saccharomyces cerevisia*) at http://www.yeastgenome.org, RGD for the rat (*Rattus norvegicus*) at http://rgd.mcw.edu/, MGD for the mouse (*Mus musculus*) at http://www.informatics.jax.org, and WormBase for the worm (*Caenorhabditis elegans*) at http://www.wormbase.org/.

## *A1.3 Important Computational Problems in Proteomics*

The above databases and results from large-scale, high-throughput experiments can be used to aid in solving important computational problems in proteomics, and this section gives an overview of some of these problems. A common theme for many of these problems is to use known, experimentally determined attributes of certain proteins to predict these attributes and other, related attributes in other proteins using machine learning and statistics. One fundamental concept and technique in bioinformatics used to aid in solving these problems is **sequence alignment** which can be defined as:

> "A sequence alignment in bioinformatics is a way of arranging DNA, RNA, or protein primary sequences to emphasize their regions of similarity, which may indicate functional or evolutionary relationships between the genes or proteins in the query. Sequences are typically written with their characters (generally amino

acids or nucleotides) in aligned columns into which gaps are inserted so that successive columns contain identical or similar characters." (http://en.wikipedia.org/wiki/Sequence_alignment).

Figure A1 is an example of a sequence alignment between two proteins (two zinc finger proteins which are highly related and this is shown by the many matching columns in the alignment). Sequence alignment is not limited to just considering exact column matches, but also considers the chemical similarity of nucleotides or amino acids. For example, an alignment column with two non-equal but chemically similar amino acids signals more similarity between the sequences than two chemically distinct aligned amino acids. There exist scoring matrices which give numerical scores to different aligned nucleotides or amino acids, and an entire sequence alignment can be scored by summing up the scores (as given by the scoring matrix) of all alignment columns. Optimal sequence alignment can be performed using algorithms based on dynamic programming, but they are computationally intensive [140, 141].

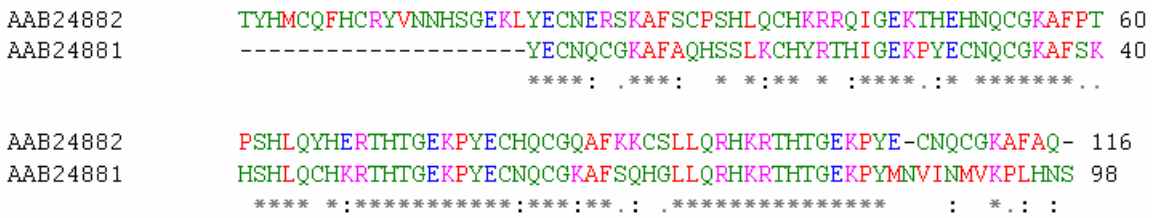



**Fig. A1. A sequence alignment, produced by the sequence alignment program ClustalW, between two human zinc finger proteins identified by GenBank accession number**. Figure obtained from http://en.wikipedia.org/wiki/Sequence_alignment.

An important application of sequence alignment is to use it to perform sequence database search, i.e. determine the sequences in a large sequence database that most

closely match a given query sequence and order them in decreasing order of similarity. Two widely used heuristic tools for doing this almost as good as dynamic programming based algorithms but much more efficiently are **BLAST** (Basic Local Alignment Search Tool) [142] and **FASTA** [143]. BLAST and FASTA essentially support a simple "nearest neighbor" approach to many bioinformatics problems. In particular, if you want to determine some attribute X which you don't know for a new protein, one simple thing you can do which often works well is to determine the new protein's closest, most related sequences in a sequence database by performing a Blast or Fasta search of that sequence database; you can then assign the value of attribute X of the top match (or majority vote of the top matches). Note that this assumes there exist in the sequence database similar enough sequences which have known values for attribute X. A similar but more refined technique is to construct a **phylogenetic tree** (which is a tree reflecting, ideally, the likely evolutionary history and divergences among a set of **homologous**, meaning evolutionarily related, sequences) and use the structure of the tree and where the query sequence falls within it to determine the value of the attribute X for the query sequence (i.e. the values for other sequences in small subtrees that also contain the query sequence are better predictors). Finally, some important generally accepted observations relating protein structure, function, and sequence are: (1) similar structure implies similar function but not necessarily detectable similarity in sequence, (2) similar sequence implies similar structure, which in turn thus implies similar function. In general, homologous proteins can have very divergent sequences and still have similar structures. Thus, the homology of two proteins could be undetectable by sequence alignment methods while the structures are still very similar. This is one of the reasons structural

genomics is important --- it extends the reach of such "nearest neighbor" methods since what are separate families at the sequence level can be combined into larger, combined structural families if their structures are determined to be homologous. Thus, e.g., attributes which were unknown in one of the families can possibly be inferred from the values of the attributes in the other family if known.

With this overview of some basic techniques used to solve computational problems in biology, some of the important computational problems in proteomics are:

- **Protein Function Prediction**. Given a protein of unknown function, predict what its function is. This can be done using the "nearest neighbor" sequence database search technique --- assign the functions of the nearest neighbors to the query sequence. Terms from the Gene Ontology, because it is a standard and has been widely accepted and used, are generally the terms used to annotate function.
- **Protein subcellular localization prediction**. Most eukaryotic proteins are encoded in the nucleus and synthesized in the cytosol, and are then transported to various subcellular compartments (golgi-apparatus, endoplasmic-reticulum, cell membrane, etc.) to do their work (where they may interact with other proteins in the same compartment). So the problem is to predict where in the cell a protein is localized. This is practically useful for determining which proteins might interact, and for drug targeting (so you can try to target your drug to proteins in a more easily accessible location in the cell, such as secreted proteins and plasma

membrane proteins which are in the extracellular space and cell membrane respectively). Again, nearest neighbor database search techniques can be used, but proteins also often have small sequence patterns within them that serve as sorting and localization signals to the cellular apparatus responsible for transporting proteins, and these can be searched for in novel proteins to help predict their subcellular localizations.

- **Protein structure prediction**. In fact, there are 2 important levels of structure: **secondary structure** refers to common local structural motifs that occur in all proteins, with the two main ones being alpha helices and beta sheets, whereas **tertiary structure** refers to the overall, global 3-D structure of a protein. Computational methods are generally very good at predicting secondary structure, up to about 90% overall prediction accuracy. Machine learning based on statistics about the propensities of amino acids to be in different types of secondary structure motifs, hidden markov models trained from known secondary structure of proteins, along with information about homologs' known secondary structure in corresponding positions in multiple sequence alignments of homologs (determined from sequence database searches) and the query sequence are all used to solve the secondary structure prediction problem very well. Tertiary structure prediction, on the other hand, is very difficult (currently intractable) and is one of the important open problems of computational biology that is actively being researched. The most practical method,

which does work reasonably well, is homology modeling as described briefly above --- if you know the structure of a protein that is similar in sequence to a query protein (i.e., in the same family as described above) you can model the structure of the query protein based on the known structure of the sequence similar protein; unfortunately, this method has limited applicability because relatively few structures are currently known compared to the number of known protein sequences. Ab initio protein structure prediction attempts to predict 3-D structure "from scratch" and generally involves some kind of heuristic search for the lowest energy conformation of a protein; current ab initio techniques generally do not work well.

- **Protein-protein interaction prediction**. The cell is complex, with a large number of different kinds of molecules, including many different proteins, packed tightly in very close proximity. Most proteins do not work in isolation but interact with other proteins to perform their functions, for example interactions are important in understanding intra-cellular signaling networks; predicting protein-protein interactions is thus an important computational problem. Interactions between pairs of proteins can be inferred experimentally using such high-throughput techniques as two-hybrid systems, affinity purification / mass spectrometry assays, or from protein microarrays, however these experimental techniques produce noisy results with many false positives. Computational methods thus play a key role in trying to refine the

experimental predictions to eliminate false positives. Bayesian integration, which combines different kinds of indirect evidence for interaction such as phylogenetic profiling (interacting proteins tend to co-evolve and measures of phylogenetic distance thus can provide some evidence for interaction), known interacting homologues in other species ("interologs"), and the presence of structural interaction motifs, is commonly used for this refinement process.

- **Protein interaction network analysis**. Once you are able to accurately uncover large numbers of protein-protein interactions from experimental and computational techniques, you can form a large interaction network out of all of them (where nodes are proteins and edges signify that the two nodes connected by the edge interact). You can then perform network analysis of the interaction network to learn more about overall cellular function --- interaction network analysis is an important step towards a systems biology understanding of the cell. For example, you can look for hubs (nodes with large degree) --- these will presumably be very important (so called "essential") proteins (since they interact with so many others) which would cause organism death upon removal (e.g. in a gene knockout experiment). In addition, cliques in the network will likely represent protein complexes.

# Appendix 2 RDF Schema of LinkHub RDF Structure

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<rdfs:Class rdf:ID="identifier_types">
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:label>Identifier types</rdfs:label>
</rdfs:Class>
<rdf:Property rdf:ID="identifier_types_type_name">
        <rdfs:domain rdf:resource="#identifier_types" />
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal" />
</rdf:Property>
<rdfs:Class rdf:ID="identifiers">
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:label>Identifiers</rdfs:label>
</rdfs:Class>
<rdf:Property rdf:ID="identifiers_type">
        <rdfs:domain rdf:resource="#identifiers" />
        <rdfs:range rdf:resource="#identifier_types" />
</rdf:Property>
<rdf:Property rdf:ID="identifiers_id">
        <rdfs:domain rdf:resource="#identifiers" />
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal" />
</rdf:Property>
<rdf:Property rdf:ID="mappings_type_synonym">
        <rdfs:domain rdf:resource="#identifiers" />
        <rdfs:range rdf:resource="#identifiers" />
</rdf:Property>
<rdf:Property rdf:ID="mappings_type_Family_Mapping">
        <rdfs:domain rdf:resource="#identifiers" />
        <rdfs:range rdf:resource="#identifiers" />
</rdf:Property>
<rdfs:Class rdf:ID="resource">
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:label>Resource</rdfs:label>
</rdfs:Class>
<rdf:Property rdf:ID="resource_name">
        <rdfs:domain rdf:resource="#resource" />
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal" />
</rdf:Property>
<rdf:Property rdf:ID="resource_description">
        <rdfs:domain rdf:resource="#resource" />
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal" />
```

```
</rdf:Property>
<rdf:Property rdf:ID="resource_url_template">
        <rdfs:domain rdf:resource="#resource" />
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal" />
</rdf:Property>
<rdfs:Class rdf:ID="resource_groups">
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:label>Resource groups</rdfs:label>
</rdfs:Class>
<rdf:Property rdf:ID="resource_group">
        <rdfs:domain rdf:resource="#resource" />
        <rdfs:range rdf:resource="#resource_groups" />
</rdf:Property>
<rdfs:Class rdf:ID="resource_accepts">
        <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:label>Resource accepts</rdfs:label>
</rdfs:Class>
<rdf:Property rdf:ID="resource_accept">
        <rdfs:domain rdf:resource="#resource" />
        <rdfs:range rdf:resource="#resource_accepts" />
</rdf:Property>
<rdf:Property rdf:ID="resource_accepts_id_acc_type">
        <rdfs:domain rdf:resource="#resource_accepts" />
        <rdfs:range rdf:resource="#identifier_types" />
</rdf:Property>
<rdf:Property rdf:ID="resource_accepts_except_type">
        <rdfs:domain rdf:resource="#resource_accepts" />
        <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal" />
</rdf:Property>
<rdf:Property rdf:ID="link_exception">
        <rdfs:domain rdf:resource="#resource" />
        <rdfs:range rdf:resource="#identifiers" />
</rdf:Property>
</rdf:RDF>
```

# Appendix 3 Full SeRQL statements for example joined YeastHub / LinkHub queries of chapter 4.

### Query 1: Finding Worm 'Interologs' of Yeast Protein Interactions

