DATA QUALITY IN GENOME DATABASES

(Research Paper)

Heiko Müller
Humboldt University Berlin, Germany
hmueller@informatik.hu-berlin.de

Felix Naumann
Humboldt University Berlin, Germany
naumann@informatik.hu-berlin.de

Johann-Christoph Freytag
Humboldt University Berlin, Germany
freytag@dbis.informatik.hu-berlin.de

Abstract: Genome databases store data about molecular biological entities such as genes, proteins, diseases, etc. The main purpose of creating and maintaining such databases in commercial organizations is their importance in the process of drug discovery. Genome data is analyzed and interpreted to gain so-called *leads*, i.e., promising structures for new drugs. Following a lead through the process of drug development, testing, and finally several stages of clinical trials is extremely expensive. Thus, an underlying high quality database is of utmost importance. Due to the exploratory nature of genome databases, commercial and public, they are inaccurate, incomplete, outdated and in an overall poor state.
This paper highlights the important challenges of determining and improving data quality for databases storing molecular biological data. We examine the production process for genome data in detail and show that producing incorrect data is intrinsic to the process at the same time highlight common types of data errors. We compare these error classes with existing solutions for data cleansing and come to the conclusion that traditional and proven data cleansing techniques of other application domains do not suffice for the particular needs and problem types of genomic databases.

Key Words: Data Cleansing, Molecular Biology, Data Errors

1 GENOME DATA IS DIRTY

Increasing interest in genome data has lead to the availability of a multitude of publicly available genome databases today[1]. Genome data includes the actual sequences of the bio-molecules, i.e., DNA, RNA, and protein, as they were observed in wet lab experiments, e.g., …`actgctgaatc`… for DNA data, and experimental data, such as the environmental circumstances of the experiment, the organism the sample was taken from, the date, etc. In addition to the raw data, genomic databases store the structural and functional classification of sequences and sub-sequences, called annotation. The process of assigning meaning to sequence data by identifying regions of interest and of determining biological function for those regions is defined as genome annotation [17]. Genome annotation is performed by biologists and bioinformaticians in universities, in publicly funded institutions, and in corporations. The annotation data represents the most important part of genomic databases, namely the deeper biological meaning of the raw data. This annotation is ex-

[1] For a comprehensive listing see http://nar.oupjournals.org/cgi/content/full/31/1/1/DC1

pensive and difficult to obtain directly through experiments. Instead very often, this biological function is determined by comparing the sequence with those of various other information sources. A typical annotation process for, say, a certain human gene gathers existing functional annotation for a similar mouse gene. This process gives rise to two of the most daunting problems within this process: the integrated access of multiple sources and the quality of the retrieved data. The former problem is common to all integrating databases and is regarded elsewhere: Sheth and Larson give a systematic overview of such systems [27]. A prominent example of an integrating genomic database is IBM's DiscoveryLink [9]. The second problem of poor quality data has so far been studied only marginally in the context of genome data, despite the importance of high data quality for ongoing genome research.

Data of poor quality in genomic databases have enormous economic and medical impact on their users/customers. For instance, errors in genome data can result in improper target selection for biological experiments or pharmaceutical research. To bring a handful of new drugs to the market, pharmaceutical companies spend up billions of dollars in research [10]. Of thousands of promising leads derived from experimental genomic data only a handful reach clinical trials and only a single drug becomes marketable. Obviously, it is of great importance to base these far-reaching decisions on high quality data.

Research in pharmacogenomics will enable pharmaceutical companies to produce drugs specifically designed against the genotype of individual patients. For such medications incorrect data (about the patient or about the drug compound and the genomic processes it affects) can lead to serious consequences regarding the health of the so-treated patient.

On a smaller scale, missing, incomplete, or erroneous information hinders the automatic processing and analysis of data, experiments based on poor quality data yield incorrect results, etc. Such annoyances lead to a loss in confidence in the underlying data source or the provider of the data, and to a rise in effort and frustration for the biologist on a day-to-day basis. Through careful analysis of the experimental and annotation pipeline of genome data, we identify five classes for poor data quality: experimental errors, analysis errors, transformation errors, propagated errors, and stale data.

