A Data Cleaning and Annotation Framework for Genome-wide Studies

Ranjani Ramakrishnan

A thesis presented to the faculty of the
OGI School of Science & Engineering
at Oregon Health & Science University
in partial fulfillment of the
requirements for the degree
Master of Science
in
Computer Science and Engineering

November 2007

The thesis "A Data Cleaning and Annotation Framework for Genome-wide Studies" by Ranjani Ramakrishnan has been examined and approved by the following Examination Committee:

Dr. Shannon McWeeney
Assistant Professor
Thesis Research Advisor

Dr. David Maier
Professor
Portland State University

Dr. Deniz Erdogmus
Assistant Professor

Acknowledgements

I would like to thank my thesis advisor Shannon McWeeney for her guidance in identifying, refining and working on this problem. I am grateful for her encouragement and support in finishing this thesis. I am deeply grateful to my committee members Drs. Dave Maier and Deniz Erdogmus for their feedback and their encouragement. I am grateful to all my friends at the Oregon Cancer Institute and the members of the ISR Dev group for their encouragement. I am deeply indebted to my parents and my sister for their support and encouragement as I was working on this thesis. Last, but not least, I am grateful to my husband Srikanth Venkataraman for all his love and support. I dedicate this thesis to Dosie, the reason I get up in the mornings.

Table of Contents

List of Tables

List of Figures

Abstract

A Data Cleaning and Annotation Framework for Genome-wide Studies

Ranjani Ramakrishnan

M.S., OGI School of Science & Engineering
At Oregon Health & Science University

November 2007

Thesis Advisor: Dr. Shannon McWeeney

Genome-wide studies are sensitive to the quality of annotation data included for analyses and they often involve overlaying both computationally derived and experimentally generated data onto a genomic scaffold. A framework for successful integration of data from diverse sources needs to address, at a minimum, the conceptualization of the biological identity in the data sources, the relationship between the sources in terms of the data present, the independence of the sources and, any discrepancies in the data. The outcome of the process should either resolve or incorporate these discrepancies into downstream analyses. In this thesis we identify factors that are important in detecting errors within and between sources and present a generalized framework to detect discrepancies. An implementation of our workflow is used to demonstrate the utility of the approach in the construction of a genome-wide mouse transcription factor binding map and in the classification of Single nucleotide polymorphisms. We also present the impact of these discrepancies on downstream analyses. The framework is extensible and we discuss future directions including summarization of the discrepancies in a biological relevant manner

Chapter 1 Introduction

1.1 Data Cleaning and Discrepancy Detection

Data cleaning, also known as data scrubbing, is defined as the process of detecting and removing errors and inconsistencies present in data sources [1]. Typically, the data cleaning process involves working with pre-existing data sources, the aim being to provide a reconciled view of high quality data. This involves detection of errors that occur within individual sources and those that arise due to discrepancies between the different sources. Errors that occur within an individual source are primarily due to missing information, invalid data and typographical errors. Discrepancies present between sources can be due to contradictory data in the sources, semantic mismatches between the data models, different naming conventions for the sources [2] and, the granularity of the data present in the sources. The types of discrepancies detected are dependent on the data sources being integrated and resolving the discrepancy or classifying it as an error will require meta-information about the data source, the relationships between the sources.

The data cleaning process is a multi-step process and is the first step in integrating data from diverse sources. Any errors missed at this step have the potential to affect the quality of the integrated data, and subsequently, any analyses that are based on the integrated data. Data cleaning, in a conventional sense, results in the resolution of discrepancies detected. However, in the biological domain, sometimes it is not possible to resolve a discrepancy with the existing or available information. For this reason, instead of making a decision with missing or insufficient information, it becomes imperative to carry forward the possible outcomes, each with an associated confidence measure. A confidence measure is a synthesis of a number of factors – including weighting a piece of information based on its source, the reliability of the source, etc. A systematic manner of

identifying discrepancies and either resolving them with some measure of confidence or flagging them for the user and tools that use the data is an important step in the integration process.

1.2 The Need to Detect Discrepancies in Genome Annotations

Genome annotation refers to information about a sequence, its biological function and role. Genome annotation refers both to the process, and the end product, by which the structural and functional class of a sequence is assigned. A sequence, in this context, refers to a string of nucleotides or amino acids. This raw data is further processed to identify sub-sequences that have biological relevance – such as CpG islands, gene boundaries in the case of DNA, and motifs such as the leucine zipper, in the case of protein sequences. This processing step can be via manual curation, computational predictions or a combination of both. Once a sequence is annotated with its role and function, it is also referred to a *genomic feature* or simply *a feature*. Annotations form an important component of the genome databases. With the proliferation of scientific databases and data warehouses, the issue of data provenance (i.e., where a piece of data came from and the process by which it arrived in the database) is crucial to ensure the accuracy of data [3, 4]. An estimated lower limit of errors in functional annotation of the large-scale sequencing projects is 8% [5]. This lower limit includes the annotation of *known* features in the genome. In general, it is estimated that 70% of the annotation *predictions* are correct which means that approximately 30% of the features are incorrectly predicted or assigned [6, 7]. Additionally, these annotations are used as the starting point for more complex features such as computational predictions of genes. By not including the uncertainty present in the input, the confidence in these models maybe inflated artificially. Using these gene predictions in annotation pipelines leads to propagation of these errors. If dubious functional assignments and annotations are used for subsequent predictions, errors will proliferate and lead to a "database explosion" [7], i.e., the integrity of the data is seriously compromised and cannot be relied upon.

The effect of poor data quality on subsequent analyses becomes apparent immediately. Structurally, a *gene* (Fig.1) is an ordered sequence of nucleotide

bases that encodes a product (this product could be just RNA, such as rRNA, or a protein). The gene includes, however, regions preceding and following the start of the coding region, the 5' *untranslated region* (UTR) and 3' UTR respectively, as well as (in eukaryotes) intervening sequences (*introns*) between individual coding segments (*exons*). The gene start includes the 5' UTR and the *transcription start site* (TSS) in some definitions. In others, the start of the transcript (not including the TSS) is defined as the start site. A similar discrepancy in definition arises in definition for the end of the gene. The end of the longest transcript is one candidate. Other definitions extend beyond the transcript and include the 3' UTR of the gene. In addition to the differences in definitions, the uncertainty in the experimental methods to identify and locate the transcripts and the UTRs adds an extra layer of complexity. Such differences in definitions result in discrepancies between sources. Not including these discrepancies in predictions and the inherent biases of the experimental techniques reduces the confidence in the prediction of gene locations. Now consider the case of life scientist who is interested only in *Single Nucleotide Polymorphisms* (*SNP*) that are located in the coding region of a gene. If there is no consistency across the annotation sources regarding the start and stop locations of a gene, the scientist may come up with different answers of whether the location of a given SNP is within the gene or external to it. If the existing data gives inconsistent answers, it becomes important to present the discrepancy to the user. To understand how pervasive this problem is, we carried out a PubMed search for the very specific term "Genome annotation". The number of articles published in the last ten years is around 2224, with 392 articles published in the last year.

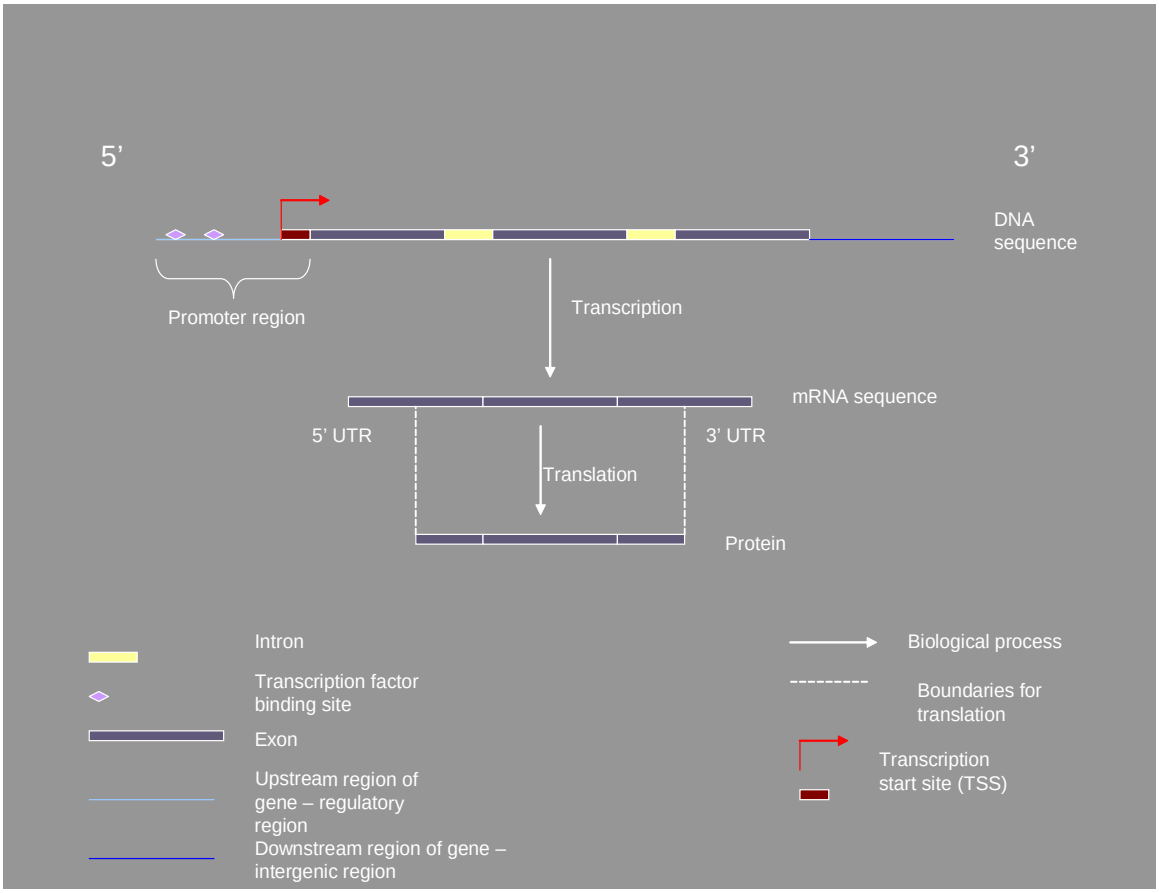

Figure 1. Schematic representation of the structure of a eukaryotic gene and the role of the different portions in coding the protein product.
The structure of the gene is defined at the level of a DNA sequence and is described from the 5' to the 3' end. This orientation is based on the physical orientation of the DNA bases. A eukaryotic gene consists of portions that code for a product (exons) interspersed with portions that do not code for any product (introns). The DNA sequence is coded into an RNA sequence by the process of transcription and in some cases into protein by the process of translation. The process of transcription is dependent on the binding of transcription factors upstream of a gene and other factors. The 5' vicinity of the gene is known as its promoter region. The 3' UTR is the portion of an mRNA from the 3' end of the mRNA to the position of the last codon used in translation. 5' UTR is the portion of an mRNA from the 5' end to the position of the first codon used in translation.

