Index-Driven XML Data Integration to Support Functional Genomics

Ela Hunt[1], Evangelos Pafilis[1,2], Inga Tulloch[1,2], and John Wilson[2]

[1] Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK
ela@dcs.gla.ac.uk
[2] Department of Computer and Information Sciences, University of Strathclyde, Glasgow G1 1XH, UK
jnw@cis.strath.ac.uk

Abstract. We identify a new type of data integration problem that arises in functional genomics research in the context of large-scale experiments involving arrays, 2-dimensional protein gels and mass-spectrometry. We explore the current practice of data analysis that involves repeated web queries iterating over long lists of gene or protein names. We postulate a new approach to solve this problem, applicable to data sets stored in XML format. We propose to discover data redundancies using an XML index we construct and to remove them from the results returned by the query. We combine XML indexing with queries carried out on top of relational tables. We believe our approach could support semi-automated data integration such as that required in the interpretation of large-scale biological experiments.

1 Introduction

The field of functional genomics promises to provide an understanding of the cellular processes that control development, health and disease. The advent of large-scale laboratory techniques including sequencing, microarrays and proteomics has revolutionised the discipline and allowed biologists to take a global view of biological processes. A sequencing run[3], a microarray experiment[4] or a 2-D protein gel[5] suddenly deliver a large amount of data that needs to be assessed with relationship to a particular problem being investigated. Current data acquisition and processing methods that rely on web browsing and querying for single genes of interest are no longer sufficient. A microarray experiment can yield a list of 500 gene names, all possibly implicated in the mechanism of a biological process or disease that is being investigated. Tools such as Sequence Retrieval System (SRS)[6] [10] provide a means of executing a query over a range of data sources but do not integrate the results. Keyword searching can be used to limit

[3] http://www.sanger.ac.uk/HGP/
[4] http://www.mged.org
[5] http://psidev.sourceforge.net/
[6] http://srs.ebi.ac.uk

the range of results but still produces large numbers of links that need to be investigated. The biologists we work with are able to identify lists of genes but have not solved the problem of data and literature acquisition that would help to place each gene in its context. The interpretation of gene expression and protein expression data requires data integration on a large scale. We believe that a new *generic* approach to data integration is needed. We can no longer afford to manually create new wrappers, mappings and views expressing the information need of every biologist performing genome or proteome analysis. A generic solution would be preferable. Since a significant number of biological databases are available in XML format, we have the opportunity to use that format for semi-automated data integration. We propose an XML data integration system that can gather all available XML data for any gene of interest and present the biologist with a digest of all known information, similar to GeneCards [16] but not limited to the *Homo sapiens* species.

The eXtensible Markup Language, XML[7], is one of the alternative formats used in biological data presentation. In particular, the SRS system handles XML easily and provides tools for the indexing of such data. SRS does not however integrate data, instead, integrated data views have to be crafted by skilled programmers. The biological users we know have to manually retrieve and integrate data. It is our intention to provide a data integration service for such users. We propose a different scenario. Our user is a biologist performing a large-scale biological experiment. The experiment produces a list of gene or protein names and our task is to find all available information about those data items from the public web databases.

We believe that a gene- or protein-focused view of all biological databases can satisfy the needs of a large number of researchers. An XML tree giving a summary of all information relevant to a given gene or protein could be processed by a data integration system and stored as a pre-processed summary. Each user would then select from that summary a relevant subset of information. Creating an XML tree for a gene can be done simplistically, by creating a root and attaching a gene-specific subtree from each database to that root. However, this solution is not satisfactory because the same data will be replicated in many subtrees. We aim to integrate the data at the XML level while taking into account both the data structure and content. We adopt ideas from the data mining community where both the tree structure and content are considered. We combine this idea with the use of database indexes that will make possible an approach similar to data mining on very large data sets. By indexing we reduce the computational complexity of the processing needed to find correspondences between different paths in XML trees, and by using simple queries on all indexed data we can develop well-founded mappings to be used to remove duplicated information.

Our work is based on the following assumptions and observations. We observe that biological databases are developed independently, and their XML structures have various levels of nesting and often use disjoint vocabularies in their tree structures. Because the data comes from various domains (for instance

[7] http://www.w3.org/TR/REC-xml

eukaryotic promoters, protein structures, genome maps), it is unlikely that automated matching of attribute names will allow us to merge the data. Each biologist has different needs and may use any number of databases. It is consequently impossible to prescribe how data should be merged and presented. On the other hand, we expect a certain degree of order and data sharing between different database trees. We assume shared database identifiers, unique IDs for single records (because most databases started as flat files each entry will have an ID) and large numbers of records of similar structure.