To tackle the problem of data errors of the five kinds mentioned earlier, a first step is to identify the producers of these errors. We have developed detailed *Information Product Maps* (IP-Maps, [26]) for the genome data production. In general there are four classes of data- (and thus error-) producers: Wet-lab experiments, semi-automated experiments, computational transformations, and computational analysis. Our analysis pinpoints the employment of each of these producers in the data production pipeline and the types of error they produce, thus providing a sound basis for quality improvement efforts.

1.1 Related work

While there has been much research in developing a general data cleansing framework [7,16,18,23], and while many data cleansing methods and applications have been developed for certain domains, such as address data [11,30] and health-care data [15,22], and many other domains, there is yet little research addressing the particular, and novel data cleansing problems as they occur in the life sciences domain. See [20] for a detailed classification and comparison of state-of-the-art data cleansing methods.

Apart from anecdotal evidence and our own experience working on some of the major life sciences databases, there are studies that show the existence of errors in genome databases. In [8] the

accuracy of several computer programs for the prediction of the structure of protein coding genes (structural annotation as explained later) is investigated. None of the programs reaches an accuracy of 100% thus yielding errors in databases containing structural annotation. The maximum accuracy that can be achieved using currently available prediction programs is believed to be 90% for protein coding regions and 70% for gene structure. Several studies [3,5,13] show the existence of errors in functional annotation of proteins (explained later). In [3] the error rate is estimated to be over 8%. This analysis was performed by comparing analysis results of three independent research groups annotating the proteome of *Mycoplasma genitalium* and counting the number of discrepancies between them. With the increased dependency on automatic annotation methods – due to the high data volume – this rate of errors can be expected only to rise. In [5] the authors use an approach based on the observation that most of the functional annotations are justified by relatively weak sequence similarities and on the considerable number of discrepancies between functions annotated for similar sequences. By extrapolating the discrepancies detected at a certain level of similarity to the number of proteins, it is possible to estimate the number of discrepancies between actual and automatically annotated functions. The expected level of error varies from less than 5% to more than 40%, depending on the type of annotated function. Finally, in [13] the authors generate a highly reliable set of annotations by carefully using automatic methods and experimental evidence. They compare their results with existing annotations and with the results of solely automatically performed annotations. For the original annotations only 63% of functional assignments within both datasets are in total agreement, while for the solely automatic annotations the precision is estimated to be 74% for the most reliable set of predictions.

1.2 Structure of this paper

In Section 2 the basic concepts of the application domain are defined. A short introduction of the underlying biological entities is followed by a list of the types of data stored in genome databases. Section 3 highlights the production of genome data using an IP map with four different producers of data, each with different characteristics. After a general classification of error types in genome databases in Section 4, we focus on interesting parts of the overall production pipeline and identify, which classes of errors are produced at which stage and suggest domain-specific quality checks to reduce these errors. Finally, in Section 5 we show why existing data cleansing techniques fall short for the especially complex domain of genome data.

2 BASIC CONCEPTS FOR DESCRIBING GENOME DATA

Similar to the term “gene” itself, ”genome data” is a term without a clear marked-off scope or commonly accepted definition. The *genome* is the entirety of genetic information of an organism. Genetic information enables organisms to exist, i.e., to transform energy from the environment, to move, to reproduce, to self-assemble (grow), and to repair themselves. Genome information is stored in the sequence of the four different building blocks, called *bases* (*adenine*, *guanine*, *cytosine*, and *thymine*), of the molecule *desoxy ribonucleic acid* (DNA). The DNA – a double stranded molecule forming the well-known double helix – is divided into transcribed and non-transcribed parts. The former are called *genes* and are the parts of main interest in biological, medical, and pharmaceutical research. *Transcription* is the first step of genome information processing. The resulting molecule, *ribonucleic acid* (RNA), is the single strand copy of a gene. It is used as a template for protein synthesis. The synthesis process, called *translation*, uses an organism-specific translation table (*genetic code*) to translate successive segments of length 3 (*codon*) each into one amino acid, the building blocks of the resulting protein. The translation always starts at a *start-codon* `atg` and ends at the first *stop-codon*, i.e., `taa`, `tag`, `tga`. The codon-structure defined by the start- and stop-codons is called the *reading frame*. There are three differ-