```
SELECT DISTINCT YeastProtein1, YeastProtein2, WormProtein1,
WormProtein2
FROM
{ppi}      it:Protein1                     {YeastProtein1},
{lhYO1}    lh:identifiers_id               {YeastProtein1},
{lhYO1}    lh:identifiers_type             {lhYOType},
{lhYO1}    lh:mappings_type_synonym        {lhUP1a},
{lhUP1a}   lh:identifiers_type             {lhUPType},
{lhUP1a}   lh:mappings_type_Family_Mapping {lhPFAM1},
{lhPFAM1}  lh:identifiers_type             {lhPFType},
{lhPFAM1}  lh:mappings_type_Family_Mapping {lhUP1b},
{lhUP1b}   lh:identifiers_type             {lhUPType},
{lhUP1b}   lh:mappings_type_synonym        {lhWO1},
{lhWO1}    lh:identifiers_type             {lhWOType},
{lhWO1}    lh:identifiers_id               {WormProtein1},
{ppi}      it:Protein2                     {YeastProtein2},
{lhYO2}    lh:identifiers_id               {YeastProtein2},
{lhYO2}    lh:identifiers_type             {lhYOType},
{lhYO2}    lh:mappings_type_synonym        {lhUP2a},
{lhUP2a}   lh:identifiers_type             {lhUPType},
{lhUP2a}   lh:mappings_type_Family_Mapping {lhPFAM2},
{lhPFAM2}  lh:identifiers_type             {lhPFType},
{lhPFAM2}  lh:mappings_type_Family_Mapping {lhUP2b},
{lhUP2b}   lh:identifiers_type             {lhUPType},
{lhUP2b}   lh:mappings_type_synonym        {lhWO2},
{lhWO2}    lh:identifiers_type             {lhWOType},
{lhWO2}    lh:identifiers_id               {WormProtein2},
{lhYOType} lh:identifier_types_type_name   {YEAST_ORF},
{lhUPType} lh:identifier_types_type_name   {UNIPROT_KB_ACC},
{lhPFType} lh:identifier_types_type_name   {PFAM_ACC},
{lhWOType} lh:identifier_types_type_name   {WORMBASE}
WHERE
YeastProtein1   = "YAL005C" AND
YeastProtein2   = "YLR310C" AND
YEAST_ORF       = "YEAST_ORF" AND
(UNIPROT_KB_ACC = "UniProtKB/Swiss-Prot Acc" OR
 UNIPROT_KB_ACC = "UniProtKB/TrEMBL Acc") AND
PFAM_ACC        = "PFAM_ACC" AND
WORMBASE        = "WORMBASE"
USING NAMESPACE
it=<http://yeasthub2.gersteinlab.org/yeasthub/schema/the_platinum_stand
ard_for_ppi20060224234451_schema.rdf>,
lh=<http://yeasthub2.gersteinlab.org/yeasthub/datasets/manual_upload/li
```

```
nkhub_schema.rdf#>
```

## Query 2: Exploring Pseudogene Content versus Gene Essentiality in Yeast and Humans

### Yeast

```
SELECT DISTINCT YeastORF, Pseudogene
FROM
{gene}     mips:ORF                        {YeastORF},
{lhYO}     lh:identifiers_id               {YeastORF},
{lhYO}     lh:identifiers_type             {lhYOType},
{lhYOType} lh:identifier_types_type_name {YEAST_ORF},
{lhYO}     lh:mappings_type_synonym        {lhUP},
{lhUP}     lh:identifiers_type             {lhUPType},
{lhUPType} lh:identifier_types_type_name {UNIPROT_KB_ACC},
{lhUP}     lh:mappings_type_synonym        {lhPG},
{lhPG}     lh:identifiers_id               {Pseudogene},
{lhPG}     lh:identifiers_type             {lhPGType},
{lhPGType} lh:identifier_types_type_name {YEAST_PGENE}
WHERE
YEAST_ORF       = "YEAST_ORF" AND
(UNIPROT_KB_ACC = "UniProtKB/Swiss-Prot Acc" OR
 UNIPROT_KB_ACC = "UniProtKB/TrEMBL Acc") AND
YEAST_PGENE     = "YEAST_PGENE"
USING NAMESPACE
mips=<http://yeasthub2.gersteinlab.org/yeasthub/schema/mips_lethal_gene
s20050608191535_schema.rdf>,
lh=<http://yeasthub2.gersteinlab.org/yeasthub/datasets/manual_upload/li
nkhub_schema.rdf#>
```

### Humans

```
SELECT DISTINCT YeastORF, HumanGene, Pseudogene
FROM
{gene}       mips:ORF                          {YeastORF},
{lhYO}       lh:identifiers_id                 {YeastORF},
{lhYO}       lh:identifiers_type               {lhYOType},
{lhYOType}   lh:identifier_types_type_name     {YEAST_ORF},
{lhYO}       lh:mappings_type_synonym          {lhUP1},
{lhUP1}      lh:identifiers_type               {lhUPType},
{lhUPType}   lh:identifier_types_type_name     {UNIPROT_KB_ACC},
{lhUP1}      lh:mappings_type_Family_Mapping   {lhPFAM},
{lhPFAM}     lh:identifiers_type               {lhPFAMType},
{lhPFAMType} lh:identifier_types_type_name     {PFAM_ACC},
{lhPFAM}     lh:mappings_type_Family_Mapping   {lhUP2},
{lhUP2}      lh:identifiers_type               {lhUPType},
{lhUP2}      lh:mappings_type_synonym          {lhUPI2},
{lhUPI2}     lh:identifiers_type               {lhUPIType},
{lhUPIType}  lh:identifier_types_type_name     {UNIPROT_KB_ID},
```

```
{lhUPI2}     lh:identifiers_id               {HumanGene},
{lhUP2}      lh:mappings_type_synonym        {lhPG},
{lhPG}       lh:identifiers_id               {Pseudogene},
{lhPG}       lh:identifiers_type             {lhPGType},
{lhPGType}   lh:identifier_types_type_name   {PGENE_LSID}
WHERE
YEAST_ORF       = "YEAST_ORF" AND
(UNIPROT_KB_ACC = "UniProtKB/Swiss-Prot Acc" OR
 UNIPROT_KB_ACC = "UniProtKB/TrEMBL Acc") AND
PFAM_ACC        = "PFAM_ACC" AND
(UNIPROT_KB_ID  = "UniProtKB/Swiss-Prot Id" OR
 UNIPROT_KB_ID  = "UniProtKB/TrEMBL Id") AND
HumanGene LIKE "*_HUMAN" AND
PGENE_LSID      = "PGENE_LSID"
USING NAMESPACE
mips=<http://yeasthub2.gersteinlab.org/yeasthub/schema/mips_lethal_gene
s20050608191535_schema.rdf>,
lh=<http://yeasthub2.gersteinlab.org/yeasthub/datasets/manual_upload/li
nkhub_schema.rdf#>
```

# Appendix 4 Results of PubMed search for UniProt P26364

**Base searches (in stemmed form) pulling in the most relevant PubMed citations on average and the top 20 overall results for a search of PubMed for citations relevant to UniProt P26364, which is a yeast adenylate kinase protein located in the mitochondrion, using the procedure described in section 5.4. The abstract, article title, chemical substances, and mesh headings sections were searched by the base searches with Swish-e.**

```
------Searches pulling in most relevant documents------
------sorted by avg normalized relevance score---------
map     0.154435058145367
kinase-encoding 0.151404184861974
cerevisia       0.144666758112637
saccharomyc     0.144648868257687
activ   0.136552545139489
treu    0.136447542238716
kinas   0.134769703960043
pak     0.134308190670839
drosophila      0.132469983327981
et      0.126286215243481
annot   0.123239734650015
align   0.122659618072846
blastp  0.120825688941596
fasta   0.120612893193172
sequenc 0.118222123395145
domain  0.116745685521193
ncbi    0.116200439267413
motif   0.115793435831974
zinc-finger     0.113569963802525
famili  0.113250448501724
thaliana        0.111608246730794
protein 0.111601173336311
homolog 0.106563175722906
integr  0.105720626409793
melanogast      0.105625362704043
embl    0.105574546312823
respons 0.103101451293047
adk     0.10250808652045
mitochondri     0.10248265587763
```

```
brucei  0.102000406074174
genom   0.101806710802482
chromosom       0.101684177283082
bind    0.100714396525705
trypanosoma     0.100688068343041
genomic_dna     0.0988185817626534
schricker       0.0986706020168536
nucleotide-binding      0.0983984054921733
bacillu 0.09784546039425
structur        0.0961442185091403
region  0.0950146751626199
gtp     0.0917005276028915
ak      0.0909596175236286
restrict        0.0899755269403705
orf     0.0892193828521216
mammalian       0.0886708095322505
dna     0.0872712064718214
site    0.0851759887816613
phosphotransferas       0.0845221245578372
put     0.0841266830399347
adenyl  0.0821743515306913
basic   0.0820972500925573
tair    0.0813874650221222
express 0.080391991882234
mip     0.0800093007602122
associ  0.079499582106676
databas 0.079046151064703
gene    0.0765910373094701
translat        0.0760619751305515
sativa  0.076002682419131
yeast   0.0759914230540377
catalyt 0.0744015050479549
mai     0.0735939329034745
p-loop  0.0734476845411023
atp     0.072140480284921
membran 0.0716559307245259
pir     0.0715983896271713
mitochondrion   0.0713704324090332
…
…