1.3 Current approaches to representing discrepancies and errors in the data

There have been a number of frameworks proposed for data cleaning [2, 8-10]. In the larger data-cleaning community, the actual representation of errors or discrepancies is a peripheral problem. The focus, however, in all these approaches has been towards reconciling any discrepancies present in the data. Workflows and transformations are

designed to help transform the data from the 'dirty" to the clean state. It is important to note the underlying assumption that the true nature of the entity being cleaned is known thereby helping the transformation into clean data possible. In contrast, in the biological domain, it is not always possible to obtain true state of an identity. The reason for this is primarily because the context in which the entity is being measured or observed is important. Often, this vital piece of information is not recorded or incomplete. For this reason, simple transformations for data cleaning are insufficient.

Domain-specific solutions, however, have been presented. In the biological domain, one approach classifies the errors based on the data production process [11]. The different classes of errors are i) experimental errors which arise from experimental setup failure or systematic errors, ii) analysis errors that arise due to misrepresentation of information, iii) transformation errors which arises when transforming data from one format to another, iv) propagated errors that arise when erroneous data is used for generating new data, and v) stale data when changes to the base data are made.

In a second approach, the authors have defined classes of discrepancies between two versions of a data source and have ranked the discrepancies based on their level of impact [12].

1.4 Drawbacks in the current approach in determining errors and representing them

Note that the process of classifying discrepancies based on the level of impact in the second approach [12] compared re-annotated data with the original annotation. The primary motivation for the re-annotation of genome data is to include up-to-date information on genes and proteins. It also provides new information to users by using improved techniques and algorithms. It may be more difficult to apply this system of classification to different annotation sources. The first approach is based on the data production methods and versioning information. When integrating multiple sources of pre-existing information, the exact process of generating a piece of data may not always be available, especially in the case of manual curation, where certain publications that do not support the data are examined but not included as being part of the decision-making process. We believe that it is imperative to include other meta-information about a source

in classifying discrepancies. In particular, the level of dependence between data sources is an important piece of information. A *discrepancy* only identifies that there is a difference in two sources. An *error* identifies which of the sources contains the correct information. An error provides more information to the user. If the data in two sources is dependent (the definition of a dependent source is presented later), a discrepancy can be classified as an error. Additionally, if one source is a primary source and the second source derives from it, the source of the discrepancy can be pinpointed more accurately, although the classification as an error may require more information.

Also, the provenance of a piece of data is useful in discrepancy resolution. Information, such as the lineage, of a data source is referred to the *meta-data* of a source in this thesis. The term "annotation" is used to refer to the data pertaining to a biological sequence. The basis for an annotation is often not clear and computational annotations versus experimental evidence must be distinguished. However, it is not enough to store whether the assignment is experimental or computational. It is also critical to store the store the source of the annotation as well. For experimental assignments, one needs to store the appropriate references. For computational assignments, the source of the functional assignment would include the group or individual who made the annotation, as well as the method and parameters used for the algorithm. Recording the provenance will allow individuals to update records annotated by that source or based on that annotation if it turns out to be erroneous (i.e., allow propagation of corrections). The STRING database, a precomputed resource of protein-protein interactions, provides relevant meta-information [13]. In addition to the results of functional links of a protein (presented with its associated confidence measures), users can navigate and explore the evidence that contributes to the results presented. As our knowledge domains become more composite in nature, with analyses being conducted on processed or computed data types, data lineage tracking is vital for error correction and data cleaning. As seen with STRING, annotation and meta-data add additional layers of complexity for genome-wide studies.

In cases when it is not possible to resolve discrepancies, the issue reduces to handling conflicts in the data (i.e., which one to choose). Deciding which piece of data to use is often seen as political and controversial. However, if a confidence measure is assigned to a gene's annotation and pointers to alternate annotations are stored, it should

be possible to handle multiple sources of annotation. Just as quality measures are suggested for sequence data [8], similarly there should be quality assessments of annotation. Suppose three groups annotate the same genomic region on Chromosome 12 in humans. If all three annotations agree, the annotation could be given a confidence measure of 100%. If none of the annotations agree, the confidence for each annotation could be 33%. Pointers to all three annotations would be stored, as well as how the annotations were generated. As seen with the STRING resource, more sophisticated algorithms for calculating the confidence in each result can be computed and presented to the user.

1.5 Problem Statement

These problems in data integration, lineage tracking, and annotation were the motivation for this work. At the heart of this approach is integrating the distributed information relevant for the analysis in a meaningful fashion and identifying *errors* in the data that may potentially affect any downstream analysis. The aims are to identify the factors that are useful in detecting discrepancies between sources, to identify the subset of discrepancies that will impact downstream analyses, and finally to classify the discrepancies as errors. The work presented in this thesis addresses the first two components. We identify the main components necessary for Heuristically Identifying Discrepancies/Errors (HIDE) affecting analyses.

The issue of integrating data from multiple annotation sources is crucial for a number of problems within the biological domain. In this thesis, we examine discrepancies in data sources critical for the positioning of experimental data, generated by high-throughput (HT) mapping of transcription factor (TF) binding sites, on the genome scaffold. The placement of these features relative to known genomic landmarks, such as genes, is critical both for validating the data and for the identification of new regulatory sites. Errors in the data sources that describe the location of the landmarks can affect the validation and discovery of such sites. Hence it is important to identify any discrepancies present and identify the subset that can affect downstream analyses.

The data integration scenario for this use case deviates slightly from the traditional scenario. Firstly, the data to be integrated is not from a single organization,

but is drawn from a number of possibly independent sources. The data present in the majority of these sources is updated frequently. Because of the nature of the data being integrated (biological), there is no consistent definition of an entity across the sources, making it difficult to isolate one source as being incorrect. The task of integration is further exacerbated by the differences in the experimental methods used to generate the data. Different techniques provide data of varying granularity and have varying confidence levels associated with the data. Also, since the data is being used in a HT scenario, there is a lack of intimate knowledge of a large percentage of the regions of the genome being examined. For this reason, the traditional approach of creating a clean data set is not practical as the entire process will need to be repeated with every new update. It becomes imperative to flag the discrepancies so that they can be incorporated in the downstream analyses. To obtain a quick manner of identifying discrepancies, we chose to use the factors that are relatively invariant between the sources, primarily the relationship between the different sources, to create links between the sources. Additionally, we introduce the concept of consequential discrepancies that will help the user identify and focus on the discrepancies that will impact downstream analyses.

The assumption is that the type of relationship between the sources limits the kinds of discrepancies between them. HIDE attempts to exploit this assumption by directing the types of tests that are needed to detect discrepancies, by identifying the relationships between the sources, i.e., by determining their context. The context is determined by the lineage relationship of the sources and the relationship of the data present in each data source. A series of definitions essential for defining HIDE, followed by a description of the essential elements of the framework is presented in the next section. A data integration scenario, derived from the genomic domain, to demonstrate the coverage of errors provided by the framework is presented. We then map back the errors detected during the data integration onto the framework. The following chapters describe the framework, a preliminary implementation of the workflow and possible extensions.

Chapter 2 A Primer on Molecular Biology for Computer Scientists

This chapter further elaborates on the topics that were introduced briefly in Chapter 1. Although not comprehensive, it emphasizes the concepts that are relevant for the use case, presented in subsequent chapters.

2.1 DNA, RNA and Proteins – Elements of biological function

The biological identity of a cell is determined by the presence of proteins and gene products at specific amounts, specific locations and at specific times during the life of the cell. The genetic information is contained in the sequence of de-oxy ribonucleic acid (DNA). DNA is then "transcribed" to form ribonucleic acid (RNA), which in turn is translated into the protein product. Transcription is the process by which the portions of a gene (Fig. 1) that code for product (but not the regulatory portions) are synthesized using the DNA as template. The process of translation refers to the formation of a functional protein product using the RNA as template. This process of information flow (from DNA to protein) is known as the Central Dogma (Fig. 2). As a result, the formation of these products at the appropriate times, locations and quantities is regulated and the control at the different stages in the lifecycle of a cell is diverse but coordinated.

The type of regulation is often described by the immediate process that it regulates or facilitates. For example, regulation of the RNA is controlled by transcriptional regulators and formation of protein is controlled by translational regulators. One such regulator of transcription is a transcription factor (TF). A TF is a specialized protein complex that binds to the DNA sequence at a specific location and recruits other proteins that help form the RNA from the neighboring DNA sequence.

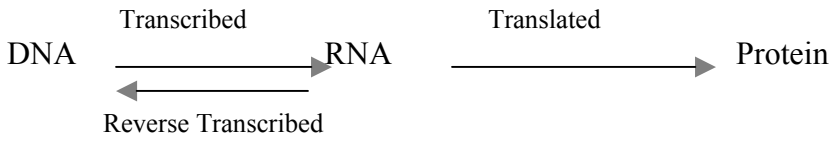

Figure 2 Simplistic depiction of the Central Dogma.

There are a number of TFs, each of which regulates a subset of the total genes in a cell. To understand regulation at the transcriptional level, it is important to identify the subset of genes that can be regulated by a particular TF. One way in which transcriptional regulation can be studied is by locating all possible binding sites and looking for genes near these binding sites. The potential binding site for the TF under study can also be extracted from these sequences. It becomes important to obtain unbiased data on TF binding prior to identifying the binding site and identifying genes regulated by the TF. One such technique is the Serial Analysis of Chromatin Occupancy (SACO). Unlike other techniques (e.g.,ChIP on chip, computational predictions) it does not use any previously known information about the site of TF binding. It also decouples the generation of the TF-binding data from the location on the genome until later in the analysis stage. For the reasons above, it is unique in providing an unbiased view of the TF binding sites. Other techniques used for TF binding studies, such as ChIP on chip, make assumptions about the length and or composition of the transcription factor binding site. The process for generating the TF map is described in detail in the next chapter.

2.2 Elements of Variation in DNA – SNPs

SNPs (single nucleotide polymorphisms) are a variation in a single base of DNA, compared to the expected base at that location [14]. They form one of the most common variations in DNA sequences and are typically detected by comparing sequences of DNA obtained from multiple individuals. On average there is about 1 SNP present in every 300 base pairs of the genome, at the population level. The majority of SNPs are located in areas that do not code for protein products, since the protein coding regions typically form 3-5 % of the entire genome.

A number of researchers believe that the common variants in the genome are important for explaining the risk of disease in populations. If this common-variant theory is true [15], SNPs - being the most common variant - are important to characterize and locate. SNPs that are located in the coding region of the genome are of particular interest as they may play a role in changing the functional role of the gene product by mechanisms such as changing the function of the protein. Determining functional roles for SNPs will also impact the drug discovery process. If a SNP impacts the role of a protein, and by extension the underlying biological process, it serves as a starting point for drug discovery by helping to understand the biological roles and identifying potential drug targets. Incorporation of SNP data will also be important for determining the efficacy of drugs, especially if they target the products affected by SNPs.

Chapter 3 Problem Addressed - SACO

3.1 The Biological Context for Discrepancy Detection

SACO is a novel, high throughput technique that generates a genome-wide map of TF binding sites [16]. The placement of the experimental data on the genome requires a complete sequence of the organism. Definitions for biological terms introduced in this section are provided in the Glossary. It uses a combination of *chromatin immunoprecipitation (ChIP)* and long *Serial Analysis of Gene Expression (SAGE)* to generate this comprehensive map (Fig. 2).