ent reading frames for each of the DNA strands. Proteins are the building blocks of living organisms performing a multitude of different functions. This process of biological information processing within an organism's cells is called the "central dogma of molecular biology" as described by Francis Crick in 1957, and is shown in Figure 1.

In principle, every piece of information about the genome and genome products of living organisms can be termed genome data. By genome data we mean information about the bio-molecules DNA, RNA, and protein, such as their sequence (composition of bases or amino acids), their structural features, and their function performed within the organism. Here, we disregard data from gene expression studies, information about protein interactions during complex biological functions as well as the 3D-structure of molecules. The main data for this study are:

- strings representing the sequences of bio-molecules,
- attributes, describing certain properties using values from a fixed set of domains, and
- annotations, i.e., functional or structural classification of for regions or collections of regions of the genome or proteins.

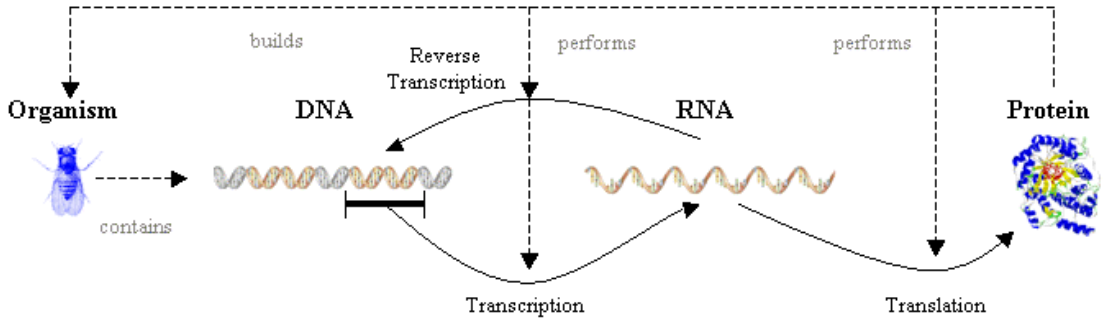

Figure 1: Central Dogma of Molecular Biology

The data is further classified into semantic classes of genome data resulting from the following general process:

- **Genome sequence data** represents the sequence of DNA molecules extracted from the cells of different organisms. They are represented by strings over the four-letter alphabet {A, C, G, T}. Each string represents either sub-parts of the complete genome or concatenated larger parts of the genome. Genome sequence data is mainly the result of sequencing projects. The most popular sequencing project is the Human Genome Project (HGP) by a consortium of research institutions, aiming at generating the human genome sequence, which has been completed in 2003. Commercial projects like the ones performed by Celera are in competition to the HPG.
- **EST sequence data** are also strings over a four letter alphabet representing transcribed parts (RNA) of the genome, called *expressed sequence tags* (ESTs).
- **Structural annotation** describes known features that are identified and shown on the genome sequence data. The features of interest are for example the occurrence of sequence patterns, *single nucleotide polymorphisms* (SNPs), e.g., proven sequence variation between individuals, and gene location and gene structure, which is of special interest for the pharmaceutical and further biological research.

- **Protein sequence data** represents the sequence of amino acid of proteins by a string over the alphabet of twenty amino acids.
- **Functional annotation** describes in non-standardized textual form the function performed by a certain protein within the organism, as well as its participation (or that of its mutations) in the development of a certain disease. Biologists enter free text descriptions at will, in different languages, using different abbreviations, etc.
- **Protein motifs** represent the conserved characteristic features of a protein family, i.e., groups of related proteins within different organisms in various forms. Often, only small parts of the protein are responsible for a certain function, and within this part several combinations of amino acids are allowed.