------RANKED RESULTS------
------RESULT 1------
Pmid: 8496185
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=8496185&dopt=Abstract
Journal: The Journal of biological chemistry.
Article Title: Molecular analysis of the essential gene for adenylate
kinase from the fission yeast Schizosaccharomyces pombe.
Year: 1993
Year: May
Year: 25
Searches :
        adk ( 20 )
Relevance scores: 0.346312134114352 (normalized), 0.346312134114352
(raw)
```

------RESULT 2------
Pmid: 8537371
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=8537371&dopt=Abstract
Journal: The Journal of biological chemistry.
Article Title: Strain-dependent occurrence of functional GTP:AMP phosphotransferase (AK3) in Saccharomyces cerevisiae.
Year: 1995
Year: Dec
Year: 29
Searches :
        ak ( 46 )
        pak ( 25 )
        aki ( 36 )
        adk ( 125 )
Relevance scores: 0.336086274632716 (normalized), 0.336086274632716 (raw)
------RESULT 3------
Pmid: 8439550
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=8439550&dopt=Abstract
Journal: Biochimica et biophysica acta.
Article Title: Identification and characterization of a yeast gene encoding an adenylate kinase homolog.
Year: 1993
Year: Feb
Year: 20
Searches :
        adk ( 6 )
Relevance scores: 0.335616794642753 (normalized), 0.335616794642753 (raw)
------RESULT 4------
Pmid: 2199332
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=2199332&dopt=Abstract
Journal: Gene.
Article Title: Isolation of a novel protein kinase-encoding gene from yeast by oligodeoxyribonucleotide probing.
Year: 1990
Year: May
Year: 31
Searches :
        kinase-encoding ( 173 )
Relevance scores: 0.328380689806839 (normalized), 0.328380689806839 (raw)
------RESULT 5------
Pmid: 7483841
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=7483841&dopt=Abstract
Journal: Yeast (Chichester, England)
Article Title: Sequence analysis of a 33.1 kb fragment from the left arm of Saccharomyces cerevisiae chromosome X, including putative

proteins with leucine zippers, a fungal Zn(II)2-Cys6 binuclear cluster domain and a putative alpha 2-SCB-alpha 2 binding site.
Year: 1995
Year: Jun
Year: 15
Searches :
put ( 22 )
Relevance scores: 0.323175507765449 (normalized), 0.323175507765449 (raw)
------RESULT 6------
Pmid: 10620778
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=10620778&dopt=Abstract
Journal: Yeast (Chichester, England)
Article Title: Isolation and sequence of the HOG1 homologue from Debaryomyces hansenii by complementation of the hog1Delta strain of Saccharomyces cerevisiae.
Year: 2000
Year: Jan
Year: 15
Searches :
cerevisia ( 198 )
saccharomyc ( 111 )
Relevance scores: 0.322399541780529 (normalized), 0.322399541780529 (raw)
------RESULT 7------
Pmid: 1587477
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1587477&dopt=Abstract
Journal: Gene.
Article Title: A putative second adenylate kinase-encoding gene from the yeast Saccharomyces cerevisiae.
Year: 1992
Year: May
Year: 1
Searches :
adk ( 3 )
Relevance scores: 0.322140449905575 (normalized), 0.322140449905575 (raw)
------RESULT 8------
Pmid: 8515773
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=8515773&dopt=Abstract
Journal: Molecular and biochemical parasitology.
Article Title: A Trypanosoma brucei gene family encoding protein kinases with catalytic domains structurally related to Nek1 and NIMA.
Year: 1993
Year: May
Searches :
kinase-encoding ( 78 )
Relevance scores: 0.319459805585679 (normalized), 0.319459805585679 (raw)
------RESULT 9------
Pmid: 14681421

Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=14681421&dopt=Abstract
Journal: Nucleic acids research.
Article Title: Saccharomyces Genome Database (SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms.
Year: 2004
Year: Jan
Year: 1
Searches :
sgd ( 14 )
saccharomyc ( 11 )
Relevance scores: 0.319087654473805 (normalized), 0.319087654473805 (raw)
------RESULT 10------
Pmid: 7926721
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=7926721&dopt=Abstract
Journal: Genes & development.
Article Title: The Doa locus encodes a member of a new protein kinase family and is essential for eye and embryonic development in Drosophila melanogaster.
Year: 1994
Year: May
Year: 15
Searches :
kinase-encoding ( 108 )
Relevance scores: 0.312693121259224 (normalized), 0.312693121259224 (raw)
------RESULT 11------
Pmid: 2022322
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=2022322&dopt=Abstract
Journal: Gene.
Article Title: Structural and functional conservation between the high-affinity K+ transporters of Saccharomyces uvarum and Saccharomyces cerevisiae.
Year: 1991
Year: Mar
Year: 1
Searches :
saccharomyc ( 7 )
Relevance scores: 0.306951025980308 (normalized), 0.306951025980308 (raw)
------RESULT 12------
Pmid: 1774787
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1774787&dopt=Abstract
Journal: Journal of molecular evolution.
Article Title: Evolution of the dec-1 eggshell locus in Drosophila. I. Restriction site mapping and limited sequence comparison in the melanogaster species subgroup.
Year: 1991

Year: Oct
Searches :
melanogast ( 33 )
Relevance scores: 0.305639230551114 (normalized), 0.305639230551114 (raw)
------RESULT 13------
Pmid: 1620094
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1620094&dopt=Abstract
Journal: Molecular & general genetics : MGG.
Article Title: A new member of the adenylate kinase family in yeast: PAK3 is highly homologous to mammalian AK3 and is targeted to mitochondria.
Year: 1992
Year: Jun
Searches :
ak ( 65 )
pak ( 7 )
Relevance scores: 0.30517396996688 (normalized), 0.30517396996688 (raw)
------RESULT 14------
Pmid: 2834085
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=2834085&dopt=Abstract
Journal: Current genetics.
Article Title: Chromosomal mapping of the uracil permease gene of Saccharomyces cerevisiae.
Year: 1986
Searches :
saccharomyc ( 194 )
Relevance scores: 0.302237799784997 (normalized), 0.302237799784997 (raw)
------RESULT 15------
Pmid: 10021364
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=10021364&dopt=Abstract
Journal: Current biology : CB.
Article Title: A Drosophila TNF-receptor-associated factor (TRAF) binds the ste20 kinase Misshapen and activates Jun kinase.
Year: 1999
Year: Jan
Year: 28
Searches :
drosophila ( 175 )
Relevance scores: 0.301733915533769 (normalized), 0.301733915533769 (raw)
------RESULT 16------
Pmid: 8017101
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=8017101&dopt=Abstract
Journal: Yeast (Chichester, England)
Article Title: Sequence comparison of the ARG4 chromosomal regions from the two related yeasts, Saccharomyces cerevisiae and Saccharomyces douglasii.

Year: 1994
Year: Mar
Searches :
saccharomyc ( 6 )
Relevance scores: 0.301032581068433 (normalized), 0.301032581068433 (raw)
------RESULT 17------
Pmid: 2656692
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=2656692&dopt=Abstract
Journal: The Journal of biological chemistry.
Article Title: Molecular cloning of Saccharomyces cerevisiae CDC6 gene. Isolation, identification, and sequence analysis.
Year: 1989
Year: May
Year: 25
Searches :
saccharomyc ( 29 )
Relevance scores: 0.300906980668241 (normalized), 0.300906980668241 (raw)
------RESULT 18------
Pmid: 1729597
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1729597&dopt=Abstract
Journal: Molecular and cellular biology.
Article Title: Dominant mutations in a gene encoding a putative protein kinase (BCK1) bypass the requirement for a Saccharomyces cerevisiae protein kinase C homolog.
Year: 1992
Year: Jan
Searches :
kinase-encoding ( 30 )
Relevance scores: 0.300575255605413 (normalized), 0.300575255605414 (raw)
------RESULT 19------
Pmid: 1592264
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=1592264&dopt=Abstract
Journal: Genes & development.
Article Title: Interaction of murine ets-1 with GGA-binding sites establishes the ETS domain as a new DNA-binding motif.
Year: 1992
Year: Jun
Searches :
et ( 21 )
Relevance scores: 0.298612573458432 (normalized), 0.298612573458431 (raw)
------RESULT 20------
Pmid: 15550393
Link to PubMed:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&list_uids=15550393&dopt=Abstract
Journal: The Journal of biological chemistry.

Article Title: Activation of p21-activated kinase 6 by MAP kinase kinase 6 and p38 MAP kinase.
Year: 2005
Year: Feb
Year: 4
Searches :
map ( 62 )
Relevance scores: 0.297846716340132 (normalized), 0.297846716340132 (raw)

# Appendix 5 Average .05 and 1.0 AUC values for TrEMBL proteins

Average .05 and 1.0 AUC values (sorted descending by AUC value, then descending by perc Features Kept) for 100 randomly sampled TrEMBL proteins and for different values for the four parameters *GO Wt, PFAM Wt, Perc Features Kept,* and *pre-IDF applied.*

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | .05 AUC |
|---|---|---|---|---|
| 0 | 0.2 | 1 | 1 | 0.03226933 |
| 0 | 0.2 | 0.9 | 1 | 0.03226933 |
| 0 | 0.2 | 0.8 | 1 | 0.03226933 |
| 0 | 0.2 | 0.7 | 1 | 0.03226933 |
| 0 | 0.2 | 0.6 | 1 | 0.03226933 |
| 0 | 0.2 | 0.5 | 1 | 0.03226933 |
| 0 | 0.2 | 0.4 | 1 | 0.03226905 |
| 0 | 0.2 | 0.3 | 1 | 0.03226719 |
| 0 | 0.2 | 0.2 | 1 | 0.03225249 |
| 0 | 0.1 | 1 | 1 | 0.03219563 |
| 0 | 0.1 | 0.9 | 1 | 0.03219563 |
| 0 | 0.1 | 0.8 | 1 | 0.03219563 |
| 0 | 0.1 | 0.7 | 1 | 0.03219563 |
| 0 | 0.1 | 0.6 | 1 | 0.03219563 |
| 0 | 0.1 | 0.5 | 1 | 0.03219563 |
| 0 | 0.1 | 0.4 | 1 | 0.03219556 |
| 0 | 0.1 | 0.3 | 1 | 0.03219438 |
| 0 | 0.2 | 0.1 | 1 | 0.03218699 |
| 0 | 0.1 | 0.2 | 1 | 0.03217805 |
| 0 | 0.1 | 0.1 | 1 | 0.03213521 |