Briefly, the process is as follows. Genomic DNA is bound to the transcription factor (TF) of interest and these binding sequences are obtained via experimental techniques [16]. The bound sequences are screened and sequenced and used to form a *Genomic Signature Tag (GST)* library. A GST is a sequence of nucleotides and is approximately 21 base pairs in length. A computationally derived library of *potential* GSTs and their locations in the genome is generated by an *in silico* digest of the genomic sequence. This process entails treating the entire genome as a string, looking for a particular motif in this sequence and extracting 16 bases flanking the motif sequence. These flanking sequences form the library of potential GSTs. The motif is determined by the restriction enzyme (RE) cutting site (a pattern of CATG). GSTs which are located within a 1000 bps of one another are clustered based on the assumption that they are functionally related.

GST clusters in the experimentally derived library are compared to the computationally derived library and their locations on the genome mapped. The relative position of these tags with respect to genomic landmarks - mRNAs, CpG islands, etc. - is then examined. The overlay of annotations, drawn from a number of sources, on the

genome is critical to the success of this technique. Any discrepancies in the annotations will correlate with the uncertainty both in the placement of the transcription factor binding sites and in the characterization of the TF. Because a windowing approach is used with a size of 1000 bps 5' to the start of a gene, any discrepancies in the start of a gene can impact the area of the genome under study. Additionally, discrepancies in the start position can impact the classification of the GST as internal or 5' to a gene. Discrepancies between annotation sources need to be identified and reconciled in a systematic fashion. Such systematic identification of errors between sources will help in streamlining and standardizing the annotation process and in the process improve the quality of data. The aim of the data integration process was to identify discrepancies present between the sources and see what impact these discrepancies had on classification of GST clusters.

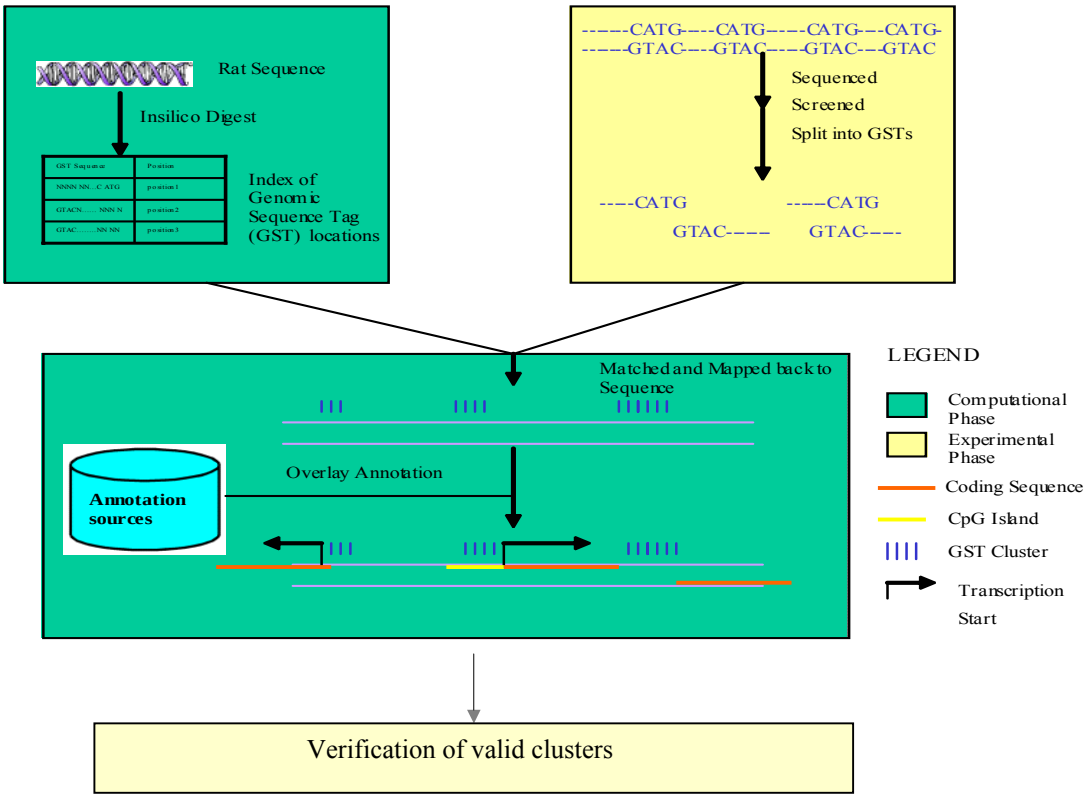

Figure 3. Current Bioinformatics pipeline for SACO.
Computational phases are highlighted in green and the experimental phases in yellow. Genomic Sequence tags (GSTs) generated experimentally are matched with in-silico generated tags and placed on the genomic scaffold. The GSTs are anchored by sequence features such as the start and stop of genes and placement of EST tags, which are derived from the annotation sources. The last step is the experimental verification of the GST locations.

3.2 Data sources used for SACO analysis

Five sources, critical to providing information about the location of landmarks on the mouse genome (mm5), were used as the sources for data integration. These were the UCSC repository [17] for mRNAs, ESTs, CpG islands and miRNAs, RNAdb [18] for noncoding RNA (ncRNAs) , ECGene [19] and Ensembl [20] for genes and the list of ultraconserved elements from UCSC [21]. The definitions for these terms are provided in the Glossary. The rationale for including these annotations is because each of them helps to identify the position of genes and location of regulatory elements. ESTs, genes and

mRNAs can help localize the presence of the genes. CpG islands, miRNAs and ncRNAs can help determine regulatory regions. By using these sources we aim to understand the TF of interest, specifically, which genes it regulates, whether it regulates other regulators and, if so, can the mechanism of regulation be determined.

The UCSC repository is a data warehouse that provides information on a number of genome landmarks. The primary annotations used for this study included the mRNA, EST, CpG island and miRNA features. The repository essentially provides the position(s) of these elements, for a given build of a genome and the species. Sequence information for each of these features is pulled from GenBank and the sequences aligned against the current genome build using the Basic Alignment Tool (BLAT). When a feature maps to multiple locations, the alignment with the highest base identity is determined. Alignments within 1% of the best alignment are retained. The data is available in the form of flat files and is derived from the MySQL database that houses all the annotation data on UCSC servers. SQL statements for the original data tables are also available with the flat files.

RNAdb contains information on 800 experimentally studied non-coding RNAs and includes miRNAS and small nucleolar RNAs (snoRNAs). There are three datasets within this repository. Fantom2 contains more than 15,000 unique, putative ncRNAs from the Functional Annotation of Mouse (FANTOM) project. Information on these sequences includes their sequence, GenBank accession, chromosomal location, transcript length, splicing status, EST hits, antisense relationship and the experimental library from which the entity originated. The sequence is a representative for the entire set of related sequences and is chosen by the Fantom2 team. We found that to obtain the actual information about the other members in the group, we had to check the parent source, Riken. RNAdb data is available in XML format with its associated schema. The XML data is from a Microsoft SQL2000 database running on a .NET platform.

ECGene is a data source that contains predicted gene models. Using mRNA, EST and protein sequences as inputs to the BLAT algorithms and using it in combination with graph theoretic analysis, gene predictions with varying confidence measures are made. The confidence measures for the different models are based on experimental evidence such as Refseq sequences associated with the model, the number of clones, mRNAs and

ESTs etc. The data is available in the form of flat files, which are derived from a relational database in the backend.

Ensembl is an integrated resource that uses a combination of automatic and manual techniques in its pipeline to generate genome annotations. For this project we used the list of known genes (experimentally verified genes) from Ensembl to provide us with information on the gene boundaries as well as the intron-exon boundaries. Ensembl data was pulled out from the core database using the BioMart data mining tool. The data was exported as a tab-delimited file prior to its being used for analyses.

The source for ultraconserved contained nucleotide sequences at least 200 bases in length that are conserved between orthologous regions of human, rat and mouse genomes. Nearly all of the segments are also conserved in chicken and dog genomes. In humans these sequences typically overlap exons of genes involved RNA processing or are present in introns or nearby genes involved in regulation of transcription and development. These are genetic elements whose function is yet to be determined. The raw sequence data (in the form of a text file) was available for download.

Chapter 4 Methods

In this chapter we propose a series of definitions that are necessary for describing HIDE. We then list factors that we identified as being important for detection of discrepancies. The final sections describe how we propose to help identify discrepancies in data, followed by a description of the implementation.

4.1 Definitions proposed for describing the data workspace

We use the following terms in describing and designing the HIDE

Expectation: Conceptual model of an entity with respect to its behavior and properties, based on the physical world and the existing knowledge base. An entity can be modeled formally in a number of ways such as using an ontology or within the schema of a database.

Error: Any deviation in behavior or properties from our expectation(s) about the entity.

Primary data source: Data that is the result of an experiment or computational prediction and that cannot be traced back to another source.

Secondary data source: A data source that is not a primary repository. A subset of the data in this repository is *derived* (obtained) from one or more primary source(s).

Direct mapping: Data elements that are common to the two sources being compared, i.e., the same data is present in both sources.

Indirect mapping: Different data are present in two sources; one source can support or contradict data in the other source.

Independent data sources: Two data sources are said to be *independent* if there exists no evidence to indicate a common lineage or in the case of shared ancestry, if a change in the data in one source does not affect the data present in the other source [22].

Consequential discrepancies: A subset of all discrepancies present between data sources. Defined relative to an analysis, it is the subset of discrepancies that may impact the result of the analysis.

To illustrate these definitions, consider the following example. Data on a particular business is obtained from an advertisement in the yellow pages and from the company website. Both can be considered as being derived sources, obtaining the information from the business (which is now the primary data source). The business (entity) in the two sources is linked by its name. If both sources provide the telephone numbers and addresses, these attributes can be directly mapped in the two sources. Our expectation is that these attributes are equal in value. If they are different, we state that there is a discrepancy and we need more information (meta-data) to resolve this discrepancy. One way of resolving the discrepancy and identifying the incorrect source (the error) can be done by contacting the primary source. Now consider the case where the advertisement states the business is open for seven days a week and the website provides hours only for Monday through Saturday. This is an example of an indirect mapping. The two sources in this example are not independent and are related through shared ancestry. Consider the case where the business changes its hours and this is reflected only in the website. This impacts the data in the advertisement, if it is not updated.

4.2 Various factors included for discrepancy detection

Independence and the mapping relationship between two sources are concepts that are important and have an effect on discrepancy detection. As defined above, independence of two sources exists in the absence of common lineage and also in certain cases in sources with shared ancestry. When two sources are **not** independent and share a common lineage, the expectation is that the sources share a common model of an entity i.e., it is valid to expect that both sources represent the entity of interest using the same data model. Hence any discrepancies present between the sources can be attributed to errors or deliberate changes, as opposed to differences in the underlying models.

The second concept that affects discrepancy detection is the mapping relationship that exists between two sources. When two sources present the same information, i.e., a direct map exists, we expect that the data will be the same for an entity. When the information is indirectly mapped, tests for criteria other than equality are required to check for discrepancies. Defining the relationship between the two sources is important for identifying the test to be used.