A characteristic genome data item is shown in Figure 2 It is a cutout of a data entry in the EMBL DNA Sequence Database [28] highlighting the different classes of information and the data sources.

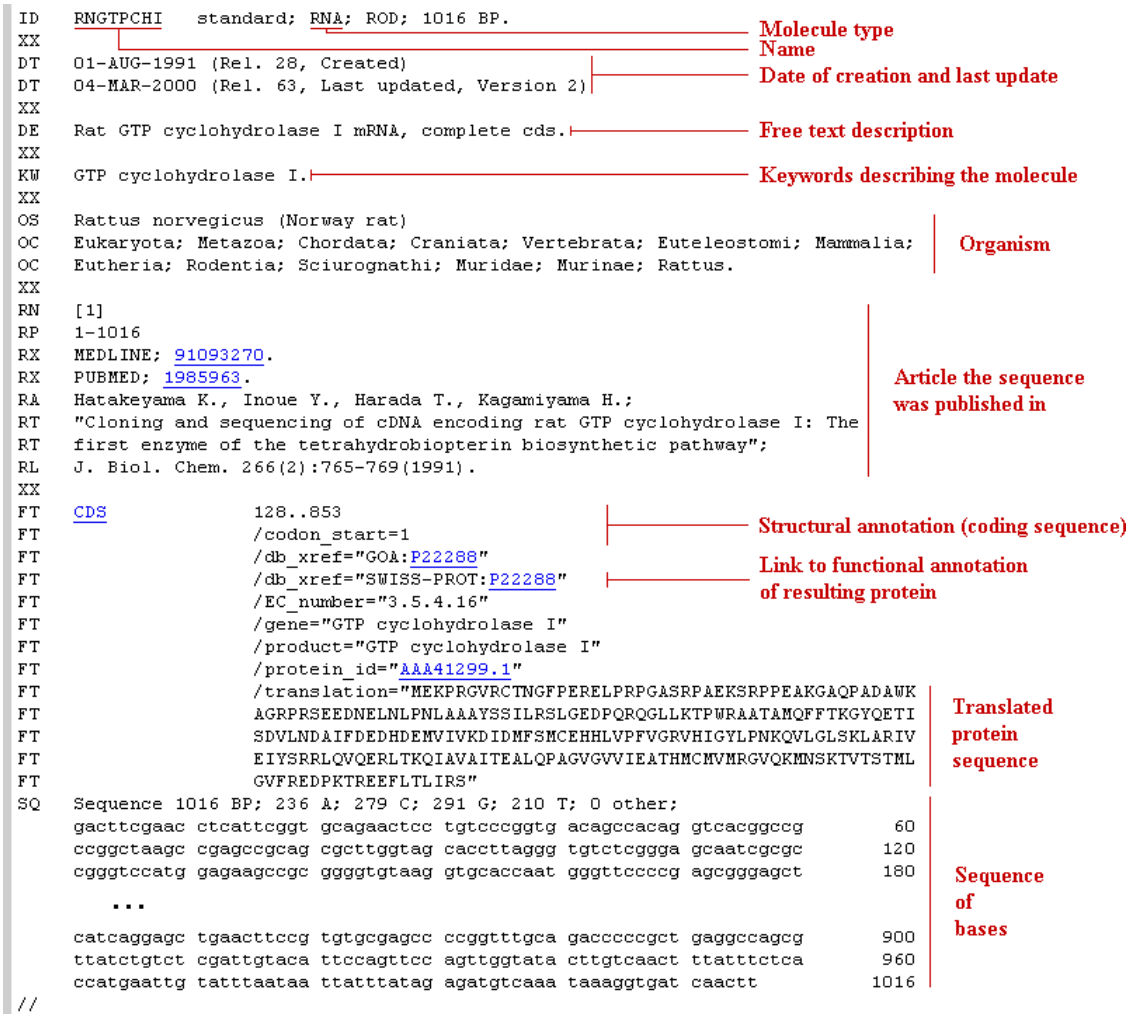

Figure 2: Exemplary genome data entry from the EMBL DNA Sequence Database

3 GENOME DATA PRODUCTION

Genome data production is performed by people with different skills and domain knowledge. The process involves

- Biologists working in the wet-lab,
- Lab assistants who install and operate machines and robots,
- and Bioinformaticians, i.e., computer users having biological expert knowledge.