| 0 | 0.3 | 1 | 1 | 0.03212749 |
| 0 | 0.3 | 0.9 | 1 | 0.03212749 |
| 0 | 0.3 | 0.8 | 1 | 0.03212749 |
| 0 | 0.3 | 0.7 | 1 | 0.03212749 |
| 0 | 0.3 | 0.6 | 1 | 0.03212749 |
| 0 | 0.3 | 0.5 | 1 | 0.03212749 |
| 0 | 0.3 | 0.4 | 1 | 0.03212723 |
| 0 | 0.3 | 0.3 | 1 | 0.03212499 |
| 0 | 0.3 | 0.2 | 1 | 0.03210133 |
| 0 | 0.3 | 0.1 | 1 | 0.03199315 |
| 0 | 0.4 | 0.3 | 1 | 0.03190595 |
| 0 | 0.4 | 1 | 1 | 0.03190503 |
| 0 | 0.4 | 0.9 | 1 | 0.03190503 |

| | | | | |
|---|---|---|---|---|
| 0 | 0.4 | 0.8 | 1 | 0.03190503 |
| 0 | 0.4 | 0.7 | 1 | 0.03190503 |
| 0 | 0.4 | 0.6 | 1 | 0.03190503 |
| 0 | 0.4 | 0.5 | 1 | 0.03190503 |
| 0 | 0.4 | 0.4 | 1 | 0.03190463 |
| 0 | 0.4 | 0.2 | 1 | 0.03187975 |
| 0 | 0.4 | 0.1 | 1 | 0.03178714 |
| 0 | 0.5 | 1 | 1 | 0.03166559 |
| 0 | 0.5 | 0.9 | 1 | 0.03166559 |
| 0 | 0.5 | 0.8 | 1 | 0.03166559 |
| 0 | 0.5 | 0.7 | 1 | 0.03166559 |
| 0 | 0.5 | 0.6 | 1 | 0.03166559 |
| 0 | 0.5 | 0.5 | 1 | 0.03166559 |
| 0 | 0.5 | 0.4 | 1 | 0.03166495 |
| 0 | 0.5 | 0.3 | 1 | 0.03166376 |
| 0 | 0.5 | 0.2 | 1 | 0.03163302 |
| 0 | 0.5 | 0.1 | 1 | 0.03148492 |
| 0 | 0.6 | 1 | 1 | 0.03139155 |
| 0 | 0.6 | 0.9 | 1 | 0.03139155 |
| 0 | 0.6 | 0.8 | 1 | 0.03139155 |
| 0 | 0.6 | 0.7 | 1 | 0.03139155 |
| 0 | 0.6 | 0.6 | 1 | 0.03139155 |
| 0 | 0.6 | 0.5 | 1 | 0.03139155 |
| 0 | 0.6 | 0.4 | 1 | 0.03139096 |
| 0 | 0.6 | 0.3 | 1 | 0.03139022 |
| 0 | 0.6 | 0.2 | 1 | 0.03136572 |
| 0 | 0 | 1 | 1 | 0.03131128 |
| 0 | 0 | 0.9 | 1 | 0.03131128 |
| 0 | 0 | 0.8 | 1 | 0.03131128 |
| 0 | 0 | 0.7 | 1 | 0.03131128 |
| 0 | 0 | 0.6 | 1 | 0.03131128 |
| 0 | 0 | 0.5 | 1 | 0.03131128 |
| 0 | 0 | 0.4 | 1 | 0.03131128 |
| 0 | 0 | 0.3 | 1 | 0.03131128 |
| 0 | 0 | 0.2 | 1 | 0.03131105 |
| 0 | 0 | 0.1 | 1 | 0.03130795 |
| 0 | 0.6 | 0.1 | 1 | 0.03119433 |
| 0.1 | 0 | 0.3 | 1 | 0.0287276 |
| 0.1 | 0 | 1 | 1 | 0.02872664 |
| 0.1 | 0 | 0.9 | 1 | 0.02872664 |
| 0.1 | 0 | 0.8 | 1 | 0.02872664 |
| 0.1 | 0 | 0.7 | 1 | 0.02872664 |
| 0.1 | 0 | 0.6 | 1 | 0.02872664 |
| 0.1 | 0 | 0.4 | 1 | 0.02872663 |
| 0.1 | 0 | 0.5 | 1 | 0.02872662 |
| 0.1 | 0 | 0.2 | 1 | 0.02872459 |
| 0.1 | 0 | 0.1 | 1 | 0.02865226 |
| 0.2 | 0 | 0.3 | 1 | 0.02694185 |

| | | | | |
|---|---|---|---|---|
| 0.2 | 0 | 0.4 | 1 | 0.02693719 |
| 0.2 | 0 | 0.5 | 1 | 0.02693667 |
| 0.2 | 0 | 1 | 1 | 0.02693666 |
| 0.2 | 0 | 0.9 | 1 | 0.02693666 |
| 0.2 | 0 | 0.8 | 1 | 0.02693666 |
| 0.2 | 0 | 0.7 | 1 | 0.02693666 |
| 0.2 | 0 | 0.6 | 1 | 0.02693666 |
| 0.2 | 0 | 0.2 | 1 | 0.02693259 |
| 0.2 | 0 | 0.1 | 1 | 0.02681958 |
| 0.3 | 0 | 0.3 | 1 | 0.02566895 |
| 0.3 | 0 | 0.4 | 1 | 0.02566634 |
| 0.3 | 0 | 1 | 1 | 0.0256653 |
| 0.3 | 0 | 0.9 | 1 | 0.0256653 |
| 0.3 | 0 | 0.8 | 1 | 0.0256653 |
| 0.3 | 0 | 0.7 | 1 | 0.0256653 |
| 0.3 | 0 | 0.6 | 1 | 0.0256653 |
| 0.3 | 0 | 0.5 | 1 | 0.02566521 |
| 0.3 | 0 | 0.2 | 1 | 0.02565918 |
| 0.3 | 0 | 0.1 | 1 | 0.02544624 |
| 0.4 | 0 | 0.3 | 1 | 0.02478174 |
| 0.4 | 0 | 0.4 | 1 | 0.02477942 |
| 0.4 | 0 | 0.5 | 1 | 0.02477875 |
| 0.4 | 0 | 1 | 1 | 0.02477874 |
| 0.4 | 0 | 0.9 | 1 | 0.02477874 |
| 0.4 | 0 | 0.8 | 1 | 0.02477874 |
| 0.4 | 0 | 0.7 | 1 | 0.02477874 |
| 0.4 | 0 | 0.6 | 1 | 0.02477874 |
| 0.4 | 0 | 0.2 | 1 | 0.02475805 |
| 0.4 | 0 | 0.1 | 1 | 0.02447707 |
| 0.5 | 0 | 0.4 | 1 | 0.02414431 |
| 0.5 | 0 | 0.5 | 1 | 0.02414315 |
| 0.5 | 0 | 1 | 1 | 0.02414312 |
| 0.5 | 0 | 0.9 | 1 | 0.02414312 |
| 0.5 | 0 | 0.8 | 1 | 0.02414312 |
| 0.5 | 0 | 0.7 | 1 | 0.02414312 |
| 0.5 | 0 | 0.6 | 1 | 0.02414312 |
| 0.5 | 0 | 0.3 | 1 | 0.02414224 |
| 0.5 | 0 | 0.1 | 1 | 0.02383403 |
| 0 | 0.6 | 1 | 0 | 0.02053266 |
| 0 | 0.6 | 0.9 | 0 | 0.02053266 |
| 0 | 0.6 | 0.8 | 0 | 0.02053266 |
| 0 | 0.6 | 0.7 | 0 | 0.02053266 |
| 0 | 0.6 | 0.6 | 0 | 0.02053119 |
| 0 | 0.6 | 0.5 | 0 | 0.02052796 |
| 0 | 0.6 | 0.4 | 0 | 0.02051924 |
| 0 | 0.5 | 0.7 | 0 | 0.02051916 |
| 0 | 0.5 | 1 | 0 | 0.02051909 |
| 0 | 0.5 | 0.9 | 0 | 0.02051909 |

| | | | | |
|---|---|---|---|---|
| 0 | 0.5 | 0.8 | 0 | 0.02051909 |
| 0 | 0.5 | 0.6 | 0 | 0.02051803 |
| 0 | 0.5 | 0.5 | 0 | 0.02051378 |
| 0 | 0.5 | 0.4 | 0 | 0.02050484 |
| 0 | 0.4 | 1 | 0 | 0.02044174 |
| 0 | 0.4 | 0.9 | 0 | 0.02044174 |
| 0 | 0.4 | 0.8 | 0 | 0.02044174 |
| 0 | 0.4 | 0.7 | 0 | 0.02044128 |
| 0 | 0.6 | 0.3 | 0 | 0.02044075 |
| 0 | 0.4 | 0.6 | 0 | 0.0204403 |
| 0 | 0.4 | 0.4 | 0 | 0.02043515 |
| 0 | 0.4 | 0.5 | 0 | 0.02043253 |
| 0 | 0.5 | 0.3 | 0 | 0.02041186 |
| 0 | 0.4 | 0.3 | 0 | 0.02037361 |
| 0 | 0.3 | 1 | 0 | 0.02025335 |
| 0 | 0.3 | 0.9 | 0 | 0.02025335 |
| 0 | 0.3 | 0.8 | 0 | 0.02025335 |
| 0 | 0.3 | 0.7 | 0 | 0.02025293 |
| 0 | 0.3 | 0.6 | 0 | 0.02025241 |
| 0 | 0.3 | 0.5 | 0 | 0.02024226 |
| 0 | 0.3 | 0.4 | 0 | 0.02024189 |
| 0 | 0.5 | 0.2 | 0 | 0.02020292 |
| 0 | 0.6 | 0.2 | 0 | 0.02020221 |
| 0 | 0.3 | 0.3 | 0 | 0.02017879 |
| 0 | 0.4 | 0.2 | 0 | 0.02009239 |
| 0 | 0.3 | 0.2 | 0 | 0.019908 |
| 0 | 0.2 | 0.6 | 0 | 0.01986638 |
| 0 | 0.2 | 1 | 0 | 0.01986622 |
| 0 | 0.2 | 0.9 | 0 | 0.01986622 |
| 0 | 0.2 | 0.8 | 0 | 0.01986622 |
| 0 | 0.2 | 0.7 | 0 | 0.0198658 |
| 0 | 0.2 | 0.5 | 0 | 0.01986351 |
| 0 | 0.2 | 0.4 | 0 | 0.01984901 |
| 0 | 0.2 | 0.3 | 0 | 0.01980921 |
| 0 | 0.6 | 0.1 | 0 | 0.0196964 |
| 0 | 0.5 | 0.1 | 0 | 0.01960129 |
| 0 | 0.2 | 0.2 | 0 | 0.01955873 |
| 0 | 0.4 | 0.1 | 0 | 0.01949362 |
| 0 | 0.3 | 0.1 | 0 | 0.01921435 |
| 0 | 0.1 | 1 | 0 | 0.01917276 |
| 0 | 0.1 | 0.9 | 0 | 0.01917276 |
| 0 | 0.1 | 0.8 | 0 | 0.01917276 |
| 0 | 0.1 | 0.7 | 0 | 0.01917234 |
| 0 | 0.1 | 0.6 | 0 | 0.01917223 |
| 0 | 0.1 | 0.5 | 0 | 0.01917195 |
| 0 | 0.1 | 0.4 | 0 | 0.01916157 |
| 0 | 0.1 | 0.3 | 0 | 0.01912103 |
| 0 | 0.1 | 0.2 | 0 | 0.01895125 |

| | | | | |
|---|---|---|---|---|
| 0 | 0.2 | 0.1 | 0 | 0.01885197 |
| 0 | 0.1 | 0.1 | 0 | 0.0182605 |
| 0 | 0 | 1 | 0 | 0.01791437 |
| 0 | 0 | 0.9 | 0 | 0.01791437 |
| 0 | 0 | 0.8 | 0 | 0.01791437 |
| 0 | 0 | 0.7 | 0 | 0.01791437 |
| 0 | 0 | 0.6 | 0 | 0.01791437 |
| 0 | 0 | 0.5 | 0 | 0.01791437 |
| 0 | 0 | 0.4 | 0 | 0.0179143 |
| 0 | 0 | 0.3 | 0 | 0.01790658 |
| 0 | 0 | 0.2 | 0 | 0.01782088 |
| 0 | 0 | 0.1 | 0 | 0.01709111 |
| 0.1 | 0 | 0.7 | 0 | 0.01005948 |
| 0.1 | 0 | 1 | 0 | 0.01005943 |
| 0.1 | 0 | 0.9 | 0 | 0.01005943 |
| 0.1 | 0 | 0.8 | 0 | 0.01005943 |
| 0.1 | 0 | 0.6 | 0 | 0.01005877 |
| 0.1 | 0 | 0.5 | 0 | 0.0100574 |
| 0.1 | 0 | 0.4 | 0 | 0.01004995 |
| 0.1 | 0 | 0.3 | 0 | 0.0100246 |
| 0.1 | 0 | 0.2 | 0 | 0.00991317 |
| 0.1 | 0 | 0.1 | 0 | 0.00959619 |
| 0.2 | 0 | 1 | 0 | 0.00827719 |
| 0.2 | 0 | 0.9 | 0 | 0.00827719 |
| 0.2 | 0 | 0.8 | 0 | 0.00827719 |
| 0.2 | 0 | 0.7 | 0 | 0.00827719 |
| 0.2 | 0 | 0.6 | 0 | 0.00827617 |
| 0.2 | 0 | 0.5 | 0 | 0.00827482 |
| 0.2 | 0 | 0.4 | 0 | 0.00827342 |
| 0.2 | 0 | 0.3 | 0 | 0.00823758 |
| 0.2 | 0 | 0.2 | 0 | 0.00819601 |
| 0.2 | 0 | 0.1 | 0 | 0.00792511 |
| 0.3 | 0 | 0.7 | 0 | 0.00759724 |
| 0.3 | 0 | 1 | 0 | 0.00759714 |
| 0.3 | 0 | 0.9 | 0 | 0.00759714 |
| 0.3 | 0 | 0.8 | 0 | 0.00759714 |
| 0.3 | 0 | 0.6 | 0 | 0.00759539 |
| 0.3 | 0 | 0.5 | 0 | 0.00759173 |
| 0.3 | 0 | 0.4 | 0 | 0.0075853 |
| 0.3 | 0 | 0.3 | 0 | 0.00754955 |
| 0.3 | 0 | 0.2 | 0 | 0.00750707 |
| 0.3 | 0 | 0.1 | 0 | 0.00730218 |
| 0.4 | 0 | 1 | 0 | 0.00722004 |
| 0.4 | 0 | 0.9 | 0 | 0.00722004 |
| 0.4 | 0 | 0.8 | 0 | 0.00722004 |
| 0.4 | 0 | 0.7 | 0 | 0.00722004 |
| 0.4 | 0 | 0.6 | 0 | 0.00721857 |
| 0.4 | 0 | 0.5 | 0 | 0.0072152 |

| | | | | |
|---|---|---|---|---|
| 0.4 | 0 | 0.4 | 0 | 0.0072041 |
| 0.4 | 0 | 0.3 | 0 | 0.00717755 |
| 0.4 | 0 | 0.2 | 0 | 0.00714179 |
| 0.5 | 0 | 0.7 | 0 | 0.00699412 |
| 0.5 | 0 | 1 | 0 | 0.0069941 |
| 0.5 | 0 | 0.9 | 0 | 0.0069941 |
| 0.5 | 0 | 0.8 | 0 | 0.0069941 |
| 0.5 | 0 | 0.6 | 0 | 0.00699256 |
| 0.5 | 0 | 0.5 | 0 | 0.0069906 |
| 0.5 | 0 | 0.4 | 0 | 0.00697842 |
| 0.4 | 0 | 0.1 | 0 | 0.00696236 |
| 0.5 | 0 | 0.3 | 0 | 0.00695857 |
| 0.5 | 0 | 0.1 | 0 | 0.00670413 |

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | 1.0 AUC |
|---|---|---|---|---|
| 0 | 0.1 | 1 | 1 | 0.927418459 |
| 0 | 0.1 | 0.9 | 1 | 0.927418459 |
| 0 | 0.1 | 0.8 | 1 | 0.927418459 |
| 0 | 0.1 | 0.7 | 1 | 0.927418459 |
| 0 | 0.1 | 0.6 | 1 | 0.927418459 |
| 0 | 0.1 | 0.5 | 1 | 0.927418459 |
| 0 | 0.1 | 0.4 | 1 | 0.927399173 |
| 0 | 0.1 | 0.3 | 1 | 0.927380207 |
| 0 | 0.2 | 1 | 1 | 0.927225042 |
| 0 | 0.2 | 0.9 | 1 | 0.927225042 |
| 0 | 0.2 | 0.8 | 1 | 0.927225042 |
| 0 | 0.2 | 0.7 | 1 | 0.927225042 |
| 0 | 0.2 | 0.6 | 1 | 0.927225042 |
| 0 | 0.2 | 0.5 | 1 | 0.927225042 |
| 0 | 0.2 | 0.4 | 1 | 0.927201928 |
| 0 | 0.2 | 0.3 | 1 | 0.927167249 |
| 0 | 0.1 | 0.2 | 1 | 0.92715958 |
| 0 | 0.2 | 0.2 | 1 | 0.926720119 |
| 0 | 0.1 | 0.1 | 1 | 0.926307698 |
| 0 | 0.3 | 1 | 1 | 0.925988632 |
| 0 | 0.3 | 0.9 | 1 | 0.925988632 |
| 0 | 0.3 | 0.8 | 1 | 0.925988632 |
| 0 | 0.3 | 0.7 | 1 | 0.925988632 |
| 0 | 0.3 | 0.6 | 1 | 0.925988632 |
| 0 | 0.3 | 0.5 | 1 | 0.925988632 |
| 0 | 0.3 | 0.3 | 1 | 0.92598823 |
| 0 | 0.3 | 0.4 | 1 | 0.925973219 |
| 0 | 0.2 | 0.1 | 1 | 0.925318643 |
| 0 | 0.3 | 0.2 | 1 | 0.925314409 |
| 0 | 0.4 | 1 | 1 | 0.924511333 |
| 0 | 0.4 | 0.9 | 1 | 0.924511333 |
| 0 | 0.4 | 0.8 | 1 | 0.924511333 |
| 0 | 0.4 | 0.7 | 1 | 0.924511333 |

| | | | | |
|---|---|---|---|---|
| 0 | 0.4 | 0.6 | 1 | 0.924511333 |
| 0 | 0.4 | 0.5 | 1 | 0.924511333 |
| 0 | 0.4 | 0.3 | 1 | 0.924496517 |
| 0 | 0.4 | 0.4 | 1 | 0.92449225 |
| 0 | 0.4 | 0.2 | 1 | 0.923874846 |
| 0 | 0.5 | 1 | 1 | 0.922956837 |
| 0 | 0.5 | 0.9 | 1 | 0.922956837 |
| 0 | 0.5 | 0.8 | 1 | 0.922956837 |
| 0 | 0.5 | 0.7 | 1 | 0.922956837 |
| 0 | 0.5 | 0.6 | 1 | 0.922956837 |
| 0 | 0.5 | 0.5 | 1 | 0.922956837 |
| 0 | 0.5 | 0.3 | 1 | 0.922950402 |
| 0 | 0.5 | 0.4 | 1 | 0.922936315 |
| 0 | 0.3 | 0.1 | 1 | 0.922854303 |