Our contributions are as follows. We identify a new type of data integration requirement. We provide a critique of existing data integration systems and we propose the idea of item-based XML data integration. We propose an index-based algorithm for redundancy removal and an application architecture to support querying. The rest of the paper follows this order and closes with a discussion and conclusions.

2 Searching On-Line Databases

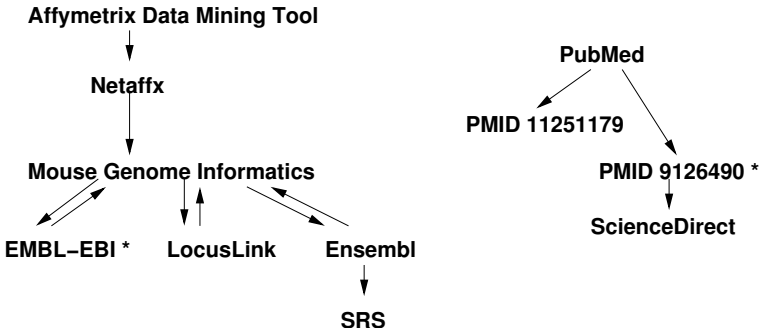

Fig. 1. Webpages visited during searching. * indicates sites that contained material that the researcher found interesting

Biologists have a variety of aims when searching on-line databases. We conducted extended interviews with four research groups that use microarray and proteomics technologies. We observed the process of data analysis performed as a series of web queries. We gathered logs of web activity using Muffin[8] and focused on web queries and data navigation paths.

2.1 Study 1, Mouse Breast Development as a Model for Cancer

Study 1 aims to understand gene expression in mouse mammary gland development and involution. A series of microarray experiments has led to the identifi-

[8] www.muffin.org

cation of 400 gene names. All genes should be studied and potential candidates identified on the basis of a known role in cancer and mammary gland development. The researcher reported spending six months characterising 100 genes by web browsing. He would like to be able to study the remaining 300 genes. The current practice is to start with the Affymetrix database[9] and for each probe name identified in the experiment, to find the corresponding gene. The link to the gene is followed, finally leading to PubMed articles[10]. If the article title correlates with the research interest of *Study 1*, it will be downloaded. The sequence of a typical search is shown in Figure 1. Leading on from the Affymetrix Data Mining Tool, the researcher found useful material in EMBL, but ultimately resorted to PubMed to issue a query for the gene name listed in the Affymetrix database. PubMed produced two articles that were of interest, one of which was pursued through to Science Direct[11], and ultimately printed out for subsequent study. This process was followed for all 100 genes and the researcher would like to be able to automate it. Moreover, he would like an additional facility, that of grouping the articles according to the number of the gene names in the query set they refer to. For instance, articles could be presented in a ranked order with those shown first that mention the greatest number of gene names out of the initial query set.

2.2 Study 2, Search for Candidate Genes for Hypertension

Study 2, working on a rat model of hypertension, wants to identify the exact genome location for 400 genes and to find supplementary information about similar genes in mouse and human. The web search using Ensembl[12], NCBI[13], MGD[14] and RGD[15] consumed a significant amount of time, simply to find gene locations. The identification of up-to-date comparative maps of human, mouse and rat for a subset of the genes located on one rat chromosome was also time-consuming. The search started from the Affymetrix database and the initial list of gene positions was gathered by web searching. The list was complemented with additional probe positions calculated by running a large-scale sequence comparison where probes from the Affymetrix database were compared to the rat genome.

2.3 Study 3, Rat Model of Schizophrenia

Study 3 is characterising 250 genes that are differentially expressed in a rat model of schizophrenia. We estimate that a full analysis of those genes will consume

[9] www.affymetrix.com
[10] www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed
[11] www.sciencedirect.com
[12] www.ensembl.org
[13] www.ncbi.nlm.nih.gov/LocusLink/
[14] www.informatics.jax.org
[15] rgd.mcw.edu

several months. The Affymetrix database of probe names is the starting point. Web queries on probe sequences lead the researcher to the DNA sequence of the rat gene or its human or mouse homologue. Those target sequences will then be used to design laboratory tests and relevant publications will be sought in PubMed and downloaded.

2.4 Study 4, *Trypanosoma brucei* Proteins

Study 4 is characterising the proteome of *Trypanosoma brucei* by 2-D gel electrophoresis [8] and mass spectrometry. Mass spectrometry-based protein identification is based on sequence searching. For each protein similar to the one observed on a 2-D gel, the study will want to gather information from the web, possibly extending to known publications, predicted genes, protein structures or protein motifs. Correlation with any published experimental data, including microarrays, would be beneficial to the study.