4.3 Detecting errors within data sources

To detect errors within sources, it is necessary to carry out both semantic and syntactic checks. These tests are typically carried out by using a combination of queries to capture the semantics embedded in the data and the use of data cleaning tools. These queries used in this step were designed using the possible list of queries described for within sources (described in Appendix A). As there were no errors detected within each source, a mapping was generated between the different sources as a two-step process – identifying the entities (and their associated properties) common to sources and by tracing the lineage of each source with respect to the other sources (Fig. 4). Based on the sources (and data therein) that we are integrating, we generated 2-way mappings. But it is obvious that such a mapping can include more than two sources. For the sake of simplicity, we do not include any circular (when a derived source serves to feed information back to the primary source) and transitive lineages in this data. Independence and lineage of sources are with respect to a particular set of attributes that are of interest to the user. It is not hard to imagine a derived source where the origin of subsets of attributes varies.

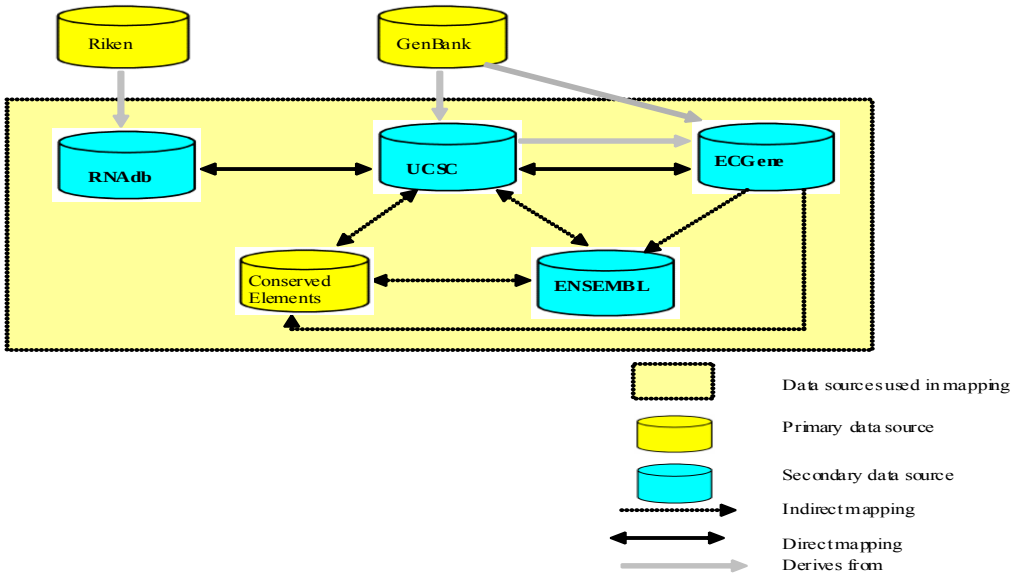

Figure 4 Relationships between the different data sources.
Data sources in the highlighted area were used for the preliminary data analyses and to design the workflow. The two sources (GenBank and Riken) outside the highlighted area were used to resolve discrepancies detected. Data sources colored blue are primary sources and the ones colored yellow are secondary sources. Three types of relationships are defined between pairs of data sources (for the attributes of interest) – derives from, direct mapped and indirect mapped. Information for creating these mappings were obtained from the documentation provided for each source[17-21, 23-26]. The websites were viewed in June 2005.

4.4 Description of HIDE proposed for discrepancy detection between sources

Our workflow for discrepancy detection is a process to help identify errors that can arise during data integration by directing testing. It is modeled as a series of steps to follow to identify the context in which two data sources are integrated. The context then determines the types of tests to be carried out, and in turn helps identify discrepancies. Given two or more data sources, the first step is to make sure each is internally consistent – syntactically and semantically. This check is typically achieved by using data cleaning tools or queries designed to detect errors, or both. Any errors that are detected at this stage will affect the results of the data integration process and hence need to be flagged.

The next step is to determine if the sources share a common ancestry. This condition includes the case when one source is a parent (or ancestor) of another source. When this status is unknown, or if there are no obvious relationships between the data sources, a test for hidden dependencies is required. This situation is detected by

answering the question "If the data were different in one source, would it have resulted in the data being in different state in the other source" [22]. One way to answer the above question would be to start at one source, envision a change in the data and trace the effect to the other source. If yes, then we can conclude there are hidden dependencies between the sources. The series of steps to identify the relationship between sources helps determine the context in which the queries will be formulated (Fig. 5).

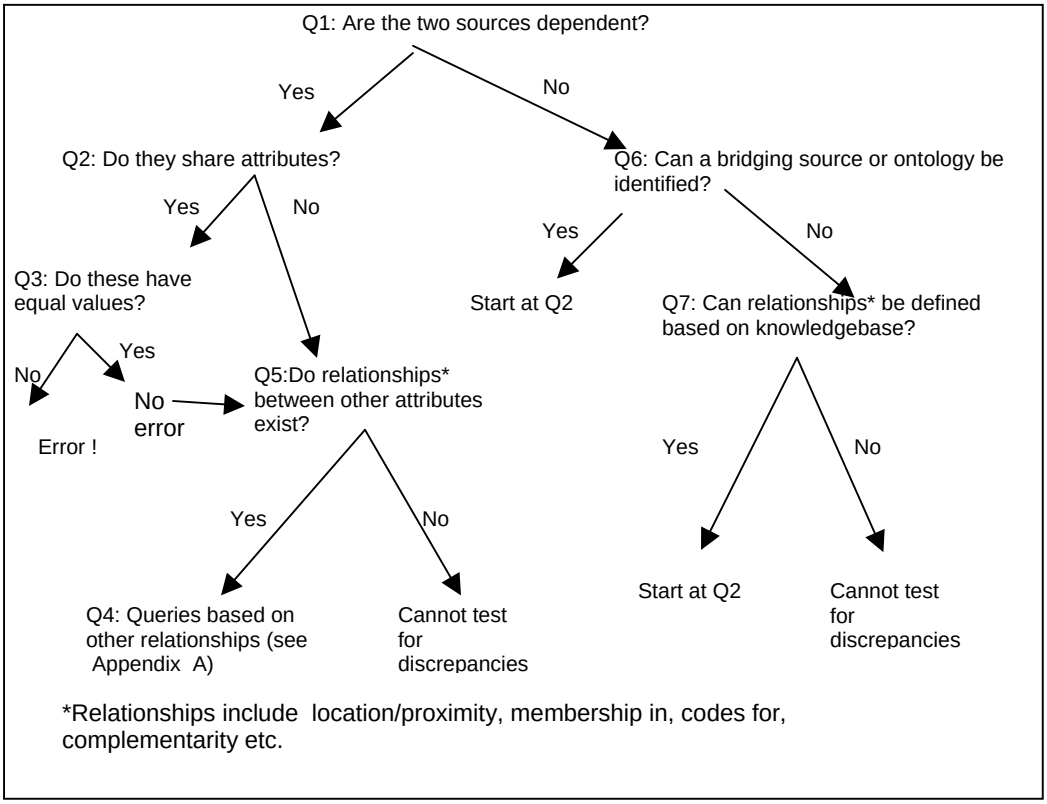

Figure 5. Schematic of the workflow to detect discrepancies and errors present between data sources. The workflow specifies the actions and the order in which they need to be executed to identify the relationships between sources and based on these relationships, helps identify sample queries that can be generated.

4.5 Role of the context in detecting discrepancies

4.5.1 Dependent data sources

If dependence is confirmed, the next step is to determine if the two data sources can be directly mapped. If common attributes are identified in both sources, for the same

instance of an entity, we expect the common attribute (present in both sources) to have the same value both places, that is, the equality condition is true. A violation of this condition in a small percentage of cases implies that there is an error in at least one of the sources. If the majority of data is incorrect, this would suggest either an incorrect mapping or a misunderstanding of the data organization in at least one of the sources.

If the data in two sources can be indirectly mapped, it is necessary to determine how the individual pairs of attributes in the two sources are related, if such a relationship exists. The following are possible relationships that might exist between pairs of attributes: complement, not equal, subsequence (part of), co-localization or proximity with, codes for (derives from), member of (membership in). If one or more of these expectations hold, queries need to be designed to test for deviations from the expectation(s). The expectations could be a composition of these relationships. Errors are detected if an expectation is violated.

4.5.2 Independent data sources

Prior to defining the mapping between sources, the first step in integrating independent sources is to determine if the concept of an entity in the two cases is actually the same. Although this determination appears to be intuitive, we have found that integrating sources based on common and well defined identifiers (which are proxies for entities and their associated semantics) fails when the associated semantics are modified by individual data sources. Such modification invalidates the expectation that the entities are the same in both sources. It is important to resolve this "identity crisis", for the identification of a discrepancy is dependent on the conceptualization of an entity being similar across data sources.

Once the definition of an entity is resolved, the steps that follow are similar to the case of the dependent sources. Relationships between pairs of attributes are either tagged as directly mapped, indirectly mapped or are non-existent, and the appropriate tests carried out to detect errors.

4.5.3 Navigating the HIDE workflow utilizing the example of SACO

The first task in utilizing HIDE is to determine if the sources are direct mapped, which is a proxy for determining if there are common entities between the two sources. This determination is by using a combination of several factors and utilizes meta-data about the sources. The factors include the lineage of the sources, the level of overlap in the data between the sources, and use of common identifiers. For the SACO use case, the mapping was achieved by all three factors mentioned above. We examined the data described in each source, the lineage of the sources and the degree of overlap in terms of data between the sources. We found evidence for dependent and independent sources. For example consider the following dependent sources, UCSC and Ensembl. The sources are determined to be dependent and found to contain common attributes. This can be traced back to Q1 and Q2 being true in Fig.5. Both contain information about genes and utilize data from GenBank to create their repositories. Common identifiers, or - in their absence - bridging sources, were also available to answer Q5 in Fig. 5. In this case we utilized UCSC's mapping for linking to Ensembl (bridging source). We were able to identify common entities between the two sources.

In the case of ECGene and Ensembl, we could not establish any direct relationships (answer to Q5 in Fig. 5) between the dependent sources as there were no bridging ontologies or sources. This arose in part because data in ECGene was generated computationally and the results of the predictions were not linked to any experimentally verified genes. This mapping presented a case of hidden dependencies because if the mRNA and EST data were different in UCSC, it could affect certain categories of data confidence models in ECGene. For example, the medium confidence model requires evidence of at least 4 clones for a single exon gene (per ECGene website, June 2007). Even if one of those clones was not present in UCSC or contained different information, it could affect the confidence model in ECGene. This is because ECGene utilizes data from UCSC in its prediction of genes. Once the links were established and some dependency detected, we estimated the degree of overlap between the sources. The examination of sources for their level of overlap was done at the attribute level in the SACO sources. If the attributes were common, the values were tested for equality between sources. If there was a discrepancy between sources it was flagged. Other

relationships between pairs of attributes were tested. For example, UCSC provides all possible locations for a transcript. RNAdb, however, provides only one location for the same transcript. The constraint is that the one location specified by RNAdb has to be a member of the set of locations reported by UCSC. We found that for a subset of the data, the expectation did not hold. Results are presented in Chapter 5.