| 0 | 0.5 | 0.2 | 1 | 0.922728792 |
| 0 | 0.4 | 0.1 | 1 | 0.921697851 |
| 0 | 0.6 | 0.3 | 1 | 0.921435344 |
| 0 | 0.6 | 1 | 1 | 0.921412749 |
| 0 | 0.6 | 0.9 | 1 | 0.921412749 |
| 0 | 0.6 | 0.8 | 1 | 0.921412749 |
| 0 | 0.6 | 0.7 | 1 | 0.921412749 |
| 0 | 0.6 | 0.6 | 1 | 0.921412749 |
| 0 | 0.6 | 0.5 | 1 | 0.921412749 |
| 0 | 0.6 | 0.4 | 1 | 0.92139326 |
| 0 | 0.6 | 0.2 | 1 | 0.921210287 |
| 0 | 0 | 1 | 1 | 0.920262344 |
| 0 | 0 | 0.9 | 1 | 0.920262344 |
| 0 | 0 | 0.8 | 1 | 0.920262344 |
| 0 | 0 | 0.7 | 1 | 0.920262344 |
| 0 | 0 | 0.6 | 1 | 0.920262344 |
| 0 | 0 | 0.5 | 1 | 0.920262344 |
| 0 | 0 | 0.4 | 1 | 0.920262344 |
| 0 | 0 | 0.3 | 1 | 0.920262344 |
| 0 | 0 | 0.2 | 1 | 0.920258842 |
| 0 | 0 | 0.1 | 1 | 0.920091065 |
| 0 | 0.5 | 0.1 | 1 | 0.919956702 |
| 0 | 0.6 | 0.1 | 1 | 0.917813697 |
| 0.1 | 0 | 0.2 | 1 | 0.915287299 |
| 0.1 | 0 | 0.1 | 1 | 0.915249164 |
| 0.1 | 0 | 0.3 | 1 | 0.915048147 |
| 0.1 | 0 | 1 | 1 | 0.915046856 |
| 0.1 | 0 | 0.9 | 1 | 0.915046856 |
| 0.1 | 0 | 0.8 | 1 | 0.915046856 |
| 0.1 | 0 | 0.7 | 1 | 0.915046856 |
| 0.1 | 0 | 0.6 | 1 | 0.915046856 |
| 0.1 | 0 | 0.5 | 1 | 0.915046448 |
| 0.1 | 0 | 0.4 | 1 | 0.915042976 |
| 0.2 | 0 | 0.2 | 1 | 0.907042642 |

| | | | | |
|---|---|---|---|---|
| 0.2 | 0 | 0.1 | 1 | 0.906995415 |
| 0.2 | 0 | 1 | 1 | 0.906583199 |
| 0.2 | 0 | 0.9 | 1 | 0.906583199 |
| 0.2 | 0 | 0.8 | 1 | 0.906583199 |
| 0.2 | 0 | 0.7 | 1 | 0.906583199 |
| 0.2 | 0 | 0.6 | 1 | 0.906583199 |
| 0.2 | 0 | 0.5 | 1 | 0.906582235 |
| 0.2 | 0 | 0.4 | 1 | 0.906575252 |
| 0.2 | 0 | 0.3 | 1 | 0.90655842 |
| 0.3 | 0 | 0.2 | 1 | 0.900802124 |
| 0.3 | 0 | 0.3 | 1 | 0.900317219 |
| 0.3 | 0 | 1 | 1 | 0.900310623 |
| 0.3 | 0 | 0.9 | 1 | 0.900310623 |
| 0.3 | 0 | 0.8 | 1 | 0.900310623 |
| 0.3 | 0 | 0.7 | 1 | 0.900310623 |
| 0.3 | 0 | 0.6 | 1 | 0.900310623 |
| 0.3 | 0 | 0.5 | 1 | 0.900309838 |
| 0.3 | 0 | 0.4 | 1 | 0.900306975 |
| 0.3 | 0 | 0.1 | 1 | 0.899705373 |
| 0.4 | 0 | 0.2 | 1 | 0.89588402 |
| 0.4 | 0 | 0.3 | 1 | 0.895466647 |
| 0.4 | 0 | 1 | 1 | 0.895458373 |
| 0.4 | 0 | 0.9 | 1 | 0.895458373 |
| 0.4 | 0 | 0.8 | 1 | 0.895458373 |
| 0.4 | 0 | 0.7 | 1 | 0.895458373 |
| 0.4 | 0 | 0.6 | 1 | 0.895458373 |
| 0.4 | 0 | 0.5 | 1 | 0.895458079 |
| 0.4 | 0 | 0.4 | 1 | 0.895451159 |
| 0.4 | 0 | 0.1 | 1 | 0.893990864 |
| 0.5 | 0 | 1 | 1 | 0.891547562 |
| 0.5 | 0 | 0.9 | 1 | 0.891547562 |
| 0.5 | 0 | 0.8 | 1 | 0.891547562 |
| 0.5 | 0 | 0.7 | 1 | 0.891547562 |
| 0.5 | 0 | 0.6 | 1 | 0.891547562 |
| 0.5 | 0 | 0.5 | 1 | 0.891547123 |
| 0.5 | 0 | 0.4 | 1 | 0.891543714 |
| 0.5 | 0 | 0.3 | 1 | 0.891471505 |
| 0.5 | 0 | 0.1 | 1 | 0.889662395 |
| 0 | 0.4 | 0.4 | 0 | 0.860175925 |
| 0 | 0.4 | 1 | 0 | 0.860168342 |
| 0 | 0.4 | 0.9 | 0 | 0.860168342 |
| 0 | 0.4 | 0.8 | 0 | 0.860168342 |
| 0 | 0.4 | 0.7 | 0 | 0.860167313 |
| 0 | 0.4 | 0.6 | 0 | 0.860155259 |
| 0 | 0.3 | 0.4 | 0 | 0.860148874 |
| 0 | 0.4 | 0.5 | 0 | 0.860134119 |
| 0 | 0.3 | 1 | 0 | 0.860090882 |
| 0 | 0.3 | 0.9 | 0 | 0.860090882 |

| | | | | |
|---|---|---|---|---|
| 0 | 0.3 | 0.8 | 0 | 0.860090882 |
| 0 | 0.3 | 0.7 | 0 | 0.860088424 |
| 0 | 0.3 | 0.6 | 0 | 0.860077572 |
| 0 | 0.3 | 0.5 | 0 | 0.860034918 |
| 0 | 0.5 | 0.7 | 0 | 0.859733596 |
| 0 | 0.5 | 1 | 0 | 0.85973313 |
| 0 | 0.5 | 0.9 | 0 | 0.85973313 |
| 0 | 0.5 | 0.8 | 0 | 0.85973313 |
| 0 | 0.5 | 0.6 | 0 | 0.859720947 |
| 0 | 0.5 | 0.5 | 0 | 0.859685219 |
| 0 | 0.5 | 0.4 | 0 | 0.859571983 |
| 0 | 0.3 | 0.3 | 0 | 0.859487188 |
| 0 | 0.4 | 0.3 | 0 | 0.859334849 |
| 0 | 0.2 | 0.4 | 0 | 0.859156266 |
| 0 | 0.2 | 1 | 0 | 0.859123408 |
| 0 | 0.2 | 0.9 | 0 | 0.859123408 |
| 0 | 0.2 | 0.8 | 0 | 0.859123408 |
| 0 | 0.2 | 0.7 | 0 | 0.859120455 |
| 0 | 0.2 | 0.6 | 0 | 0.859116082 |
| 0 | 0.2 | 0.5 | 0 | 0.859100624 |
| 0 | 0.6 | 0.7 | 0 | 0.859015099 |
| 0 | 0.6 | 1 | 0 | 0.859014732 |
| 0 | 0.6 | 0.9 | 0 | 0.859014732 |
| 0 | 0.6 | 0.8 | 0 | 0.859014732 |
| 0 | 0.6 | 0.6 | 0 | 0.858999012 |
| 0 | 0.6 | 0.5 | 0 | 0.858957891 |
| 0 | 0.6 | 0.4 | 0 | 0.858758813 |
| 0 | 0.2 | 0.3 | 0 | 0.858698273 |
| 0 | 0.5 | 0.3 | 0 | 0.858221514 |
| 0 | 0.6 | 0.3 | 0 | 0.857490193 |
| 0 | 0.1 | 0.4 | 0 | 0.85637968 |
| 0 | 0.1 | 1 | 0 | 0.856335328 |
| 0 | 0.1 | 0.9 | 0 | 0.856335328 |
| 0 | 0.1 | 0.8 | 0 | 0.856335328 |
| 0 | 0.1 | 0.7 | 0 | 0.856333924 |
| 0 | 0.1 | 0.6 | 0 | 0.856330397 |
| 0 | 0.1 | 0.5 | 0 | 0.856327404 |
| 0 | 0.3 | 0.2 | 0 | 0.856273781 |
| 0 | 0.2 | 0.2 | 0 | 0.856214732 |
| 0 | 0.1 | 0.3 | 0 | 0.855795543 |
| 0 | 0.4 | 0.2 | 0 | 0.855183156 |
| 0 | 0.5 | 0.2 | 0 | 0.854868523 |
| 0 | 0.1 | 0.2 | 0 | 0.853912353 |
| 0 | 0.6 | 0.2 | 0 | 0.853545315 |
| 0 | 0 | 1 | 0 | 0.849313752 |
| 0 | 0 | 0.9 | 0 | 0.849313752 |
| 0 | 0 | 0.8 | 0 | 0.849313752 |
| 0 | 0 | 0.7 | 0 | 0.849313752 |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0.6 | 0 | 0.849313752 |
| 0 | 0 | 0.5 | 0 | 0.849313752 |
| 0 | 0 | 0.4 | 0 | 0.849313752 |
| 0 | 0 | 0.3 | 0 | 0.849239801 |
| 0 | 0 | 0.2 | 0 | 0.84844112 |
| 0 | 0.5 | 0.1 | 0 | 0.846103585 |
| 0 | 0.4 | 0.1 | 0 | 0.846096159 |
| 0 | 0.6 | 0.1 | 0 | 0.845719287 |
| 0 | 0.3 | 0.1 | 0 | 0.844952441 |
| 0 | 0.2 | 0.1 | 0 | 0.844545819 |
| 0 | 0.1 | 0.1 | 0 | 0.843920554 |
| 0 | 0 | 0.1 | 0 | 0.837379538 |
| 0.1 | 0 | 1 | 0 | 0.798235804 |
| 0.1 | 0 | 0.9 | 0 | 0.798235804 |
| 0.1 | 0 | 0.8 | 0 | 0.798235804 |
| 0.1 | 0 | 0.7 | 0 | 0.798235292 |
| 0.1 | 0 | 0.6 | 0 | 0.798227836 |
| 0.1 | 0 | 0.5 | 0 | 0.79818086 |
| 0.1 | 0 | 0.4 | 0 | 0.798057907 |
| 0.1 | 0 | 0.3 | 0 | 0.79745182 |
| 0.1 | 0 | 0.2 | 0 | 0.794274812 |
| 0.1 | 0 | 0.1 | 0 | 0.786972726 |
| 0.2 | 0 | 1 | 0 | 0.774103461 |
| 0.2 | 0 | 0.9 | 0 | 0.774103461 |
| 0.2 | 0 | 0.8 | 0 | 0.774103461 |
| 0.2 | 0 | 0.7 | 0 | 0.774102998 |
| 0.2 | 0 | 0.6 | 0 | 0.774087745 |
| 0.2 | 0 | 0.5 | 0 | 0.774000066 |
| 0.2 | 0 | 0.4 | 0 | 0.773825807 |
| 0.2 | 0 | 0.3 | 0 | 0.772839322 |
| 0.2 | 0 | 0.2 | 0 | 0.770281592 |
| 0.2 | 0 | 0.1 | 0 | 0.763802265 |
| 0.3 | 0 | 1 | 0 | 0.761292369 |
| 0.3 | 0 | 0.9 | 0 | 0.761292369 |
| 0.3 | 0 | 0.8 | 0 | 0.761292369 |
| 0.3 | 0 | 0.7 | 0 | 0.761292072 |
| 0.3 | 0 | 0.6 | 0 | 0.761273574 |
| 0.3 | 0 | 0.5 | 0 | 0.761196197 |
| 0.3 | 0 | 0.4 | 0 | 0.76094147 |
| 0.3 | 0 | 0.3 | 0 | 0.759592558 |
| 0.3 | 0 | 0.2 | 0 | 0.757211099 |
| 0.4 | 0 | 1 | 0 | 0.753255644 |
| 0.4 | 0 | 0.9 | 0 | 0.753255644 |
| 0.4 | 0 | 0.8 | 0 | 0.753255644 |
| 0.4 | 0 | 0.7 | 0 | 0.75325525 |
| 0.4 | 0 | 0.6 | 0 | 0.75323626 |
| 0.4 | 0 | 0.5 | 0 | 0.753168826 |
| 0.4 | 0 | 0.4 | 0 | 0.752626448 |

| | | | | |
|---|---|---|---|---|
| 0.4 | 0 | 0.3 | 0 | 0.751476699 |
| 0.3 | 0 | 0.1 | 0 | 0.750840502 |
| 0.4 | 0 | 0.2 | 0 | 0.749353571 |
| 0.5 | 0 | 1 | 0 | 0.747748232 |
| 0.5 | 0 | 0.9 | 0 | 0.747748232 |
| 0.5 | 0 | 0.8 | 0 | 0.747748232 |
| 0.5 | 0 | 0.7 | 0 | 0.747747713 |
| 0.5 | 0 | 0.6 | 0 | 0.747728178 |
| 0.5 | 0 | 0.5 | 0 | 0.747642659 |
| 0.5 | 0 | 0.4 | 0 | 0.747060672 |
| 0.5 | 0 | 0.3 | 0 | 0.746006021 |
| 0.4 | 0 | 0.1 | 0 | 0.743414865 |
| 0.5 | 0 | 0.1 | 0 | 0.736150694 |

# Appendix 6 Results of paired t-tests for testing significance of results for TrEMBL from chapter 5.

The t-tests were done using Microsoft Excel 2003's Data Analysis add-in. For both .05 AUC and 1.0 AUC we demonstrate (with a figure showing the output of the paired t-test below each bullet point below) that the means are statistically significantly different for the below pairs of conditions. The P-value to look at is the one-tail value since the hypothesis is that one condition will improve the AUC score (i.e. result in a statistically different mean AUC value).

- PFAM and GO weights of 0, percent features kept of 0.5; comparing no pre-IDF step versus pre-IDF step, with the hypothesis that the pre-IDF step will improve the .05 AUC score. This is borne out by the highly significant one-tail P-value of 1.82527E-31.

| | *no pre-IDF* | *pre-IDF* |
|---|---|---|
| Mean | 0.017914374 | 0.03131128 |
| Variance | 0.00010387 | 0.000112774 |
| Observations | 100 | 100 |
| Pearson Correlation | 0.714657717 | |
| Hypothesized Mean Difference | 0 | |
| df | 99 | |
| t Stat | -17.02114064 | |
| P(T<=t) one-tail | 1.82527E-31 | |
| t Critical one-tail | 1.660391157 | |
| P(T<=t) two-tail | 3.65054E-31 | |
| t Critical two-tail | 1.9842169 | |

- GO weight of 0, percent features kept of 0.5 and pre-IDF step; comparing PFAM weight of 0 to PFAM weight of 0.2, with the hypothesis that the weight of 0.2

will improve the .05 AUC score. Again, the hypothesis is borne out but at a smaller (but still highly significant) one-tail P-value of 0.001744804.

| | *PFAM 0* | *PFAM 0.2* |
|---|---|---|
| Mean | 0.03131128 | 0.032269328 |
| Variance | 0.000112774 | 0.00011118 |
| Observations | 100 | 100 |
| Pearson Correlation | 0.954268603 | |
| Hypothesized Mean Difference | 0 | |
| df | 99 | |
| t Stat | -2.992853593 | |
| P(T<=t) one-tail | 0.001744804 | |
| t Critical one-tail | 1.660391157 | |
| P(T<=t) two-tail | 0.003489608 | |
| t Critical two-tail | 1.9842169 | |

- No pre-IDF step, GO weight of 0, percent features kept of 0.5; comparing PFAM weight of 0 to PFAM weight of 0.6, with the hypothesis that the weight of 0.6 will improve the .05 AUC score. This is borne out by the significant one-tail P-value of 1.80004E-05.

| | *PFAM 0* | *PFAM 0.6* |
|---|---|---|
| Mean | 0.017914374 | 0.020527959 |
| Variance | 0.00010387 | 0.000117757 |
| Observations | 100 | 100 |
| Pearson Correlation | 0.837114435 | |
| Hypothesized Mean Difference | 0 | |
| df | 99 | |
| t Stat | -4.3281505 | |
| P(T<=t) one-tail | 1.80004E-05 | |