3 Existing Integrative Tools

3.1 Entrez

Entrez is possibly the most popular biological query system[16]. It offers a query interface which allows the biologist to distribute the query over all of Entrez databases. However, there is no further data integration. Query results list each of the underlying data sources together with the corresponding number of hits in each database. Clicking on any of the databases which show a match will deliver a list of links to data items stored in each database.

3.2 Sequence Retrieval System (SRS)

SRS is a warehousing and indexing system [10]. The system regularly downloads flat files and XML databases during low usage periods. Indexes are constructed in order to support queries over a number of data sources. Relational databases are not mirrored but their query facilities are captured in such a fashion that the databases can be queried effectively over the Internet. A user can select a set of target databases and issue a query spanning a selection of data sources. The query is then sent to remote relational databases and to the local flat files and XML data sources. The results are then brought together for the user. The results are presented as web pages of links or tables of data. Figure 2 shows an example result set.

[16] www.ncbi.nlm.nih.gov/Entrez/

EMBL	Accession (Links to SVA)	Description	SeqLength
EMBL:BC052031	BC052031;	Mus musculus suppressor of cytokine signaling 3, mRNA (cDNA clone MGC:62384 IMAGE:5707539), complete cds.	2545
EMBL:BC054214	BC054214;	Xenopus laevis suppressor of cytokine signaling 3, mRNA (cDNA clone MGC:64393 IMAGE:6878712), complete cds.	1694
EMBL:BI373842	BI373842;	RE61259.5prime RE Drosophila melanogaster normalized Embryo pFlc-1 Drosophila melanogaster cDNA clone RE61259 5 similar to Socs36E: FBan0015154 GO:[signal transduction (GO:0004871)] located on: 2L 36E5-36E5;: 05/16/2001, mRNA sequence.	512
EMBL:BI605303	BI605303;	RH70884.5prime RH Drosophila melanogaster normalized Head pFlc-1 Drosophila melanogaster cDNA clone RH70884 5 similar to Socs36E: FBan0015154 GO:[signal transduction (GO:0004871)] located on: 2L 36E5-36E5;: 08/24/2001, mRNA sequence.	571

Fig. 2. SRS nucleotide database query result for *socs3*

3.3 GeneCards

The information content of GeneCards [16] focuses on the human genome. Data is integrated either by establishing local caches or by executing queries against on-line databases. Local caches are held in flat file format although this is currently being migrated to XML. Data is integrated from around 50 sources including Swiss-Prot, OMIM, Ensembl and LocusLink. The search engine removes redundancy and presents a digest of the information extracted. The HUGO database is used to identify the aliases of a particular gene. The results page presents a synopsis of the function of the gene and also a link to PubMed to allow the identification of literature that describes the gene. Source articles are not directly referenced on the results page but a variety of links (in addition to PubMed) are provided that lead to publications.

3.4 MedMiner

Medminer [17] is a text mining tool. It filters article abstracts and presents the most relevant portions. It consists of three components: the facility that queries web databases, the text filtering engine and the user interface. PubMed and GeneCards are used by MedMiner as information sources. The GeneCards query is used to return genes that are relevant to the user. The user can then select one of those genes for inclusion in a PubMed query. Arguments can be added to the PubMed query to initiate the filtering process. The abstracts of the returned articles are broken into sentences and each sentence is tested to see whether it conforms to the relevance criterion. The latter is true when the sentence contains the query arguments and one of the MedMiner keywords. These keywords are truncated terms associated with biological processes. When the user wishes to study the genes associated with a certain phenomenon, the co-occurrence of a gene identifier with a phenomenon relevant term in the abstract

of an article increases the possibility that this article is an important one. After the text mining has been concluded, the citations that pass the relevance filter are grouped according to the keywords they contain. MedMiner, instead of showing the whole article abstract, displays only the sentence found to be relevant, with the keywords and the query argument, highlighted. The use of keywords as the article relevance metric runs the risk of high false negatives, especially if the keywords have not been wisely selected and they do not cover all the aspects of a certain phenomenon or process. A potential limitation is that relationships between keywords and gene identifiers that span multiple sentences will not be picked up. The choice of displaying only the relevant sentence instead of the abstract itself, can result in significant time savings. However there is a trade off with the possibility that important pieces of information existing in the rest of the abstract will never be displayed.