Production of genome data is done in collaboration by different workgroups and different institutions from around the world, using their own, often proprietary, techniques, methods, and protocols. This setup alone implies the poor data quality of the end product, as we will argue later. The main data-producing techniques for genome data are:

- **Wet-lab experiments** (performed by biologists, shown in the IP maps as ): Within the wet-lab biologists perform experiments on the living organism and explore interesting features, such as organism behavior after manipulation under given conditions. The results are interpreted and transformed into information stored in digital format within usually proprietary data storage system.
- **Semi-automated experiments** (performed by lab assistants, shown as ): Automata and robots support biologists in performing those experiments that are dreary and must be repeated often. This automation increases the throughput and lowers the error rate, because machines are able to perform without fatigue. For example, sequence determination is a very error prone process when performed by humans. Biologists still have to perform the experimental set-up, but the automata generate the information directly in digital format.
- **Computational transformation** (performed by bioinformaticians or lab assistants, shown as ): Transforms data from one representation into another that is better manageable and interpretable by humans or machines. This involves translation of sequence information (e.g., base calling) or concatenation of strings (e.g., sequence assembly). Computational transformations do not require additional knowledge-based interpretation of the results.
- **Computational analysis** (performed by biologists bioinformaticians, shown as ): The results of experiments are interpreted by human experts using computer software to produce new information. This is termed data analysis, and plays an important role in genome data production. The role is so important, because of the huge amount of data produced, mainly by semi-automated experiments, that is to be analyzed. Most of this work is sequence analysis, i.e., giving meaning to sequence data produced by sequencing labs. Here, digital information is interpreted to generate new digital information.

Often, a genome data product can be derived alternatively by wet-lab experiments or computational analysis. There is a time/quality trade-off involved, as experiments are more accurate than computational analysis, while also being more expensive and time consuming.

From this description it has already become clear that genome data production is an interdependent process. The information gained in one step is used and further analyzed in the following step, generating new knowledge and information. The information gained is eventually re-used as input in further data generation and analysis. The overall process is shown in Figure 3. Genome data is produced in four (mostly) dependent steps:

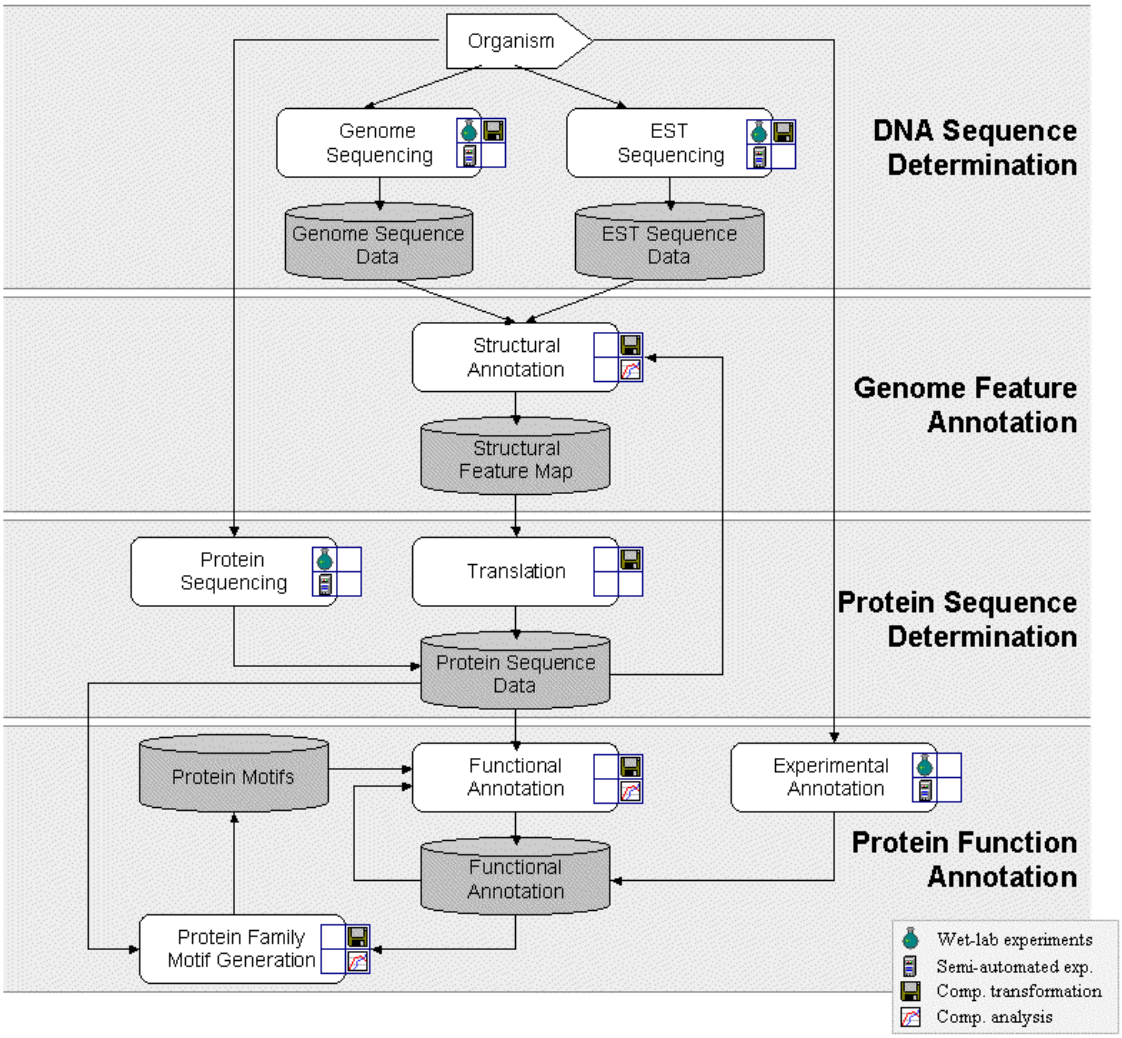

Figure 3: General Genome Data Production Process

Step 1 **DNA sequence determination**: Starting from the living organism, the sequences of the DNA (genome sequence data) and of the transcribed genome regions (EST sequence data) are generated. This is performed in a combination of wet-lab experiments, and semi-automated experiments, and also includes computational transformation. The results are strings representing DNA sequences and attributes describing the sequence properties, such as the organism it was taken from.

Step 2 **Genome feature annotation**: Using the information produced in DNA sequence determination, relevant biological regions and structural features are identified on the genome sequence data, e.g., the localization and structure of genes. The data is mostly generated by computational analysis or it reflects the results of other experiments mapped onto the genome sequence using computer programs. The result are structural classifications of sequence regions.

Step 3 **Protein sequence determination**: The sequence of proteins within an organism is determined either experimentally using extracted proteins from the living organism or through computational transformation using the information from the first two subprocesses. In the first case, the production mainly is performed experimentally, because

semi-automation is only marginal within this process. In the second case this is performed by simple translation of the genome sequence of the identified genes. In both cases, the result is a string representing the amino acid sequence of a protein.

Step 4 **Protein functional annotation**: Using the protein sequences resulting from protein sequence determination, the function of the protein performed within the organism is described. This is normally done by assigning the protein to different classes of biological function based on features of the amino acid sequence. Protein functional annotation can be performed either experimentally, which is time consuming, or manually, which is fast but error prone. The result is a set of functional classifications for each protein.

In recent years, a multitude of tools and protocols for the production of genome data have been developed. The usage and combination of these tools and protocols within the genome data production process varies among institutions and workgroups and also changes over time. In most cases these changes remain undocumented for the outside world - making it hard to reconstruct the production process. A standard procedure for genome data production does not exist. Therefore, we gave only a general overview of the basic steps involved. The occurring errors, which influence the quality of the resulting data in the different sub-processes, are described in the following.