| t Critical one-tail | 1.660391157 | |
| P(T<=t) two-tail | 3.60008E-05 | |
| t Critical two-tail | 1.9842169 | |

- PFAM and GO weights of 0, percent features kept of 0.5; comparing no pre-IDF step versus pre-IDF step, with the hypothesis that the pre-IDF step will improve

the 1.0 AUC score. This is borne out by the highly significant one-tail P-value of 1.07767E-24.

| | *no pre-IDF* | *pre-IDF* |
|---|---|---|
| Mean | 0.849313752 | 0.920262344 |
| Variance | 0.008145876 | 0.004643865 |
| Observations | 100 | 100 |
| Pearson Correlation | 0.818618518 | |
| Hypothesized Mean Difference | 0 | |
| df | 99 | |
| t Stat | -13.60388477 | |
| P(T<=t) one-tail | 1.07767E-24 | |
| t Critical one-tail | 1.660391157 | |
| P(T<=t) two-tail | 2.15533E-24 | |
| t Critical two-tail | 1.9842169 | |

- GO weight of 0, percent features kept of 0.5 and pre-IDF step; comparing PFAM weight of 0 to PFAM weight of 0.1, with the hypothesis that the weight of 0.1 will improve the 1.0 AUC score. The hypothesis is borne out at a smaller (but still highly significant) one-tail P-value of 0.00091023.

| | *PFAM 0* | *PFAM 0.1* |
|---|---|---|
| Mean | 0.920262344 | 0.927418459 |
| Variance | 0.004643865 | 0.003610441 |
| Observations | 100 | 100 |
| Pearson Correlation | 0.947039754 | |
| Hypothesized Mean Difference | 0 | |
| df | 99 | |
| t Stat | -3.204617438 | |
| P(T<=t) one-tail | 0.00091023 | |
| t Critical one-tail | 1.660391157 | |
| P(T<=t) two-tail | 0.001820459 | |
| t Critical two-tail | 1.9842169 | |

- No pre-IDF step, GO weight of 0, percent features kept of 0.5; comparing PFAM weight of 0 to PFAM weight of 0.4, with the hypothesis that the weight of 0.4

will improve the 1.0 AUC score. This is borne out by the significant one-tail P-value of 0.002952176.

.

| | *PFAM 0* | *PFAM 0.4* |
|---|---|---|
| Mean | 0.849313752 | 0.860134119 |
| Variance | 0.008145876 | 0.008609263 |
| Observations | 100 | 100 |
| Pearson Correlation | 0.912098936 | |
| Hypothesized Mean Difference | 0 | |
| df | 99 | |
| t Stat | -2.813913422 | |
| P(T<=t) one-tail | 0.002952176 | |
| t Critical one-tail | 1.660391157 | |
| P(T<=t) two-tail | 0.005904352 | |
| t Critical two-tail | 1.9842169 | |

# Appendix 7 Average .05 and 1.0 AUC values for Swiss-Prot proteins

Average .05 and 1.0 AUC values (sorted descending by AUC value) for 200 randomly sampled Swiss-Prot proteins and for different values for the three parameters *GO Wt, PFAM Wt,* and *pre-IDF applied.* A constant value of 0.5 was chosen for *Perc Features Kept*.

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | .05 AUC |
|---|---|---|---|---|
| 0 | 0.1 | 0.5 | 1 | 0.035710069 |
| 0 | 0 | 0.5 | 1 | 0.035667273 |
| 0 | 0.2 | 0.5 | 1 | 0.035598894 |
| 0 | 0.3 | 0.5 | 1 | 0.035449905 |
| 0 | 0.4 | 0.5 | 1 | 0.035273387 |
| 0 | 0.5 | 0.5 | 1 | 0.035075739 |
| 0 | 0.6 | 0.5 | 1 | 0.034880836 |
| 0.1 | 0 | 0.5 | 1 | 0.033101119 |
| 0.2 | 0 | 0.5 | 1 | 0.03152662 |
| 0.3 | 0 | 0.5 | 1 | 0.03052458 |
| 0.4 | 0 | 0.5 | 1 | 0.029841108 |
| 0.5 | 0 | 0.5 | 1 | 0.029324578 |
| 0 | 0.2 | 0.5 | 0 | 0.025253218 |
| 0 | 0.3 | 0.5 | 0 | 0.02521937 |
| 0 | 0.1 | 0.5 | 0 | 0.025167611 |
| 0 | 0.4 | 0.5 | 0 | 0.02511875 |
| 0 | 0.5 | 0.5 | 0 | 0.024973051 |
| 0 | 0 | 0.5 | 0 | 0.024920735 |
| 0 | 0.6 | 0.5 | 0 | 0.024822486 |
| 0.1 | 0 | 0.5 | 0 | 0.019269518 |
| 0.2 | 0 | 0.5 | 0 | 0.01633386 |
| 0.3 | 0 | 0.5 | 0 | 0.01486524 |
| 0.4 | 0 | 0.5 | 0 | 0.014003045 |
| 0.5 | 0 | 0.5 | 0 | 0.013448272 |

| GO Wt | PFAM Wt | Perc Features Kept | pre-IDF applied? | 1.0 AUC |
|---|---|---|---|---|
| 0 | 0.1 | 0.5 | 1 | 0.950538504 |
| 0 | 0 | 0.5 | 1 | 0.950253129 |
| 0 | 0.2 | 0.5 | 1 | 0.950104292 |
| 0 | 0.3 | 0.5 | 1 | 0.949411567 |

| | | | | |
|---|---|---|---|---|
| 0 | 0.4 | 0.5 | 1 | 0.948616386 |
| 0 | 0.5 | 0.5 | 1 | 0.947779343 |
| 0 | 0.6 | 0.5 | 1 | 0.946911817 |
| 0.1 | 0 | 0.5 | 1 | 0.939513618 |
| 0.2 | 0 | 0.5 | 1 | 0.932525065 |
| 0.3 | 0 | 0.5 | 1 | 0.927766385 |
| 0.4 | 0 | 0.5 | 1 | 0.924251327 |
| 0.5 | 0 | 0.5 | 1 | 0.92151618 |
| 0 | 0.1 | 0.5 | 0 | 0.897383042 |
| 0 | 0 | 0.5 | 0 | 0.897098083 |
| 0 | 0.2 | 0.5 | 0 | 0.896862903 |
| 0 | 0.3 | 0.5 | 0 | 0.89594127 |
| 0 | 0.4 | 0.5 | 0 | 0.894805191 |
| 0 | 0.5 | 0.5 | 0 | 0.893529819 |
| 0 | 0.6 | 0.5 | 0 | 0.892189275 |
| 0.1 | 0 | 0.5 | 0 | 0.869953588 |
| 0.2 | 0 | 0.5 | 0 | 0.850793653 |
| 0.3 | 0 | 0.5 | 0 | 0.838570808 |
| 0.4 | 0 | 0.5 | 0 | 0.830194804 |
| 0.5 | 0 | 0.5 | 0 | 0.82409206 |

# Appendix 8 Results of paired t-tests for testing significance of results for Swiss-Prot from chapter 5.

The t-tests were done using Microsoft Excel 2003's Data Analysis add-in. For both .05 AUC and 1.0 AUC we provide a figure showing the output of the paired t-test below each bullet point for each test tried. The P-value to look at is the one-tail value since the hypothesis is that one condition will improve the AUC score (i.e. result in a statistically different mean AUC value). In this case for Swiss-Prot, not all tests resulted in statistically significantly different means. The pre-IDF step always resulted in a statistically significantly improved mean, while the addition of PFAM documents did not except for one case (although addition of PFAM documents always resulted in a larger mean).

- PFAM and GO weights of 0, percent features kept of 0.5; comparing no pre-IDF step versus pre-IDF step, with the hypothesis that the pre-IDF step will improve the .05 AUC score. This is borne out by the highly significant one-tail P-value of 1.15955E-51.

| | *No pre-IDF* | *pre-IDF* |
|---|---|---|
| Mean | 0.024920735 | 0.035667273 |
| Variance | 8.94461E-05 | 4.70996E-05 |
| Observations | 200 | 200 |
| Pearson Correlation | 0.634394836 | |
| Hypothesized Mean Difference | 0 | |
| df | 199 | |
| t Stat | -20.64490582 | |

| | | |
|---|---|---|
| P(T<=t) one-tail | 1.15955E-51 | |
| t Critical one-tail | 1.652546747 | |
| P(T<=t) two-tail | 2.31911E-51 | |
| t Critical two-tail | 1.971956498 | |

- GO weight of 0, percent features kept of 0.5 and pre-IDF step; comparing PFAM weight of 0 to PFAM weight of 0.1, with the hypothesis that the weight of 0.1 will improve the .05 AUC score. In this case, although the mean for a PFAM weight of 0.1 is greater, the hypothesis is not borne out given that the P-value is only 0.221144011 which is greater than the 0.05 significance level. It is still more likely, however, that PFAM weight of 0.1 improves performance, but we cannot statistically conclude this at commonly accepted levels of significance (0.05).

| | *PFAM 0.1* | *PFAM 0* |
|---|---|---|
| Mean | 0.035710069 | 0.035667273 |
| Variance | 4.74679E-05 | 4.70996E-05 |
| Observations | 200 | 200 |
| Pearson Correlation | 0.993472165 | |
| Hypothesized Mean Difference | 0 | |
| df | 199 | |
| t Stat | 0.769873213 | |
| P(T<=t) one-tail | 0.221144011 | |
| t Critical one-tail | 1.652546747 | |
| P(T<=t) two-tail | 0.442288021 | |
| t Critical two-tail | 1.971956498 | |

- No pre-IDF step, GO weight of 0, percent features kept of 0.5; comparing PFAM weight of 0 to PFAM weight of 0.2, with the hypothesis that the weight of 0.2 will improve the .05 AUC score. This is borne out by the significant one-tail P value of 0.005884367.

| | *PFAM 0.2* | *PFAM 0* |
|---|---|---|
| Mean | 0.025253218 | 0.024920735 |
| Variance | 8.65155E-05 | 8.94461E-05 |

| | | |
|---|---|---|
| Observations | 200 | 200 |
| Pearson Correlation | 0.980698366 | |
| Hypothesized Mean Difference | 0 | |
| df | 199 | |
| t Stat | 2.542452979 | |
| P(T<=t) one-tail | 0.005884367 | |
| t Critical one-tail | 1.652546747 | |
| P(T<=t) two-tail | 0.011768733 | |
| t Critical two-tail | 1.971956498 | |

- PFAM and GO weights of 0, percent features kept of 0.5; comparing no pre-IDF step versus pre-IDF step, with the hypothesis that the pre-IDF step will improve the 1.0 AUC score. This is borne out by the highly significant one-tail P-value of 2.19681E-37.

| | *No pre-IDF* | *pre-IDF* |
|---|---|---|
| Mean | 0.897098083 | 0.950253129 |
| Variance | 0.003908059 | 0.001343006 |
| Observations | 200 | 200 |
| Pearson Correlation | 0.654180762 | |
| Hypothesized Mean Difference | 0 | |
| df | 199 | |
| t Stat | -15.83494434 | |
| P(T<=t) one-tail | 2.19681E-37 | |
| t Critical one-tail | 1.652546747 | |
| P(T<=t) two-tail | 4.39361E-37 | |
| t Critical two-tail | 1.971956498 | |

- GO weight of 0, percent features kept of 0.5 and pre-IDF step; comparing PFAM weight of 0 to PFAM weight of 0.1, with the hypothesis that the weight of 0.1 will improve the 1.0 AUC score. The hypothesis is not borne out as the P-value is 0.149192923 which is greater than the commonly accepted significance level of 0.05. However, the mean for PFAM 0.1 is greater and the P-value is still fairly small, so it is more likely than not that adding PFAM documents at weight 0.1 does improve the 1.0

AUC.

| | *PFAM 0.1* | *PFAM 0* |
|---|---|---|
| Mean | 0.950538504 | 0.950253129 |
| Variance | 0.001330293 | 0.001343006 |
| Observations | 200 | 200 |
| Pearson Correlation | 0.994406528 | |
| Hypothesized Mean Difference | 0 | |
| df | 199 | |
| t Stat | 1.042627424 | |
| P(T<=t) one-tail | 0.149192923 | |
| t Critical one-tail | 1.652546747 | |
| P(T<=t) two-tail | 0.298385847 | |
| t Critical two-tail | 1.971956498 | |