4.6 Implementation of HIDE in a discrepancy detection tool

To help a user detect discrepancies in data, we have implemented the workflow as a user-driven application using a MySQL back end. Data, in tab-delimited format, and meta-data in a specified format are used as initial inputs to the program. The user is then guided through a series of steps to create mappings between data sources. The initial data and the user's input are used to detect discrepancies, both within and between sources. Summaries of the discrepancies and the actual details of the discrepancies are available to the user in a text file that can be visualized within the genome browser. The discrepancy detection tool is implemented as a Java application. User interfaces were designed with Java Swing and the output, depending on the user's requirements, is either a summary of the discrepancies or a text file that can be visualized using the UCSC genome browser and allows the user to drill down into the details of the discrepancy.

4.7 Description of the tool: Implementation details and Current Functionality

The functionality of the tool can be divided into three broad components. The first deals with checking for deviations in data within each source, specifically the type and the length of fields, and is based on the meta-data provided. The second component consists of creating mappings between sources, based on user input, and creating queries to detect discrepancies between sources. A possible extension will allow the automatic generation of mappings and allow user to edit the mappings as opposed to creating the mappings. Queries will be customized based on these mappings. The final component is the visualization of the discrepancies. Again, based on user input, a domain-specific, detail-oriented view or a summary of discrepancies is available to the user.

4.7.1. Assumptions

In the preliminary implementation of the workflow, we assume that all the information needed to define the relational model of the entity and the data to create the mappings between sources will be provided by the user. This assumption is not unreasonable for the current use case as the data used is well-curated and the minimal information required is easily accessible to the end user of the system. Additionally, we do not incorporate independence criteria of the sources in our implementation as we currently don't distinguish between a discrepancy and an error. The independence information is required for the distinction. In the current implementation we restrict the output to detecting discrepancies between data sources, regardless of the level of independence between pairs of data sources.

4.7.2 Initial Inputs

There are two initial inputs that are required to start the discrepancy-detection process. The first file is a meta–data file. The file currently has to include information about the data source, the type of entity it describes, the date the file was created/downloaded, the version of the data source and each attribute of the entity present in the file – including the name of the attribute, the data type of the attribute, information about allowing null values. The meta-data file can also be used to specify keys – both simple and composite for a particular data source (Fig. 6). The second input file is the data file. Currently, the accepted format for the data file is a tab-delimited file with one row describing each instance of an entity. The order of attributes specified in the meta-data file needs to be preserved in the data file.

4.7.3 Process used

Data is loaded into temporary files and checked for discrepancies related to the data type, the length of the fields and, null values (when they are not allowed). Rows that deviate from the specifications provided in the meta-data file are written out into a separate log file and error messages indicating the columns with the errors are provided for the user with-in the workflow prototype. Currently, checks for simple data types are implemented within the tool.

After the check for errors within each source is completed, data is loaded into final tables that have been created, based on the meta-data provided. The user is allowed to create mappings between attributes in the different data sources. These mappings are then used to create custom queries to detect discrepancies between the different sources. For the current implementation, the types of relationships between pairs of attributes that can be checked are simple relations. These details of the discrepancies can be visualized within the UCSC genome browser which is a domain-specific solution that allows the user to visualize the discrepancies in the position of different genomic features in the context of a common scaffold. In addition, the discrepancies are summarized by the data source and the location on the chromosome. The discrepancies are presented to the user graphically.

```
NAME UCSCmRNA
VERSION may2004
DATE DOWNLOADED 5/12/2004
entityType mrna
# `bin` smallint(5)
# `matches` int(10)
# `misMatches` int(10)
# `repMatches` int(10)
# `nCount` int(10)
# `qNumInsert` int(10)
# `qBaseInsert` int(10)
# `tNumInsert` int(10)
# `tBaseInsert` int(10)
# `strand` char(2) NOT NULL
# `qName` varchar(255) NOT NULL default ''
# `qSize` int(10) unsigned
# `qStart` int(10) unsigned
# `qEnd` int(10)
# `tName` varchar(255)
# `tSize` int(10)
# `tStart` int(10)
# `tEnd` int(10)
# `blockCount` int(10)
# `blockSizes` longblob
# `qStarts` longblob
# `tStarts` longblob
```

Figure 6 Sample meta-data file. The meta-data information should include the name of the source, the version number and the date of the download. In addition, each attribute of the entity present in the file – including the name of the attribute, the data type of the attribute (these are restricted to SQL data types), any restrictions on the length of the field, allowing null values and any default values need to be specified. The order of attributes has to be the same as the data. The names of the columns are defined by the user and are used by the tool to report errors.

4.7.4 Output of tool

The output of the tool consists of text files and summary graphs. Within source errors related to errors in the data type are flagged in log file (Fig. 7). Incorrect records, determined by deviations from the data description provided by the user, are flagged in an error log (Fig. 8). Discrepancies detected in the position of the genome annotations are visualized within the UCSC genome browser (Figs. 9 and 10).

```
misMatches has rows with incorrect format for an int
tBaseInsert has rows with incorrect format for an int
Please refer to log file C:\tmpUCSCmRNAlog.txt with incorrect data
```

Figure 7 Sample error messages displayed within the tool to indicate that there are problems with the data types within one data source. This is part of the process to detect data errors within a source.

```
611     2468    0       abc     0
613     2468    0       abc     0
614     2468    0       abc     0
615     2468    0       abc     0
```

Figure 8 Sample error log containing all rows with data errors. Typically these are flagged for the user when there is a mismatch between the data type in the meta data file and the actual data.

```
##Example
browser position chr1:1-100250
browser hide all
track name=regulatory description="Regulatory Regions" visibility=2
chr1  demo  GST   3345  8745  .     .     .     .
chr1  demo  GST   4345  9745  .     .     .     .
```

Figure 9 The sample .gff file to be loaded into the UCSC browser. Here the discrepancy in a feature is the custom track that is to be loaded into the browser.

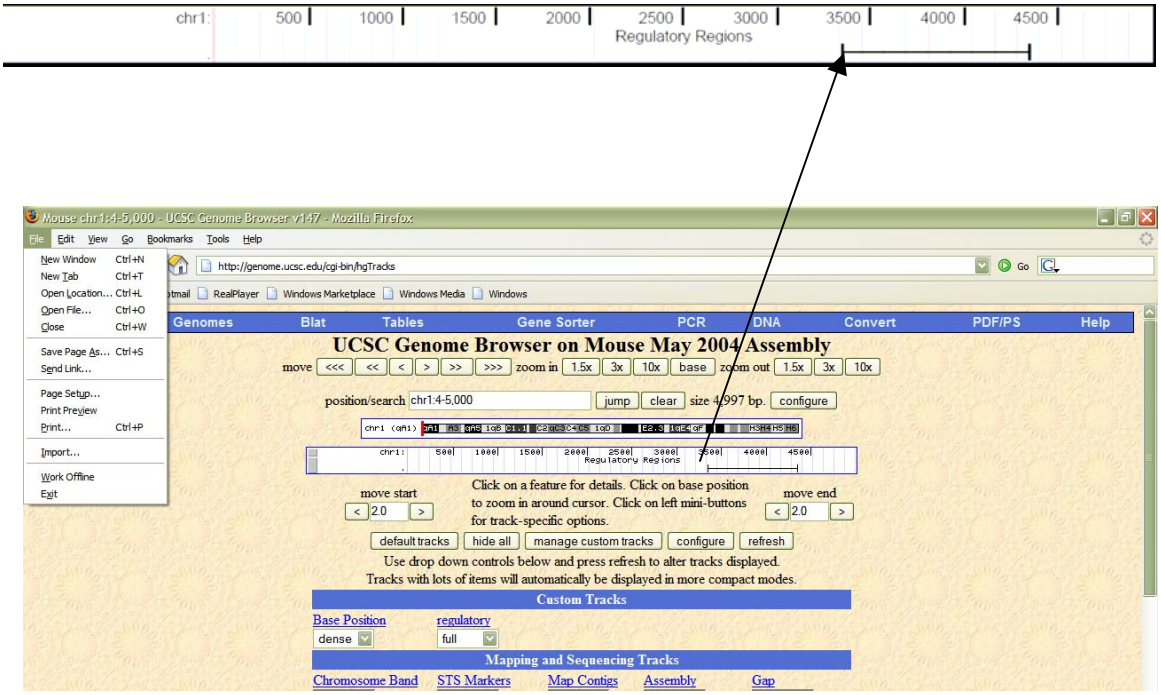

Figure 10 GFF file visualized in the genome browser. The discrepancy track is magnified and presented above the screen shot. Each discrepancy in a feature is represented as a tick mark along the chromosome.

Chapter 5 Results

5.1 Discrepancies detected in SACO using the workflow

The first category of discrepancies we detected was related to direct mappings between data sources, in particular mismatches in the length of mRNAs. We compared the sequences from RNAdb and UCSC, identified by their *accession numbers*. An accession number is a unique identifier that is associated with a *sequence record*. A sequence record includes information about the sequence, in addition to other information such as the date of submission, the name of the submitter, etc. The assumption is that the sequence associated with the accession number is the same across multiple sources. The first source, RNAdb, derives from the Riken source and the second source, UCSC, is derived from GenBank. We found that in UCSC, accession numbers are unique, are associated with sequences of a unique length and map to multiple locations. In RNAdb, each accession number was found to have only one location and sequence length. We found 23 (< 1% of RNAdb records) accession numbers, which were assumed to represent one sequence each, were discordant in the length of the mRNA with data for the same accessions from UCSC. This number may be an under-representation of the discrepancies as there could be sequences with the same length but different base composition.

It was also noted that twelve accession numbers in RNAdb have locations that are discordant with UCSC's predicted set of locations. The same discrepancies were detected between RNAdb and Riken (the primary source) (Fig. 11). On further examination, it was determined that for RNAdb the accession number is a representative accession for a number of clones. Each entry has a *consensus* length and sequence location. It then becomes clear that the number of sequences actually discordant can be much higher than detected as the representation of the sequence is not the same in the two sources. In particular, same length is not a guarantee that the same sequences are being compared.

Because RNAdb contains information only on the representative sequence, it is not possible to compile a list of all sequences associated with each representative and then compare this larger set to the sequences from UCSC.