3.5 Limitations of Existing Tools

GeneCards is specific to the human genome and does not cover mouse, rat, or other organisms. MedMiner inherits the same weakness. The GeneCards system is based on hard-coded schema mappings and is not easily extended to add new data resources, or new XML files acquired from laboratory equipment or from collaborators. The approach is not flexible or scalable, however it represents a significant step in data integration methods. In fact, an approach that combines both literature and database resources and is extensible to any species of interest is required. Ideally, a PubMed abstract could be co-indexed with terms found in traditional databases to achieve an integration of textual, experimental and other annotation data present in public repositories. Further to that, queries over sequence data should be part of the data integration system allowing the integration of textual, database and sequence data to be even more effective.

4 Structural Conflicts in Data

The significance of XML in the integration of biological databases has been recognised for some time [1]. Some databases, such as Swiss-Prot, are available for download in XML format whereas other systems such as PubMed are able to return the results of queries as XML. The increasing conformance to the XML standard will help with the outstanding issues of integration between heterogeneous databases. Significant difficulties will however remain as a result of the current divergence in data design and the expected continuation of this divergence. Figures 3 and 4 show parts of the XML trees for the Swiss-Prot database and queries returned from PubMed. The gene name in Swiss-Prot is found by following the path: sptr/entry/reference/gene/name. There is no equivalent path in PubMed, but the same name may be found in an abstract, by following the path: PubMedArticleSet/PubMedArticle/MedlineCitation/Article/-Abstract/AbstractText. There is also another connection between the two trees.

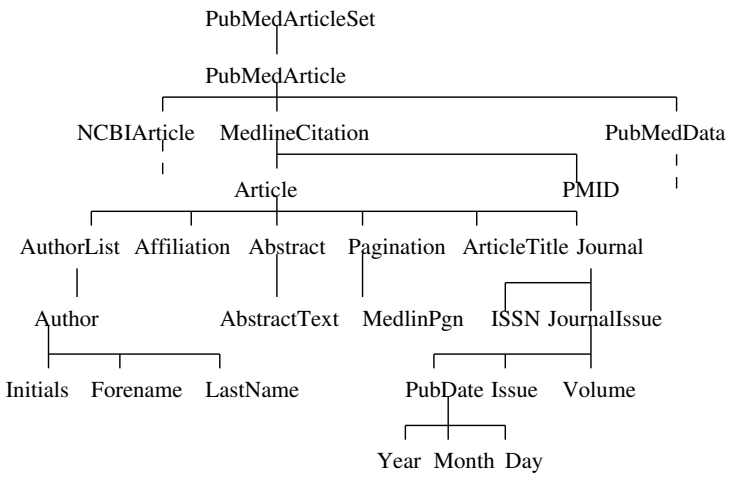

Fig. 3. Tree structure for PubMed query results

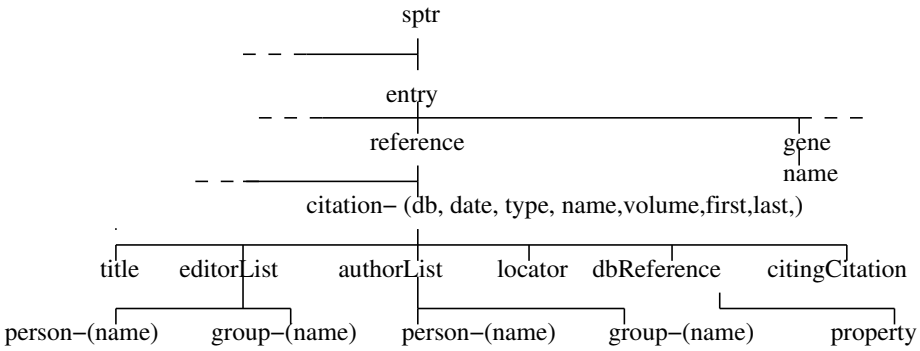

Fig. 4. Tree structure for the Swiss-Prot database. Attributes associated with tag names are bracketed

A Swiss-Prot entry records a PubMed article as a leaf on the path sptr/-entry/reference/citation/dbReference and the key used by Swiss-Prot is the key present in the PubMed tree reachable via PubMedArticleSet/PubMedArticle/-MedlineCitation/PMID. It can clearly be seen that the two structures contain several incompatibilities. Although there are some matches between the structures at the level of tag names (eg *authorList*) there are many instances of data items that contain common semantics but are represented by varying structures. Dates in Swiss-Prot are represented by a single string giving the month and year whereas in PubMed they are represented by a sub-tree. Attributes in Swiss-Prot contain equivalent semantic content that is held by tags and their values in PubMed.