- No pre-IDF step, GO weight of 0, percent features kept of 0.5; comparing PFAM weight of 0 to PFAM weight of 0.1, with the hypothesis that the weight of 0.1 will improve the 1.0 AUC score. This is not borne out since the P-value is 0.224964315 which is greater than the commonly accepted significance level of 0.05. However, the mean for PFAM 0.1 is greater and by the P-value it is still more likely that adding PFAM documents at weight 0.1 does improve the 1.0 AUC.

| | *PFAM 0.1* | *PFAM 0* |
|---|---|---|
| Mean | 0.897383042 | 0.897098083 |
| Variance | 0.003884362 | 0.003908059 |
| Observations | 200 | 200 |
| Pearson Correlation | 0.996367991 | |
| Hypothesized Mean Difference | 0 | |
| df | 199 | |
| t Stat | 0.757027727 | |
| P(T<=t) one-tail | 0.224964315 | |
| t Critical one-tail | 1.652546747 | |
| P(T<=t) two-tail | 0.44992863 | |
| t Critical two-tail | 1.971956498 | |

# Bibliography


1. Sujansky, W., *Heterogeneous database integration in biomedicine.* J Biomed Inform, 2001. **34**(4): p. 285-98.

2. Berners-Lee, T., et al., *The World-Wide Web.* ACM Communications., 1994. **37**(3): p. 76-82.

3. Lawrence, S. and C.L. Giles, *Searching the world wide Web.* Science, 1998. **280**(5360): p. 98-100.

4. Berners-Lee, T., J. Hendler, and O. Lassila, *The semantic web*, in *Scientific American*. May 2001. p. 35-43.

5. Antoniou, G. and F. Van Harmelen, *A semantic Web primer*. Cooperative information systems. 2004, Cambridge, Mass.: MIT Press. xx, 238 p.

6. Shadbolt, N., W. Hall, and T. Berners-Lee, *The Semantic Web Revisited.* IEEE Intelligent Systems, 2006. **21**(3): p. 96-101.

7. World Wide Web Consortium (W3C). http:// www.w3.org

8. Gruber, T. What is an Ontology. http://ksl-web.stanford.edu/kst/what-is-an-ontology.html

9. Noy, N. and D. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology.* Stanford University Knowledge Systems Laboratory, Technical Report KSL-01-05, March (2001).

10. Resource Description Framework (RDF). http://www.w3.org/RDF/

11. Naming and Addressing: URIs, URLs, ... http://www.w3.org/Addressing/

12. D2RQ V0.4 - Treating Non-RDF Databases as Virtual RDF Graphs. http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/

13. Sesame RDF Database. http://www.openrdf.org

14. Berners-Lee, T. Relational Databases and the Semantic Web. http://www.w3.org/DesignIssues/RDB-RDF.html

15. RDF Vocabulary Description Language 1.0: RDF Schema. http://www.w3.org/TR/rdf-schema/

16. OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/

17. The Rule of Least Power. http://www.w3.org/2001/tag/doc/leastPower.html

18. SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/

19. Haarslev, V. and R. Möller, *Racer: An OWL Reasoning Agent for the Semantic Web.*

20. Bairoch, A., et al., *The Universal Protein Resource (UniProt).* Nucleic Acids Res, 2005. **33**: p. 154–159.

21. Benson, D.A., et al., *GenBank.* Nucleic Acids Res, 2006. **34**(Database issue): p. D16-20.

22. Bateman, A., *The Pfam Protein Families Database.* Nucleic Acids Research, 2002. **30**(1): p. 276-280.

23. Christie, K., et al., *Saccharomyces Genome Database(SGD) provides tools to identify and analyze sequences from Saccharomyces cerevisiae and related sequences from other organisms.* Nucleic Acids Research, 2004. **32**(Database Issue).

24. Kumar, A., et al., *The TRIPLES database: a community resource for yeast molecular biology.* Nucleic Acids Res, 2002. **30**(1): p. 73-5.

25. Karp, P.D., *Database links are a foundation for interoperability.* Trends Biotechnol, 1996. **14**(8): p. 273-9.

26. Cheung, K.H., et al., *YeastHub: a semantic web use case for integrating data in the life sciences domain.* Bioinformatics, 2005. **21 Suppl 1**: p. i85-96.

27. Broekstra, J. and A. Kampman, *SeRQL: A Second Generation RDF Query Language.* Proc. SWAD-Europe Workshop on Semantic Web Storage and Retrieval, 2003.

28. KALFOGLOU, Y. and M. SCHORLEMMER, *Ontology mapping: the state of the art.* The Knowledge Engineering Review, 2003. **18**(01): p. 1-31.

29. Dou, D., D. McDermott, and P. Qi, *Ontology Translation on the Semantic Web.* International Conference on Ontologies, Databases and Applications of Semantics, 2003: p. 952–969.

30. Ontology Matching. http://www.ontologymatching.org/

31. Lenat, D., *CYC: a large-scale investment in knowledge infrastructure.* Communications of the ACM, 1995. **38**(11): p. 33-38.

32. del.icio.us. A social bookmarks manager. http://del.icio.us

33. Connotea: Free online reference management for all researchers, clinicians and scientists. http://www.connotea.org/

34. Shirky, C. Ontology is Overrated: Categories, Links, and Tags. http://www.shirky.com/writings/ontology_overrated.html

35. Stoilos, G., G. Stamou, and S. Kollias, *A String Metric for Ontology Alignment.* Proceedings of the 4 thInternational Semantic Web Conference (ISWC-05), 2005: p. 624-637.

36. Ashburner, M., et al., *Gene Ontology: tool for the unification of biology.* Nature Genetics, 2000. **25**: p. 25-29.

37. Berman, H., et al., *The Protein Data Bank.* Nucleic Acids Research, 2000. **28**(1): p. 235-242.

38. Google Web Search Engine. http://www.google.com

39. Yahoo Web Search Engine. http://search.yahoo.com

40. PubMed Biomedical Literature Search. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed

41. Brin, S. and L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine.* WWW7 / Computer Networks, 1998. **30**(1-7): p. 107-117.

42. Montelione, G.T. and S. Anderson, *Structural genomics: keystone for a Human Proteome Project.* Nat Struct Biol, 1999. **6**(1): p. 11-2.

43. Gerstein, M., et al., *Structural genomics: current progress.* Science, 2003. **299**(5613): p. 1663.

44. Chandonia, J.M. and S.E. Brenner, *The impact of structural genomics: expectations and outcomes.* Science, 2006. **311**(5759): p. 347-51.

45. Northeast Structural Genomics Consortium (NESG). http://www.nesg.org

46. Goh, C.S., et al., *SPINE 2: a system for collaborative structural proteomics within a federated database framework.* Nucleic Acids Res, 2003. **31**(11): p. 2833-8.

47. TargetDB. http://targetdb.pdb.org/

48. PepcDB. http://pepcdb.pdb.org

49. Tatusov, R.L., et al., *The COG database: an updated version includes eukaryotes.* BMC Bioinformatics, 2003. **4**: p. 41.

50. Mewes, H.W., et al., *MIPS: analysis and annotation of proteins from whole genomes.* Nucleic Acids Res, 2004. **32**(Database issue): p. D41-4.

51. Engelman, D., T. Steitz, and A. Goldman, *Identifying Nonpolar Transbilayer Helices in Amino Acid Sequences of Membrane Proteins.* Annual Review of Biophysics and Biophysical Chemistry, 1986. **15**(1): p. 321-353.

52. Hulo, N., et al., *The PROSITE database.* Nucleic Acids Res, 2006. **34**(Database issue): p. D227-30.

53. Wootton, J.C. and S. Federhen, *Analysis of compositionally biased regions in sequence databases.* Methods Enzymol, 1996. **266**: p. 554-71.

54. Goh, C.S., et al., *Mining the structural genomics pipeline: identification of protein properties that affect high-throughput experimental analysis.* J Mol Biol, 2004. **336**(1): p. 115-30.

55. Gollub, J., et al., *The Stanford Microarray Database: data access and quality assessment tools.* Nucleic Acids Res, 2003. **31**(1): p. 94-6.

56. Edgar, R., M. Domrachev, and A.E. Lash, *Gene Expression Omnibus: NCBI gene expression and hybridization array data repository.* Nucleic Acids Res, 2002. **30**(1): p. 207-10.

57. Bader, G.D., D. Betel, and C.W. Hogue, *BIND: the Biomolecular Interaction Network Database.* Nucleic Acids Res, 2003. **31**(1): p. 248-50.

58. Peri, S., et al., *Human protein reference database as a discovery resource for proteomics.* Nucleic Acids Res, 2004. **32**(Database issue): p. D497-501.

59. Joshi-Tope, G., et al., *Reactome: a knowledgebase of biological pathways.* Nucleic Acids Res, 2005. **33**(Database issue): p. D428-32.

60. Hill, A. and H. Kim, *The UAB Proteomics Database.* Bioinformatics, 2003. **19**(16): p. 2149-51.

61. Desiere, F., et al., *The PeptideAtlas project.* Nucleic Acids Res, 2006. **34**(Database issue): p. D655-8.

62. Blake, J.A., et al., *The Mouse Genome Database (MGD): updates and enhancements.* Nucleic Acids Res, 2006. **34**(Database issue): p. D562-7.

63. Cheung, K.H., et al., *PhenoDB: an integrated client/server database for linkage and population genetics.* Comput Biomed Res, 1996. **29**(4): p. 327-37.

64. Shannon, W., R. Culverhouse, and J. Duncan, *Analyzing microarray data using cluster analysis.* Pharmacogenomics, 2003. **4**(1): p. 41-52.

65. Manduchi, E., et al., *RAD and the RAD Study-Annotator: an approach to collection, organization and exchange of all relevant information for high-throughput gene expression studies.* Bioinformatics, 2004. **20**(4): p. 452-9.

66. Buneman, P., et al., *A data transformation system for biological data sources.* Proceedings of 21st International Conference on Very Large Data Bases, 1995: p. 158-169.

67. Zdobnov, E.M., et al., *The EBI SRS server--recent developments.* Bioinformatics, 2002. **18**(2): p. 368-73.

68. Lee, T.J., et al., *BioWarehouse: a bioinformatics database warehouse toolkit.* BMC Bioinformatics, 2006. **7**: p. 170.

69. Birkland, A. and G. Yona, *BIOZON: a hub of heterogeneous biological data.* Nucleic Acids Res, 2006. **34**(Database issue): p. D235-42.

70. Critchlow, T., et al., *DataFoundry: information management for scientific data.* IEEE Trans Inf Technol Biomed, 2000. **4**(1): p. 52-7.

71. SHETH, A. and J. LARSON, *Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases'.* ACM Computing Surveys, 1990. **22**(3).