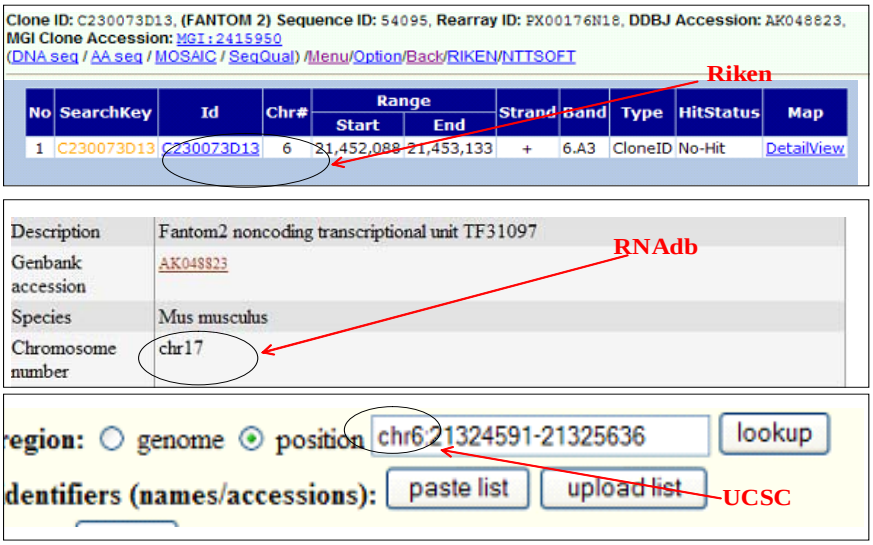

Figure 11 Discrepancies in the chromosomal location of clones in mouse based on pair wise direct mapping between data sources.
The chromosomal position for the GenBank Accession ID AK048823 in the three sources – Riken (top panel), RNAdb(middle panel) and UCSC (bottom panel). The curated location in RNAdb is not consistent with its parent source, Riken, or with UCSC.

A second category of discrepancies we detected were related to indirect mappings. We had lists of predicted (with high confidence) and known genes from ECGene and Ensembl respectively. In the absence of common identifiers, we were interested in seeing the degree of overlap between the genes in the two sources. We examined the number of genes with the same starts and stops. We used two very simple metrics, the number of genes in each source and the overlap of genes (based on the start and stop). We observe a very low concordance between the sources for both metrics (Table 1), which is expected since the underlying gene models are different. Also, we note that the exact matching of gene boundaries is not useful metric in this case because discrepancies arise due to differences in lengths of introns, repeat sequences, etc. Without mapping the

predicted genes in ECGene to an experimentally verified gene, it is not possible to obtain a better mapping of the two sources.

We propose more flexibility in designing the mapping parameter is required. For example, gene boundaries need to be compared within a sequence window and not exactly at boundaries.

Table 1 Discrepancies in gene number and overlapping gene boundaries when comparing and ECGene' s high confidence model with all genes from ENSEMBL.

Chromosome	ECGene High Confidence Model	Ensembl (All genes)	#Genes with common boundaries
1	5583	1422	2
2	7754	1793	1
3	4568	1180	1
4	5738	1498	1
5	5707	1397	3
6	5036	1329	1
7	7129	1921	2
8	4658	1170	0
9	5260	13115	0
10	4519	1110	0
11	7316	1797	1
12	3339	817	1
13	3379	923	0
14	3370	871	1
15	3718	910	3
16	3144	779	2
17	4319	1092	1
18	2515	618	0
19	3078	789	0
X	3109	1249	0
Y	220	225	0

5.2 Impact of discrepancies on downstream analyses for SACO

Of critical interest in the analysis of SACO data is the localization of the GSTs relative to annotations in order to categorize binding sites (i.e., upstream 5′, internal or 3′). Therefore, we examined discrepancies in the start position between two data sources: Ensembl (Release 41, using build 36) and UCSC's known gene tracks (using NCBI build 36).

To carry out the comparison, the same entities needed to be identified in both sources. At issue is the lack of a common identifier between the two sources and the fact that UCSC chooses only one reference transcript to represent the gene. For simplicity in this use case, UCSC's mapping to ENSEMBL was utilized. It should be noted that having a mapping to the other data source is potentially the best-case scenario.

With the mapping between the two sources provided, we examined the start and stop locations for the same genes from both sources. We expect that the attributes have the same values for both start and stop. Using the mapping conditions and check conditions specified above, we were able to identify a total of 4498 genes which had different start positions in the two sources (Fig 12). The absolute difference in the start position ranged from 1 to 8.4 x 10^7 base pairs (bps). The largest category (38.6 %) was genes that differed by less than 5 bps (of which 80% were only different by 1 bp, most likely due to differences in index notation). Most striking was the observation that 19% differed by more than 5kb. Because the positioning of SACO tags relative to annotation is critical, we examined the number of tags that would be classified as internal or 5′ upstream. For this simple example, 5′ was defined as 1kb upstream of the TSS.

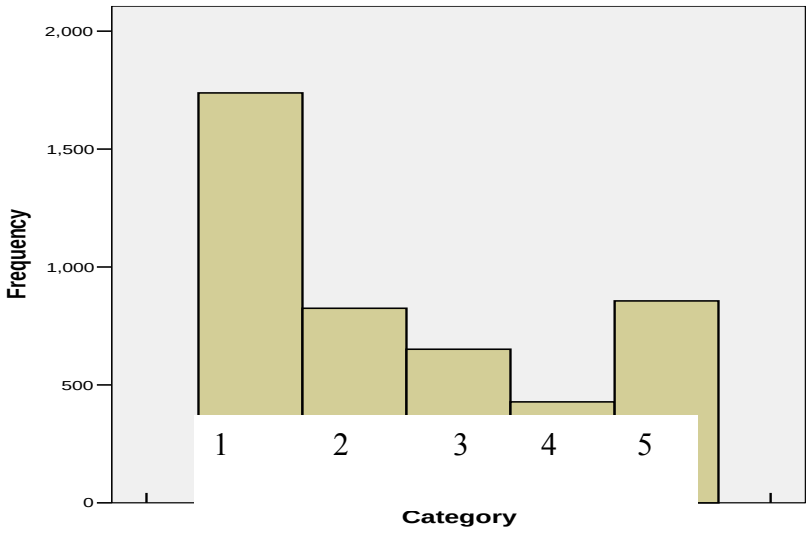

Figure 12. Histogram of the mapped genes categorized by the differences in the start position obtained from the two sources (Ensembl and UCSC knowngene).
Category 1 corresponds to differences less than 5 bps, Category 2 consists of genes with differences between 5 and less than 100 bps, Category 3 – between 100 and 1000 bps. Category 4 – greater than 1000 bps and less than 5000 bps. Category 5 – greater than 5000 bps.

To assess the effect of the discrepancies on experimental data, the NeuroD GST library generated by the SACO process (described previously) was used. NeuroD is a transcription factor that controls molecules that are involved in cell survival and differentiation in multiple tissues. Also known as BETA2, it is involved in a number of diseases, including diabetes, ataxia and deafness [27]. Because the discrepancies are highly correlated with the confidence both in the location and characterization of the tags, we examined the number of tags that could be located within genes and the number that could be located 1000 bps before the start of a gene.

Based on our definition of consequential discrepancies, a subset of these discrepancies has the potential to change the state of a GST – from being a potential TF binding site located near a gene to one that is not near a gene start. The threshold for an annotation impacting a GST depends on two factors. The first is the difference in the start position (delta) between the two annotations and the second is the distance of the GST to the nearest annotation start (start pos).

For this simple analysis, we can define a threshold value such that the sum should be less than 1000, that is

$$\text{delta} + \text{start pos} \leq 1000 \text{ bps}$$

Any discrepancy that fails to satisfy this condition can potentially impact the status of a GST.

Using Ensembl as the reference source, we found that of a total of 26124 experimental tags, 10158 unique GSTs (38.8%) were located within the gene boundaries. 1880 (7.2%) were identified as being located within 1000 bps of the start of a gene. Using UCSC's known genes as the reference, we found 9800 GST tags (37.5%) were located within a gene. 376(1%) GSTs were identified as being located 1000 bps before a gene. One reason for the higher placement of GSTs with Ensembl (38.8% v/s 37.5% for UCSC's known genes) is because all the transcripts from a gene are included in the data set. In the case of UCSC known genes, only one transcript per gene is represented. We found that a total of 1304 GSTs were impacted by consequential discrepancies that we identified using the definition above. As mentioned previously, this has the impact to change the sequences that we use for downstream analyses such as motif finding.

5.3 Application of HIDE to other Biological Use Cases

The identification of discrepancies can be critical for a number of different biological (and non-biological scenarios) and can have serious implications for any downstream or subsequent analysis of the integrated data. Here we briefly consider another example of the utility of HIDE, the development of a *SNP panel* (a chip-based technique used for the simultaneous genotyping of a number of SNPs) for a disease association study. SNPs or Single nucleotide polymorphisms, being a common variant (with the least frequent allele having an abundance of 1% or greater), can help localize regions of the genome that are important in disease, acting as a surrogate marker for the disease locus [27]. A SNP is classified as *coding* if it is located within the coding portion of a gene. A *non-coding SNP* is one that does not lie in the coding portion of a gene. A coding SNP may affect the biological function of the protein that a gene codes for if there is a change at the amino acid level. In these cases, the SNP itself may be a causal variant,

rather than just a marker associated with a *causal variant* (the variant that is responsible for the functional change). Many researchers will design SNP panels to therefore maximize the number of coding SNPs. This panel will be used to design a customized genotyping array for analyzing SNPs of interest in a case-control or population study. A SNP not included in the array cannot be included in the downstream analysis and hence it is important to ensure that the most accurate and complete information is available during construction of the panel. The panel is to be used as an analysis tool in population-based studies. Not including a SNP in the panel will impact the effect of the SNP at the population level.

The classification of a SNP as either coding or non-coding is dependent upon the annotations for gene starts and stops. Hence data sources that provide information on gene locations are of interest. Any discrepancies between these sources may affect the classification of SNPs. The problem is now reduced into one of discrepancy detection, which can now be handled by HIDE. Additionally, the SNP study highlighted here is carried out in the human genome, as opposed to the mouse genome for SACO, indicating the generalizability of HIDE at the genomic level.

5.3.1 Background for the SNP Use Case

With the advances in new technologies, researchers can design custom SNP arrays to interrogate specific regions or genes of the genome. In our case, the SNP study was restricted to a specific region of the human genome implicated in Alzheimer's disease via an association study. The investigators at the OHSU Layton Aging and Alzheimer's Disease Center were interested in the genetic variation in the gene MCPH1 (located on chromosome 8). For genotyping the coding SNPs in the study population, it was important to identify the set of coding SNPs first.

5.3.2 Detecting the Impact of discrepancies on SNP categories

Two data sources, UCSC's known genes and Ensembl were identified by the investigator as being the source of annotations. We utilized the two data sources identified by the investigator as the source for gene annotations. Locations for the individual SNPs were obtained from the dbSNP [14] database. The SNP repository, dbSNP, is a primary data source and contains information submitted by individual

researchers and consortiums. Because our study focused in looking for discrepancies in one gene and the SNPs that were affected, we looked at the data at the transcript level. As both sources used the common gene identifier (MCPH1), we chose to use this identifier to create the mapping between the two sources. There were multiple transcripts associated with the gene in each source. We compared the equality of the following aggregate scores to detect discrepancies. For the start position, we calculated the aggregate score for each source across the transcripts and tested for equality of these aggregate scores.

$StartScore_{source,\ gene} = \min_{i} (transcriptstart_{gene})$ where i is the number of transcripts

$StopScore_{source,\ gene} = \max_{i} (transcriptstop_{gene})$ where i is the number of transcripts

We found three transcripts annotated with the gene name in UCSC and two transcripts in Ensembl (Table 2). This suggests that there are discrepancies in the start and end of the gene across the sources as the UCSC gene start (defined as the minimum of starts of all transcripts) is located before the Ensembl start. The UCSC gene end (the maximum of all transcript ends) is located after the Ensembl gene end. On examining the number of SNPs affected, we found that discrepancies in the start position did not impact the classification of any known SNPs. However in the case of the end position, the classifications of 26 SNPs were affected by the discrepancies (Fig. 13).