5 XML Result Construction

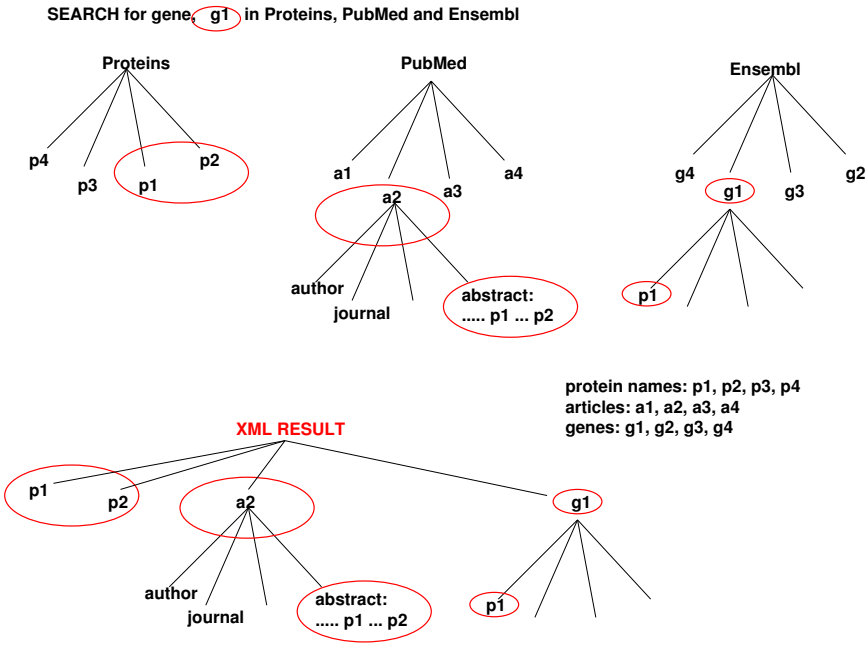

Fig. 5. A schematic view of data querying and integration using XML trees

The four biological studies we reported on are characterised by the use of a variety of databases that are manipulated using web interfaces, most of which offer a simple query modality and return the result as a mixture of textual information and web links. The starting point of the search is the result from a laboratory experiment. In the microarray context it is a list of probe names and in proteomics it is a list of peptide masses delivered by a mass spec instrument. In both cases the researchers enter their query in one of the available interfaces and follow a list of links returned by the query. The resulting data returned from each of those queries represents a subtree of an XML tree for each database being queried. Figure 5 shows a query for a gene *g1* which returns two items *p1, p2* from a protein database, an article *a2* from PubMed and a gene *g1* from Ensembl. Conceptually, all the information returned by those queries (or obtained by following the links between databases), represents a union of three XML subtrees. These subtrees can be brought together, by adding a root and attaching the subtrees to it. We believe that we could build such XML trees for each gene of interest. This approach would go further than the SRS system which presents the user with a list of items. One of the difficulties the user has

with SRS is that the data returned from the queries is presented as a sequence of items, rather than integrated in such a way as to present a unified result. This forces the researcher to follow the links, save the web pages and then deal with a large body of heterogenous data that contains duplication and is not ordered in a meaningful way. We want to take advantage of SRS, and extend it to reconcile the results returned from a number of databases. We will find data repetitions and remove them first. The next step will be to provide the facility to select the information in the tree according to semantic criteria, so that the user interested in protein structures gets the structure information and a user interested in gene localisation gets a part of the XML tree showing gene locations. This grouping can be performed by placing each source database into a general category according to the information it stores. This is similar to the grouping of the databases indexed by the SRS at EBI[17]. Additionally, if a user wants to work with a list of genes, clustering operations could be carried out to highlight the genes that share subtrees of data. Finally, we need to think how to present this information in a way that supports the understanding of this complex data. This last issue remains outside the scope of this work.

6 XML Indexing and Data Integration

A high-level overview of database integration techniques is provided by Garcia-Molina and colleagues [6] who distinguish between federation, warehousing and mediation. In the biological arena, NCBI databases are a *federation* where the sources are independent, but call on one another to supply additional information. The second approach, *warehousing*, requires constructing local copies of relevant databases. SRS is a warehousing system that keeps data fresh by downloading new information overnight and re-indexing it. Lastly, *mediation* is the use of software components that support queries over data sources using various storage formats and schemas. Discovery Link [10] is a mediation system that distributes relational queries and brings the results together. Warehousing is assumed to provide more efficient queries, because the dependence on external servers and the volume of internet traffic is reduced. However, warehousing or mediation do not automatically integrate data. To integrate data, one needs to write complex queries, and none of the tools reviewed in Section 3 offer a query facility that could be used by a biologist who does not understand the semantics and syntax of the underlying data sources.