72. Kolatkar, P.R., et al., *Development of software tools at BioInformatics Centre (BIC) at the National University of Singapore (NUS).* Pac Symp Biocomput, 1998: p. 735-46.

73. Haas, L., et al., *DiscoveryLink: A system for integrated access to life sciences data sources.* IBM Systems Journal, 2001. **40**(2): p. 489-511.

74. Marenco, L., et al., *QIS: A framework for biomedical database federation.* J Am Med Inform Assoc, 2004. **11**(6): p. 523-34.

75. Wang, X., R. Gorlitsky, and J.S. Almeida, *From XML to RDF: how semantic web technologies will change the design of 'omic' standards.* Nat Biotechnol, 2005. **23**(9): p. 1099-103.

76. Hucka, M., et al., *The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.* Bioinformatics, 2003. **19**(4): p. 524-31.

77. Hermjakob, H., et al., *The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data.* Nat Biotechnol, 2004. **22**(2): p. 177-83.

78. Golbeck, J., et al., *The national cancer institute's thesaurus and ontology.* Journal of Web Semantics, 2003. **1**(1): p. 12.

79. W3C Semantic Web Health Care and Life Sciences Interest Group. http://www.w3.org/2001/sw/hcls/

80. Baader, F. and ebrary Inc. The description logic handbook theory, implementation, and applications. http://site.ebrary.com/lib/yale/Doc?id=10069975 Online book

81. Dublin Core Metadata Initiative. http://dublincore.org/

82. Yu, H., et al., *Genomic analysis of essentiality within protein networks.* Trends Genet, 2004. **20**(6): p. 227-31.

83. Guelzim, N., et al., *Topological and causal structure of the yeast transcriptional regulatory network.* Nat Genet, 2002. **31**(1): p. 60-3.

84. Wuchty, S., *Evolution and topology in the yeast protein interaction network.* Genome Res, 2004. **14**(7): p. 1310-4.

85. Neumann, E., *A life science Semantic Web: are we there yet?* Sci STKE, 2005. **2005**(283): p. pe22.

86. Hendler, J., *Communication. Science and the semantic web.* Science, 2003. **299**(5606): p. 520-1.

87. NCBI FASTA format description. http://www.ncbi.nlm.nih.gov/blast/fasta.shtml

88. MolmovDB Database of Macromolecular Movements. http://www.molmovdb.org

89. Pseudogene.org. http://www.pseudogene.org

90. HARRIS, S. and N. SHADBOLT, *SPARQL query processing with conventional relational database systems.* Lecture notes in computer science. **3807**: p. 235-244.

91. RDQL - A Query Language for RDF. http://www.w3.org/Submission/RDQL/

92. Yu, H., et al., *Annotation transfer between genomes: protein-protein interologs and protein-DNA regulogs.* Genome Res, 2004. **14**(6): p. 1107-18.

93. Jansen, R., et al., *A Bayesian networks approach for predicting protein-protein interactions from genomic data.* Science, 2003. **302**(5644): p. 449-53.

94. Chen, N., et al., *WormBase: a comprehensive data resource for Caenorhabditis biology and genomics.* Nucleic Acids Res, 2005. **33**(Database issue): p. D383-9.

95. Zhang, Z. and M. Gerstein, *Large-scale analysis of pseudogenes in the human genome.* Curr Opin Genet Dev, 2004. **14**(4): p. 328-35.

96. Harrison, P.M. and M. Gerstein, *Studying genomes through the aeons: protein families, pseudogenes and proteome evolution.* J Mol Biol, 2002. **318**(5): p. 1155-74.

97. MIPS. http://mips.gsf.de/genre/proj/yeast/

98. Harrison, P., et al., *A small reservoir of disabled ORFs in the yeast genome and its implications for the dynamics of proteome evolution.* J Mol Biol, 2002. **316**(3): p. 409-19.

99. Zhang, Z., et al., *Millions of years of evolution preserved: a comprehensive catalog of the processed pseudogenes in the human genome.* Genome Res, 2003. **13**(12): p. 2541-58.

100. Salton, G., *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. 1989: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

101. Salton, G. and C. Buckley, *Term-weighting approaches in automatic text retrieval.* Information Processing and Management: an International Journal, 1988. **24**(5): p. 513-523.

102. Salton, G. and M. McGill, *Introduction to Modern Information Retrieval*. 1986: McGraw-Hill, Inc. New York, NY, USA.

103. Yang, Y. and X. Liu, *A re-examination of text categorization methods.* Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999: p. 42-49.

104. Williams, K. and R. Calvo. *A Framework for Text Categorization*. in *7th Australasian Document Computing Symposium*. 2002. Syndey, Australia.

105. Sebastiani, F., *Machine learning in automated text categorization.* ACM Computing Surveys (CSUR), 2002. **34**(1): p. 1-47.

106. Porter, M., *An algorithm for suffix stripping.* Program, 1980. **14**(3): p. 130–137.

107. Witten, I., A. Moffat, and T. Bell, *Managing gigabytes: compressing and indexing documents and images.* The Morgan Kaufmann Series In Multimedia Information And Systems, 1999: p. 519.

108. Medical Subject Headings. http://www.nlm.nih.gov/mesh/meshhome.html

109. Google Scholar. http://scholar.google.com/

110. Swish-e Simple Web Indexing System for Humans -Enhanced. http://www.swish-e.com

111. Rabinowitz, J., *How to index anything.* Linux Journal, 2003. **2003**(111).

112. SGD curated gene literature bibliography. ftp://genome-ftp.stanford.edu/pub/yeast/literature_curation/gene_literature.tab

113. Witten, I.H. and E. Frank, *Data mining : practical machine learning tools and techniques*. 2nd ed. Morgan Kaufmann series in data management systems. 2005, Amsterdam ; Boston, MA: Morgan Kaufman. xxxi, 525 p.

114. Fawcett, T., *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers.* HP Laboratories technical report, 2003.

115. Swets, J., R. Dawes, and J. Monahan, *Better decisions through science.* Sci Am, 2000. **283**(4): p. 82-7.

116. Dalgaard, P., *Introductory statistics with R*. 2002: Springer New York.

117. Aphinyanaphongs, Y., A. Statnikov, and C.F. Aliferis, *A comparison of citation metrics to machine learning filters for the identification of high quality MEDLINE documents.* J Am Med Inform Assoc, 2006. **13**(4): p. 446-55.

118. Carroll, J., et al., *Named Graphs.* Journal of Web Semantics, 2005. **3**(4).

119. Clark, T., S. Martin, and T. Liefeld, *Globally distributed object identification for biological knowledgebases.* Briefings in Bioinformatics, 2004. **5**(1): p. 59-70.

120. Tabulator: Async Javascript and Semantic Web. http://www.w3.org/2005/ajar/tab.html

121. Cohen-Boulakia, S., C. Froidevaux, and E. Pietriga, *Selecting Biological Data Sources and Tools with XPR, a Path Language for RDF.* Pacific Symposium on Biocomputing (PSB), Maui, Hawaii.(2006).

122. Angles, R. and C. Gutierrez, *Querying RDF Data from a Graph Database Perspective.* 2nd. European Semantic Web Conference (ESWC2005).

123. Weiss, S.M., *Text mining : predictive methods for analyzing unstructured information*. 2005, New York: Springer. xii, 237 p.

124. Shatkay, H. and R. Feldman, *Mining the biomedical literature in the genomic era: an overview.* J Comput Biol, 2003. **10**(6): p. 821-55.

125. Friedman, C., et al., *GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles.* Bioinformatics, 2001. **17 Suppl 1**: p. S74-82.

126. Salton, G., *The SMART retrieval system*. 1971: Prentice-Hall.

127. Salton, G. and C. Buckley, *Improving retrieval performance by relevance feedback.* Journal of the American Society for Information Science, 1990. **41**(4): p. 288-297.

128. Schaffer, A.A., et al., *Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements.* Nucleic Acids Res, 2001. **29**(14): p. 2994-3005.

129. Kahle, B., *Wide Area Information Servers: An Executive Information System for Unstructured Files.* Electronic Networking: Research, Applications and Policy, 1992.

130. McCahill, M., *The Internet Gopher: A distributed server information system.* ConneXions–The Interoperability Report, 1992. **6**(7): p. 10–14.

131. PubMed Computation of Related Articles. http://www.ncbi.nlm.nih.gov/entrez/query/static/computation.html

132. Jakulin, A., D. Mladenic, and B. Fortuna, *Ontology grounding.* Proceedings of the 8th International multi-conference Information Society IS-2005, 2005: p. 170–173.

133. Yuan, S. and J. Sun, *Ontology-based structured cosine similarity in document summarization: with applications to mobile audio-based knowledge management.* Systems, Man and Cybernetics, Part B, IEEE Transactions on, 2005. **35**(5): p. 1028-1040.

134. Bernstam, E.V., et al., *Using citation data to improve retrieval from MEDLINE.* J Am Med Inform Assoc, 2006. **13**(1): p. 96-105.

135. Douglas, S.M., G.T. Montelione, and M. Gerstein, *PubNet: a flexible system for visualizing literature derived networks.* Genome Biol, 2005. **6**(9): p. R80.

136. Cantor, C.R., *Orchestrating the Human Genome Project.* Science, 1990. **248**(4951): p. 49-51.

137. Lander, E.S., et al., *Initial sequencing and analysis of the human genome.* Nature, 2001. **409**(6822): p. 860-921.

138. Venter, J.C., et al., *The sequence of the human genome.* Science, 2001. **291**(5507): p. 1304-51.

139. Crick, F., *Central dogma of molecular biology.* Nature, 1970. **227**(5258): p. 561-3.

140. Needleman, S.B. and C.D. Wunsch, *A general method applicable to the search for similarities in the amino acid sequence of two proteins.* J Mol Biol, 1970. **48**(3): p. 443-53.

141. Smith, T.F. and M.S. Waterman, *Identification of common molecular subsequences.* J Mol Biol, 1981. **147**(1): p. 195-7.

142. Altschul, S.F., et al., *Basic local alignment search tool.* J Mol Biol, 1990. **215**(3): p. 403-10.

143. Pearson, W.R., *Rapid and sensitive sequence comparison with FASTP and FASTA.* Methods Enzymol, 1990. **183**: p. 63-98.