4 ERRORS IN GENOME DATA PRODUCTION

From the description of the genome data production process we can define several classes of errors within genome data:

- **experimental errors** due to unnoticed experimental setup failure or systematical errors,
- **analysis errors** due to misinterpretation of information,
- **transformation errors** while performing transformations of information from one representation into another or one medium to another , e.g., data input,
- **propagated errors**, when erroneous data is used for the generation of new data, and
- **stale data**, i.e., unnoticed changes to base data on which a data item depends and that falsify it.

For the special case of errors in protein function annotation the TABS standard (Transitive Annotation-Based Scale, [21]), defines classes of errors as (listed in descending order of gravity for error propagation): False positive, over-prediction, domain error, false negative, under-prediction, undefined source, and typographical error. Their classification is oriented towards the actual data, while our classification stems from the analysis of the data production process.

4.1 DNA Sequence Determination

DNA sequence determination starts from individual organisms and comprises the two parts *genome data sequencing* and *EST data sequencing*. We ignore the second part for brevity.

In DNA sequence determination (Figure 4), after isolating the DNA molecules from the cells, they are split into overlapping parts of about 1,000 bases, and then the sequence is determined for each of the parts using sequencing automata and software programs (*base calling*). Afterwards, the resulting sequence strings are input into an assembly program, which produces a representative sequence of the entire genome as a textual string.

Errors

The main types of erroneous information are incorrect sequence data and property values. They are both caused either by experimental errors or by transformation errors.

Experimental errors: The data quality mainly depends on the sequence preparation step and the experimental setup, as well as on the base composition of the DNA to be sequenced. Especially DNA regions containing high amounts of bases G and C, e.g., …`gcgagtgcgacgttcg`…, are difficult to sequence, because of physical constraints. In regions of repeating bases, e.g., …`gatggtgaaaaaaaaa`…, there is the possibility of missing a base because of overlapping signals. Poor experimental practices or improper usages of chemicals cause sample contamination or preparation failure.

Transformation errors: In the beginnings of DNA sequencing base calling has been an error-prone step. This has been improved with the use of modern high-throughput sequencing automata, such as the ABI 3730xl DNA Analyzer from Applied Biosystems. In [25] the error rate in sequences for six different sequencing projects is estimated between 0,23% and 2,58%. In sequence assembly segments of DNA with near-identical sequence (segmental duplications), accounting for ~5% of the human genome, can result in sequence miss-assignment and wrong assembly of the sequenced parts. It is estimated that ~1.3% of the overall sequence of the June 2002 human genome draft sequence are erroneous due to assembly errors [4].

Quality checking

Reliable quality checks in DNA sequence determination can be performed only after base detection. However, only fatal experimental errors are detected, by searching for abnormal output display characteristics. Individual sequence transformation errors cannot be detected this way. Particularly, so-called *frame-shifts* are a major problem, i.e., a missing or inserted base in the sequence string. When translating these sequences, the resulting protein has a completely different sequence, because it is translated out of frame. There are techniques for detecting such errors, but they mainly rely on the correct protein sequence already being in a database. To receive error free sequences, each part is sequenced multiple times.