There are two facets to data integration. One is the integration of schemas and the other the integration of data values. So far, schema integration has received the most attention and recent reviews of schema matching have been conducted by Rahm and Bernstein [15] and Do and colleagues [3]. Assuming that one-to-one schema mappings can be generated, there is no accepted way to store them, manipulate them or to merge more than 2 schemas [12], unless they cover the same or very similar data [7]. Current work in the area of schema mapping does not concern itself with data values. However, there have been attempts to

[17] srs.ebi.ac.uk

include data values in query view definition, most recently by Yan and colleagues [18].

Wrapping and mediation systems abstract from data values. This is partly predicated by the fact that they are designed to perform lazy access to data. Example systems include Bio-Kleisli, K2, Tambis, Discovery Link and OPM [10]. XML based approaches in this area focus on query languages and schema transformations [11, 15, 13, 14, 5].

Recent work in machine learning (ML) is beginning to address the use of data values in data integration. Kurgan and co-workers [9] performed ML on XML tags but did not consider all the data values stored in the tree. Chua and co-workers [2] performed relational data integration via attribute data matching, assuming the knowledge of entity matches. The strength of the method lies in the identification of a variety of statistical tests that are applied in attribute value matching. The method is of interest and we are planning to test it in XML data integration, by applying it to leaves that cannot be matched using relational methods. Doan *et al.* [4] use ML in concept matching. They match pairs of ontologies, which include data values stored in tree leaves. Their system uses a subset of data values to achieve concept matching and takes into account constraints, neighbourhoods of nodes, data paths and heuristics. Despite the progress reported in ML, we believe that current ML approaches suffer from several drawbacks. They can only match a pair of schemas at a time. They use only some of the data because the matching process entails an exhaustive comparison of all attributes or nodes, which leads to a combinatorial explosion of matching activity. We believe that ML could be applied as a last step in data matching and should be supported by indexing, in order to contain the combinatorial explosion of matching.

7 Integrating Data for One Gene or Protein

The semantics of XML data rest both in the tree structure and in the leaves. We intend to benefit from both types of information. To encode the tree structure, we index all paths, including the leafs. When the data is indexed, each leaf value can be examined in turn, to see if it is a candidate for matching (redundancy removal). For each leaf value we check how many times it occurs in the data, and count how many distinct paths lead to it. Groups of leaves that share the same paths are immediate candidates for path merging if they come from distinct databases. If a leaf is reachable via two distinct paths in one database, the semantics of those paths need to be examined either automatically (by algorithms) or with expert help. For any two matching paths, as identified by shared leaf sets, we can also check the immediate leaf neighbourhood to identify shared subtrees. We now outline some of the queries to use in data integration.

The simplest case is the same leaf being reachable by two different paths in two databases. This can be formalised as paths *DB1/p1/a1/leaf1* and *DB2/p2/a2/leaf1*, which can be merged given a significant number of leaves that share two such patterns. The case of matching leaves in the same database, which

needs further disambiguation can be expressed as *DB1/p1/a1/leaf1* and *DB1/p2/a2/leaf1*. Shared subtrees can be captured as paths that share leaves and prefixes, i.e. *DB1/p1/a1/leaf1* and *DB1/p1/a2/leaf2*, matching the pair consisting of *DB2/p3/a3/leaf1* and *DB2/p3/a4/leaf2*, again based on a significant number of subtrees sharing the pattern. A further step would be to explore paths of type *DB1/p1/a1/leaf1*, *DB2/p2/a2/leaf5*, *DB2/p2/a2/leaf6* where *leaf1 = leaf5 + leaf6* (+ expresses concatenation), in order to identify trees that split longer strings into multiple leaves. Finally, linguistic data mining could be performed on leaves to identify matches that are not exact. We are planning to use techniques similar to those described by [12, 4, 2].

The analysis of leaf values assumes that such values are indexed with the paths leading to them. We propose to use a relational representation that will enable the identification of shared leaf sets.

8 The Indexing Algorithm

Database names are stored in a relation *DB(i, DBname)*. Each record in a database has a *Key* stored in a relation *Key(j, Keyname)*. Tags are recorded in a relation *Tag(k, Tagname)*. XML paths and their components, excluding the leaves, are indexed in relation *Path(l, k, depth)* where l is the path ID, k is the tag ID and *depth* is the distance of the tag in the path from the root. Text contained in the leaves is stored in a relation *Leaf(m, Leaftext)* and the relationship between the database i, key j, path l and leaf m, with leaf order *order* within a given record is expressed as *Data(i, j, l, m, order)*. The proposed index can be built in one text traversal, using relational technology and memory-resident tries. As the XML data is traversed, it is written line by line to the database, with appropriate tables being updated to reflect the incoming data. We will use memory-resident tries to store tags, keys, paths and leaves, with the relevant identifiers of each entry kept in the tries to avoid database lookups during the indexing phase.