Table 2. Details of the transcripts annotated with the gene identifier MCPH1 in the UCSC and Ensembl databases. Discrepancies are present in both the start and end positions of the genes. The gene start in a database is defined as the minimum of all transcript starts in the database. The gene end is defined as the maximum of all the transcript ends. This definition holds true because the gene is positioned on the + strand of the genome. The + or forward strand, is the DNA strand where the base pairs increase when moving from the 5´ to 3´.

Transcript ID	Source	Start	Stop
NM_024596	UCSC	6251529	6493434
Uc003wqh.1	UCSC	6251529	6291496
Uc003wqi.1	UCSC	6251529	6493434
OTTHUMG00000139041	Ensembl	6251486	6489391
ENSG00000147316	Ensembl	6251530	6488550

5.3.3 Results

Discrepancies in the end positions can be seen in the UCSC browser window (Fig. 9) and are overlaid with the set of SNPs affected. The line representing the discrepancy track is at the top of the figure and is annotated with the source of the data (the annotations are currently not a feature of HIDE). The SNP track, from dbSNP, is overlaid on the discrepancy track with all the details of the identifiers and their positions visible within the genome browser. A total of 26 SNPs were now classified as coding SNPs based on the data from the two sources. These SNPs would not have been included without the HIDE analysis and discrepancy detection. Given that there are only a total of 96 SNPs that can be included as part of the array, this subset of SNPs that were missed form a significant portion (27%), if included in the analysis.

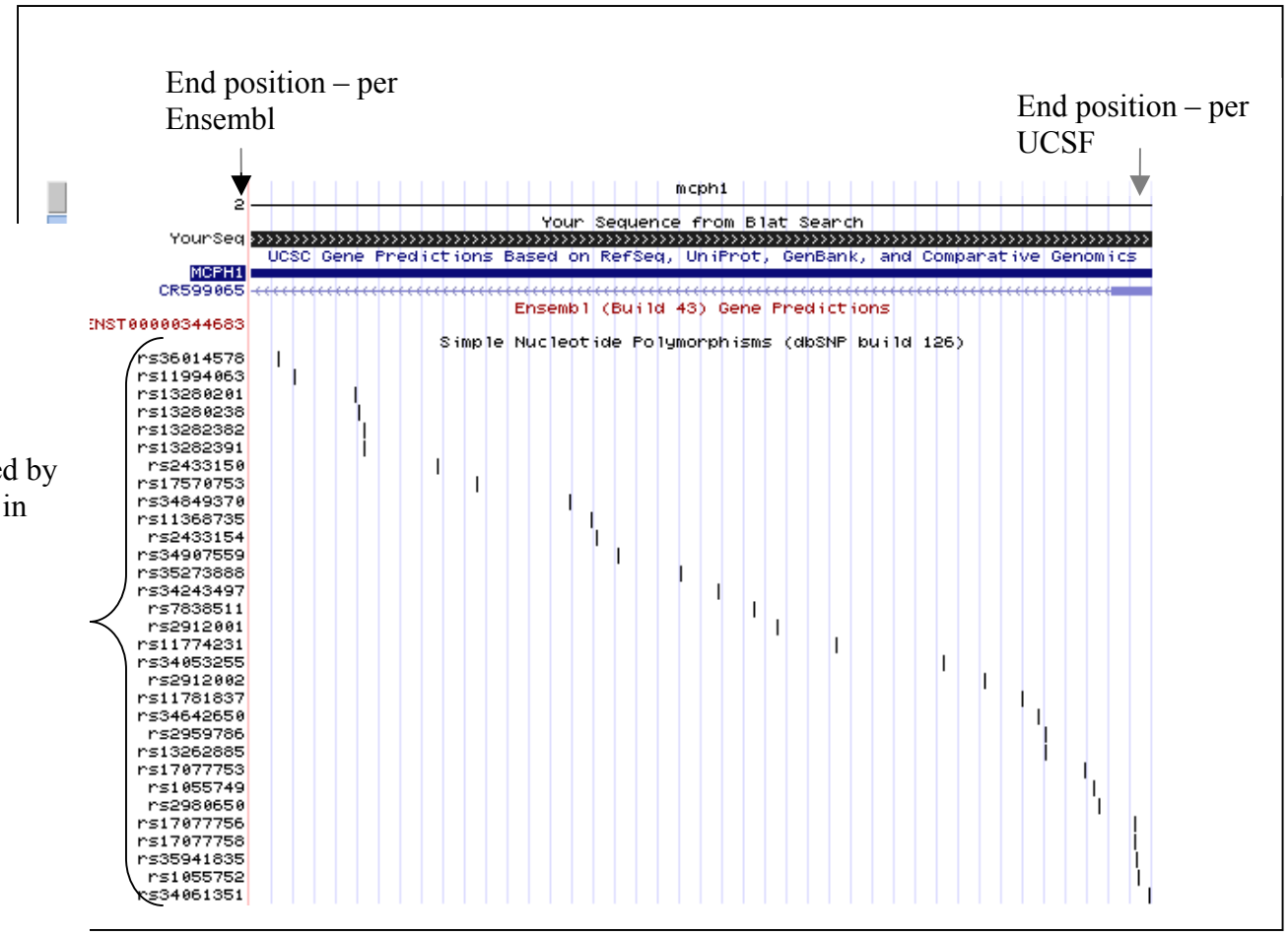

Figure 13. Impact of discrepancies visualized in the UCSC genome browser.
The line at the top indicates the discrepancies in the end positions in the two data bases – UCSC and Ensembl. The list of 26 SNPs and their locations, derived from dbSNP, are overlaid on the discrepancy track.

Chapter 6 Conclusions

6.1 Issues addressed in this study

In this thesis, we propose a systematic workflow to detect errors that arise during integration of data using a combination of lineage information and independence of data sources. The workflow we have proposed currently consists of a set of definitions and is modeled as a flowchart to help the user determine the queries that are necessary for data cleaning, given the data and the relationships that exist between the data. Additionally, we prototyped computer tools to help with parts of the task. We also propose how such a mapping can be extended and enumerate a number of challenges which will need to be addressed before achieving complete automation of such a framework.

Traditionally, the approach for data cleaning has been using data cleaning tools and carried out in a batch processing model [1]. Additionally, the incorrect information is "transformed" to remove the error. This approach presents a major drawback for the biologist who may wish to know the actual error or discrepancy and is not interested in transformation of the data to enforce conformance between the sources. Additionally, a large percentage of the data a biologist deals with are drawn from legacy sources and it may not be possible to incorporate changes back into the sources. For the reasons above, it becomes important to have a mechanism of representing the errors to the biologist and in a manner that does not need to modify the underlying resources. One solution to handle such legacy data, proposed by Lincoln Stein, was to allow annotations to be housed in multiple servers in a distributed manner (DAS). The data would be integrated on a need-to basis by the client side machine (read the user).

The errors that are detected between sources will be determined by the actual sources that are included in the integration process. It is highly probable addition of new

sources will result in new errors being discovered. To avoid re-mapping all the data, providing a generalized, superimposed framework [30] that models or captures the relationships between the sources will allow the biologist to include new resources in the analysis. A generalized framework that directs testing will also allow the biologist to fashion queries to detect errors that will impact their analyses. The concept of the superimposed framework has been proposed by Delcambre and Maier [30].

6.2 Extensions to the framework

The current *implementation* of the workflow is limited in its ability to serve as a general purpose framework. To extend its utility, we propose a number of extensions to the current implementation.

6.2.1. Use of a mediator architecture for data integration

In the workflow we have developed, we have identified factors – independence of data sources, status of a source (primary versus not), detecting invisible inputs, matching entities, etc. - as some of the challenges we faced in integrating the data.

The scenario that we dealt with was a two-way mapping (mapping between two data sources for each entity). Although the challenges we identified are relevant for a multi-way mapping, it is not feasible to have 2-way maps for each pair of resources. In addition to becoming infeasible computationally for a large number of sources, it fails to provide a consolidated picture of the data. To generalize this notion of mapping, we propose the use of an ontology to serve as a global schema [27-29]. Individual data elements from the different sources are then mapped into the concepts of the ontology. This ontology is a superimposed layer [30] that is visualized and provides the user with summary statistics of discrepancies that exist between the sources. This visualization layer also serves as the starting point for the user to drill down to the individual sources of data. Additionally, a confidence measure for each entity in the ontology is calculated by weighting each source [31]. Initially, equal weights would be assigned to sources.

The advantage of using the approach outlined above is that it provides the biologists with a mechanism of visualizing the errors in a context that is important to them, that of biological entities. A second advantage is from the integration perspective.

By mapping to the concepts into the ontology, it becomes easier to match entities in the database. However, there are challenges to this approach [19], which indicate that different sets of criteria will have to be addressed when integrating independent sources.

Finally, by mapping the data into the ontology, it will be possible for the user to visualize the data coverage provided by the sources and provide some direction about entities that have insufficient or no coverage in the analysis. Such a visualization of the coverage may provide the user in identifying new sources that need to be brought into the ontology. For example in the case of SACO, suppose that genomic features are part of the ontology. So in addition to a gene, there are entities such as mRNAs, ncRNAs, miRNAs, etc. Each of these precursor entities are related to one another by their position of their precursors on the sequence. One source (RNAdb) used in the analysis provides extensive annotation for groups of ncRNAs, identified by a representative sequence and position. The information about each group's members is hidden. The use of representatives for a set is insufficient for the SACO process, since the location of every precursor in the group is important. For this reason, the user may choose to include the primary source Riken in the analysis pipeline to obtain information about all the members in the group. The use of an ontology will also help in alleviating the limitation of this approach – invalid expectations. By mapping the data into the ontology, the identification of equivalent entities in the different databases becomes standardized and simpler, although a different set of challenges will need to be addressed to have a seamless integration of the entities. An implementation of this concept is presented in Biowarehouse[32], which serves as an integrated repository for a number of biological data sources. The mapping techniques presented in this thesis can be used to map the individual data sources into the Biowarehouse schema. Additionally, by specifying the mapping rules at the level of the ontology, the process is decoupled from the data sources. New data sources can be added easily and can utilize the mapping information already available. By mapping data into an ontology, issues arising due to differences in naming conventions across data sources and the challenge of object identification are expected to be alleviated.

6.2.2 Learning from data

In the current implementation of the workflow, we use the user's input for both the meta-data and creating mappings between sources. One possible extension is to learn from the data and use the information obtained to both create tables and to flag deviations from the learned information. This approach of learning from the data and flagging deviations has been implemented in a data integration tool, Potter's wheel [10]. Tools such as Potter's wheel can be implemented easily as a part of the workflow. The advantage of Potter's wheel is that it eliminates the need for the user's input of data types. The program is capable of learning the information by sampling the data.