The complexity of building the index can be expressed in relationship to the length of the text n. It is $O(n \log n)$ because the average tree height is $\log n$. The creation of new integer identities for *tokens* like paths, keys, leaves, etc. is done in logarithmic time because it requires one lookup of the last integer used in each domain (constant time) and one traversal from the trie root to a leaf.

9 Application Architecture

The process of data integration required by functional genomics consists of four phases. In Phase 1 an initial list of gene names is produced. This list can be downloaded from the Affymetrix database, which maps probe names to gene names. Alternatively, a search program like Mascot[18], which accepts a list of peptide masses as input and produces a list of matching proteins, can provide the

[18] www.matrixscience.com/

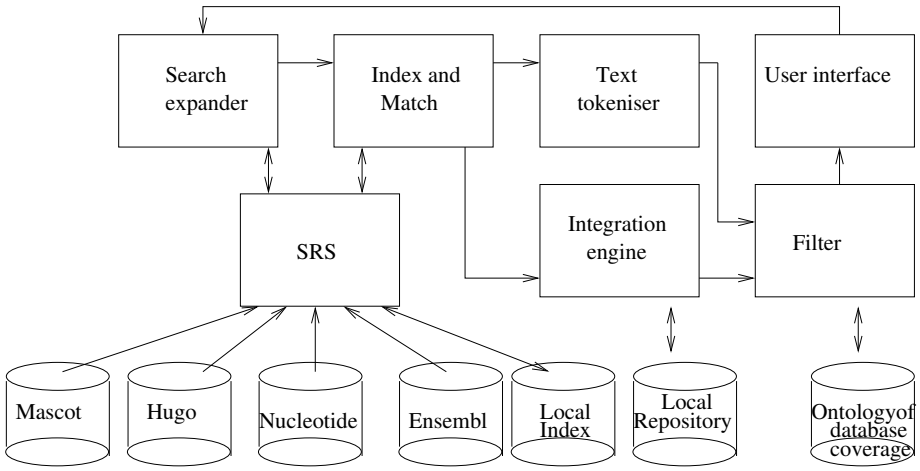

Fig. 6. Architecture for integrated search over biological databases

protein names. In Phase 2 the query is expanded to add a list of synonyms to each gene or protein name. We will adopt the solution used by the GeneCards which uses the HUGO database of gene names to expand searches with gene synonyms, but we will use synonym tables for the human, mouse and rat genomes. In Phase 3 a query is issued to the SRS system and links to individual database entries from several databases are returned. Results are returned from these databases either as objects or XML trees. All structures are converted to the common XML format prior to reconciliation, and added to the index. A local repository is used to store XML structures. In Phase 4 the data are integrated and delivered to the user. It is possible to carry out all four phases in the background and prepare a daily up-to-date digest of information for all genes of interest. When a user poses a query, the pre-processed results for each gene are brought in and further operations involving the selection of a subset of data relevant to the particular user can be performed. Figure 6 captures our system architecture for data integration.

10 Discussion, Conclusions and Further Work

We believe we have captured a new type of user requirement in the area of biological database integration. This requirement consists of the need to acquire all relevant data about a list of gene or protein names produced by a large scale experiment. The further part of this requirement is to remove redundancies in the data and to allow for selection and clustering of the results. This requirement is hard to satisfy using either schema matching or manual data search approaches. We sketch out how this requirement could be met with a new approach to XML data integration.

To our knowledge none of the existing approaches to XML data integration has proposed to index the data leaves and examine the relationships between leaf values in the context of the paths present in the tree. Similarly, we have not seen any accounts of the use of data indexes in machine learning approaches to data integration. We propose to use a path and leaf index that can be built in $O(n \log n)$ time for a dataset of size n. The index can then be traversed with relational queries to identify leaf matches and subtree matches. Additionally, data mining can be carried out on leaf sets to identify opportunities for redundancy removal.

We propose to implement the system as an added feature on top of the SRS system which will mirror a variety of databases. The databases have to be grouped using a simple ontology of data coverage, similar to the current SRS arrangement. The SRS system will be enhanced with an indexing facility for all XML paths and a data mining facility that will reduce data duplication within XML trees for each gene of interest. The data will be filtered according to the database coverage criteria, possibly clustered with regard to genes that share publications and sequence data and presented to the user. The issues of clustering and data visualisation are outside this proposal and will achieve attention in the future.

We are implementing the system we proposed in order to test the viability of our approach. The algorithms we sketched out need significant refinement and the system architecture needs to be drawn in more detail. We will store a subset of SRS data and test our ideas in the context of mouse, rat and human genomes. We will co-index experimental data produced by the biologists and present an integrated view of the external and private data sources.

We conclude that we have identified a new data integration requirement that arises in functional genomics research. We assessed the existing tools and approaches in this area and proposed a new approach that might contribute to the discussion on XML data integration and possibly lead to a better understanding of the problems of automated data integration. We are currently implementing a system prototype and will evaluate our approach with significant amounts of data.

Acknowledgements

We thank our collaborators at the University of Glasgow: Dr T. Stein and Prof. B. Gusterson of the Division of Molecular Pathology and Cancer Sciences, Dr N. Hanlon and Prof. A. F. Dominiczak of the British Heart Foundation Glasgow Cardiovascular Research Centre, A. Jones of the Department of Computing Science. We also thank Dr C. Winchester and A. Marquet of YRING (Yoshitomi Research Institute of Neuroscience in Glasgow), at the University of Glasgow and the University of Strathclyde. This research was supported by the Medical Research Council of the UK, the Royal Society, the Carnegie Trust for the Universities of Scotland and by a Synergy grant from the University of Glasgow and the University of Strathclyde.

References

1. Achard, F. *et al.*: XML, Bioinformatics and Data Integration. Bioinformatics. **17** (2001) 115–125
2. Chua, C. E. H. *et al.*: Instance-Based Attribute Identification in Database Integration. The VLDB Journal. **12** (2003) 228–243
3. Do, H. H. *et al.*: Comparison of Schema Matching Evaluations. Proc. GI-Workshop Web and Databases, Erfurt. (2002) http://lips.informatik.uni-leipzig.de:80/pub/2002-28
4. Doan, A. *et al.*: Learning to Match Ontologies on the Semantic Web. The VLDB Journal. **12** (2003) 303–319
5. Fan, H. and Poulovassilis, A.: Using AutoMed Metadata in Data Warehousing Environments. DOLAP03, (2003) 86 – 93
6. Garcia-Molina, H. *et al.*: Database Systems - The Complete Book. Prentice Hall (2002)
7. Halevy, A. *et al.*: Crossing the Structure Chasm. CIDR-2003, (2003) www-db.cs.wisc.edu/cidr/program/p11.pdf
8. Jones, A. *et al.*: Proposal for a Standard Representation of Two-Dimensional Gel Electrophoresis Data. Comparative and Functional Genomics. **4** (2003) 492–501
9. Kurgan, L. *et al.*: Semantic Mapping of XML Tags Using Inductive Machine Learning. ICMLA02, (2002) 99–109
10. Lacroix, Z. and Crichlow, T. (ed): Bioinformatics. Managing Scientific Data. Morgan Kaufmann (2003)
11. Madhavan, J. *et al.*: Generic Schema Matching with Cupid. VLDB01, Morgan Kaufmann (2001) 49–58
12. Madhavan, J. *et al.*: Representing and Reasoning About Mappings Between Domain Models. AAAI/IAAI-02, AAAI Press, (2002) 80–86
13. McBrien, P. and Poulovassilis, A.: A Semantic Approach to Integrating XML and Structured Data Sources. CAiSE01, Lecture Notes in Computer Science, Vol 2068 (2001) 330–345
14. McBrien, P. and Poulovassilis, A.: Schema Evolution in Heterogeneous Database Architectures, A Schema Transformation Approach. CAiSE02, Lecture Notes in Computer Science, Vol 2348 (2002) 484–499
15. Rahm, E. and Bernstein, P. A.: A Survey of Approaches to Automatic Schema Matching. The VLDB Journal. **10** (2001) 334–350
16. Safran, M. *et al.*: Human Gene-Centric Databases at the Weizmann Institute of Science: GeneCards, UDB, CroW 21 and HORDE. Nucleic Acids Res. **31** (2003) 142–146
17. Tanabe, L *et al.*: MedMiner: an Internet Text-Mining Tool for Biomedical Information with Application to Gene Expression Profiling. BioTechniques. **27** (1999) 1210–1217
18. Yan, L.L. *et al.*: Data-Driven Understanding and Refinement of Schema Mappings. SIGMOD Record. **30** (2001) 485–496