6.2.3 Classifying Discrepancies as Errors

A certain level of minimum information is required to classify a discrepancy as an error. The first is related to the dependence between sources. If two sources are dependent but present different information about a common entity, the discrepancy can be viewed as an error. Further information such as time the data was generated, expert inputs, etc. will be required to localize the error to one data source. A second piece of information is tracing the lineage of the data. If the data in both sources can be traced back to the original source, it may be possible to determine if a discrepancy is truly an error. Currently, the workflow does not attempt to classify a discrepancy as an error. An extension would be to utilize more extensive meta-data to detect errors in data sources. Consider the previously introduced example of a business listing in the yellow pages and the company website. If there is a discrepancy between the two sources related to hours of operation, it may be necessary to obtain more information to decide which source is incorrect. Without this extra piece of information, it is only possible to identify the discrepancy between the twos sources, not the incorrect source.

References

1. Rahm E, Do H-H: **Data Cleaning: Problems and Current Approaches** *IEEE Bulletin of the Technical Committee on Data Engineering* 2000, **23**(4).
2. Levy AY: **Logic-Based Techniques in Data Integration**. In: *Logic Based Artificial Intelligence.* Edited by Minker J: Kluwer Publishers; 2000.
3. Buneman P, Khanna S, Tan W-C: **Data Provenance: Some Basic Issues**. *Lecture Notes In Computer Science* 2000, **1974**:87-93.
4. Buneman P, Khanna S, Tan W-C: **Why and Where: A Characterization of Data Provenance**. In: *International Conference on Database theory: 2001*; 2001: 316-333.
5. Brenner S: **Errors in Genome Annotation**. *Trends in Genetics* 1999, **15**(4):132-133.
6. Bork P: **Powers and Pitfalls in Sequence Analysis: The 70% hurdle**. *Genome Research* 2000, **10**:398-400.
7. Devos D, Valencia A: **Intrinsic Errors in Genome Annotation**. *Trends in Genetics* 2001, **17**(8):429-431.
8. Calvanese D, De Giacomo G, Lenzerini M, Nardi D, Rosant R: **A Principled Approach to Data Integration and Reconciliation in Data Warehousing**. In: *Design and management of Data Warehouses: 1999; Heidelberg, Germany;1999*: 16/1-16/11.
9. Galhardas H, Florescu D, Shasha D, Simon E: **Declarative Data Cleaning: Language, Model and Algorithms**. In: *VLDB: 2001; Rome, Italy;2001*: 371-380.
10. Raman V, Hellerstein JM: **Potter's Wheel: An Interactive Framework for Data Transformation**. In: *VLDB: 2001; Rome, Italy;2001*:381-390.
11. Muller H, Naumann F, Freytag JC: **Data Quality in Genome Databases**. In: *International Conference on Information Quality: 2003; Cambridge, MA*; 2003: 269-284.
12. Ouzanis C, Karp P: **The Past, Present and Future of Genome-wide re-annotation**. *Genome Biology* 2002, **3**(2):2001.2001-2001.2006.
13. von Mering, C, Huynen, M, Jaeggi, D, Schmidt, S, Bork, P and B. Snel. STRING: **A Database of predicted Functional Associations between Proteins.** *Nucl Acids Res* 2003. **31**:258-261.
14. Sherry S, Ward M, Sirotkin K: **Use of Molecular Variation in the NCBI dbSNP Database.** *Human Mutation* 2000, **15**(1):68-75.
15. Risch N, Merikangas K: **The Future of Genetic Studies of Complex Human Diseases.** *Science* (1996). **273**:1516−1517.

16. Impey S, McCorkle S, Cha-Molstad H, Dwyer J, Yochum G, Boss J, McWeeney S, Dunn J, Mandel G, Goodman R: **Defining the CREB Regulon: A Genome-wide Analysis of Transcription Factor Regulatory Regions**. *Cell* 2004, **119**(7):1041-1054.
17. Karolchik D, Hinrichs AS, Furey TS, Roskin KM, Sugnet CW, Haussler D, Kent WJ: **The UCSC Table Browser Data Retrieval Tool**. *Nucl Acids Res* 2004, **32**(suppl_1):D493-496.
18. Pang KC, Stephen S, Engstrom PG, Tajul-Arifin K, Chen W, Wahlestedt C, Lenhard B, Hayashizaki Y, Mattick JS: **RNAdb- A Comprehensive Mammalian noncoding RNA Database**. *Nucl Acids Res* 2005, **33**(suppl_1):D125-130.
19. Kim P, Kim N, Lee Y, Kim B, Shin Y, Lee S: **ECgene: Genome Annotation for Alternative Splicing**. *Nucl Acids Res* 2005, **33**(suppl_1):D75-79.
20. Hubbard T, Andrews D, Caccamo M, Cameron G, Chen Y, Clamp M, Clarke L, Coates G, Cox T, Cunningham F *et al*: **Ensembl 2005**. *Nucl Acids Res* 2005, **33**(suppl_1):D447-453.
21. Bejerano G, Pheasant M, Makunin I, Stephen S, Kent WJ, Mattick JS, D. H: **Ultraconserved Elements in the Human Genome.** *Science* 2004, **304**(5675):1321-1325.
22. Bancilhon F, Spyratos. N: **Independent Components of Databases**. In: *7th VLDB: 1981; Cannes, France;1981*:398-408.
23. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL: **GenBank**. *Nucl Acids Res* 2005, **33**(suppl_1):D34-38.
24. Griffiths-Jones S: **The microRNA Registry**. *Nucl Acids Res* 2004, **32**(suppl_1):D109-111.
25. Kanz C, Aldebert P, Althorpe N, Baker W, Baldwin A, Bates K, Browne P, van den Broek A, Castro M, Cochrane G *et al*: **The EMBL Nucleotide Sequence Database**. *Nucl Acids Res* 2005, **33**(suppl_1):D29-33.
26. Maglott D, Ostell J, Pruitt KD, Tatusova T: **Entrez Gene: Gene-Centered Information at NCBI**. *Nucl Acids Res* 2005, **33**(suppl_1):D54-58.
27. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT *et al*: **Gene Ontology: Tool for the Unification of Biology**. *Nat Genet* 2000, **25**(1):25-29.
28. Hakimpour F, Geppert A: **Global Schema Generation Using Formal Ontologies**. In: *Conceptual Modeling - ER 2002*; 2002.
29. Lenzerini M: **Data integration: a theoretical perspective** In: *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems 2002; Madison, Wisconsin* ACM Press; 2002: 233-246
30. Delcambre LML, Maier D: **Models for Superimposed Information** In: *Proceedings of the Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling 1999*: Springer-Verlag; 1999: 264-280
31. Wu X, Zhang S: **Synthesizing High-Frequency Rules from Different Data Sources**. *IEEE Transactions on Knowledge and Data Engineering* 2003, **15**(2):352-367.

32. Lee, T J, Poulio, Y, Wagner, V, Gupta, P, Stringer-Calvert, D W J, Tanenbaum, J and P. Karp. **BioWarehouse: a bioinformatics database warehouse toolkit**. *Bioinformatics* 2006. 7:**170**.

Appendix A List of possible Queries

Name of Query	Position in flowchart	Description of Query	Notes
Test for Equality	Q3	Checks that an entity has the same value for a property	Need a minimum of two attributes that are common between sources – the entity identifier and the property. *Caveat: make sure entities compared are the same. E.g. Accession nos. in Fantom2 were not the same as accession numbers in UCSC*
Test for Membership/ Test for Location	Answer to Q5 identifies membership/sharing as property that exists between attributes. *E.g. Riken chooses one representative location for a sequence (curated) which is selected from a list of possible locations in UCSC.*	Pull out the set of possible values. Check if the particular value is present in the set or check if locations overlap.	Expect the parent source to contain the entire set.
Test for Complement	Answer to Q5 identifies relationship as complement. *E.g. non coding sequences cannot be in same location as coding sequences.*	Pull out possible values that are not allowed. Check if the particular value is present	Possible values can include exact values, range restrictions, etc.
Test for proximity/ context	Answer to Q5 identifies proximity of entities	Check if the location of an entity lies within a window (specified by user) of the other entity	Depending on the type of entity and the property being compared, this query can cover both temporal and spatial proximity and will be at different granularities – same compartment of cell, same chromosome, same cell cycle phase etc. Can be thought of as variant of the test for membership where it is necessary to check if both entities belong to the same set. Similar to the test for dependence at data source level.
Test for precursor of / Codes for	Answer to Q5 identifies one entity is derived from another	Check if the first entity can be translated to the second entity exactly	May be time consuming and intensive. There may be other proxy queries that can be used instead. (Issue of balancing the no. and cost of querying as opposed to complete coverage).

Glossary

3' UTR (untranslated region) is the portion of an mRNA from the 3' end of the mRNA to the position of the last codon used in translation.

5' UTR is the portion of an mRNA from the 5' end to the position of the first codon used in translation.

Association Study is a commonly used genetic tool that tests for the co-occurrence of a genetic trait and disease phenotype.

ChIP or **chromatin immunoprecipitation** refers to the process by which proteins are bound to DNA, the bound sequences subsequently isolated and sequenced. The aim of the process is to identify, characterize and localize the protein binding sites.

Clone A section of DNA that has been inserted into a vector molecule, such as a plasmid or a phage chromosome, and then replicated to form many identical copies.

CpG island They are regions with a high percentage of cytosine and guanine bases relative to the local background. The "islands" range from a few hundred to a few thousand bases in length.

EST An expressed sequence tag (EST) is a small part of the coding portion of a gene. It is often used to localize the gene on the genome.

Gene is a basic unit of inheritance.

mRNA are messenger RNA molecules that are translated into protein.

microRNAs (**miRNA**) are single-stranded RNA molecules of about 21-23 nucleotides in length thought to regulate the expression of other genes.

ncRNA is RNA that is not translated to protein product. It is believed to have a role in translational regulation.

Restriction Enzymes are special proteins that cut DNA strands at or near specialized motifs.

rRNA A class of RNA molecules that are a component of the protein complex involved in translation.

SAGE (Serial Analysis of Gene Expression) is a technique that allows rapid, detailed analysis of thousands of transcripts in a cell.
siRNA A class of RNA molecules that regulate gene expression by binding to and preventing the translation of MRNA to protein [25].

SNP (**Single nucleotide polymorphism)** is a change in a single base of DNA, compared to the expected base at that location [26].

TF (Transcription factor) A TF is a specialized protein complex that binds to the DNA sequence at a specific location and recruits other proteins that help form the RNA from the corresponding DNA sequence.

Ultraconserved elements are nucleotide sequences at least 200 bases that are conserved between orthologous regions of human, rat and mouse genomes [19].

Biographical Sketch

Ranjani Ramakrishnan was born in 1974 in India. She completed her undergraduate education at the Birla Institute of Technology and Science, India where she received a dual degrees in Biology and Electrical and Electronics Engineering. She subsequently received an M.S degree in Genetics from the University of North Carolina, Chapel Hill. She is currently interested in applying Graphical Models to learn biological networks and regulation from high throughput data. She received the Markey Fellowship as a graduate student at Chapel Hill. During her Masters program at OGI she received travel awards from the Government of Japan and AIST and the NSF to attend the first International BioPAX conference and the Summer School on Biocomplexity, respectively.