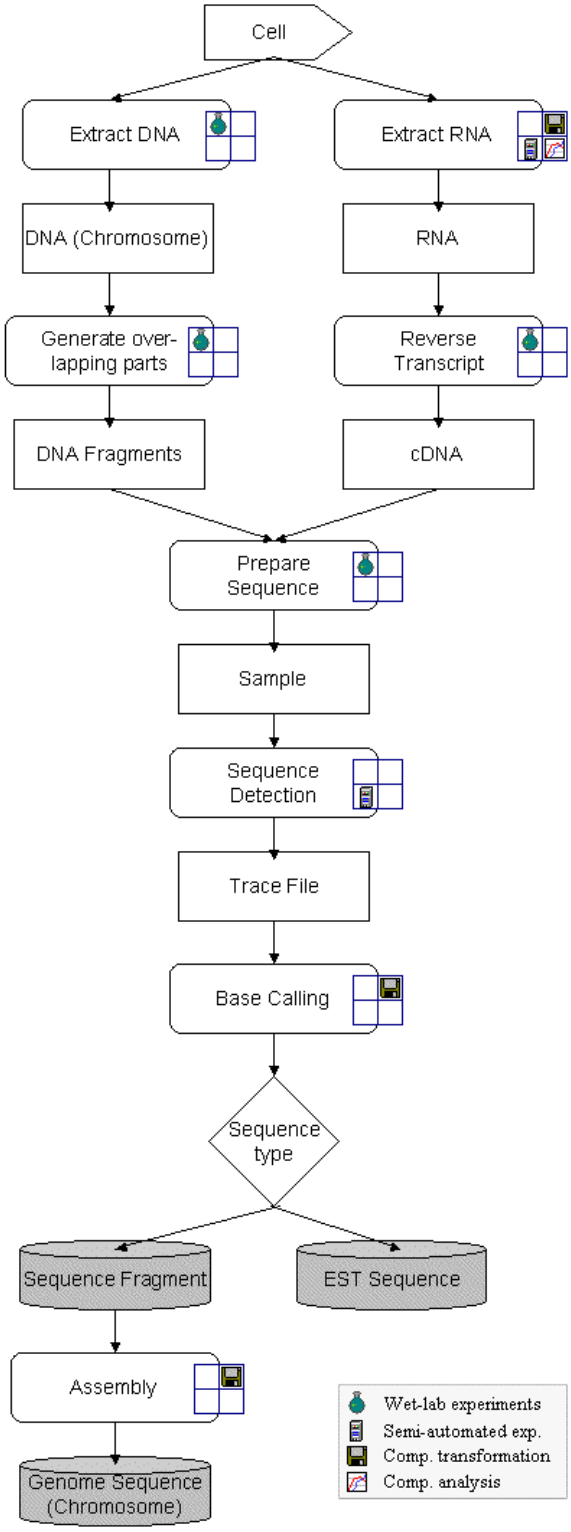

Figure 4: DNA Sequence Determination Process

4.2 Genome Feature Annotation

Genome feature (or structural) annotation results from performing a set of operations on the genome sequence data, e.g. sequence alignment or pattern search, making use of existing genome data and their annotations, e.g. aligning EST sequences against the genome to identify transcribed regions. Interpretation and combination of the results is guided by expert knowledge in form of *annotation rules*. These rules form the *annotation pipeline*, i.e., the description of the information production process. Often alternative ways for genome feature annotation are used, depending on the expert's preferences.

Errors

Errors in genome feature annotation include analysis errors, propagated errors, and stale data. They result in incorrect structural annotations.

Analysis errors: Incomplete or uncertain domain knowledge, or careless interpretation of operation results can lead to misinterpretation and erroneous annotations. For example, predicting genes by simply using the occurrences of start/stop-codon pairs results in a high number of wrongly predicted genes.

Propagated Errors: Errors in the genome sequence or genome data used within the annotation pipeline are propagated through the pipeline and result in misinterpretations and annotation errors later on. Sequence errors imply non-existent patterns or miss existing ones. Errors within additional data, e.g. EST sequences, can lead to operation results that cause erroneous interpretations.

Stale data: Annotation based on outdated data yields results different from annotations based on current data, causing inconsistency. The fact that changes to data items for the most part remain unnoticed by the depending data items is a major problem within genome data.

The errors in genome feature annotation are further classified as:

- false positives, e.g., parts classified as gene which are not coding for a protein,
- false negatives, e.g., parts not classified as gene which are coding for proteins, and
- Incomplete or partially (in-) correct information. This information, e.g., the uncertain start codon of a gene, is still included in several databases to avoid information loss. For example, in ENSEMBLE (Version 7.29, [12]) 36.77% of the predicted transcripts were incomplete.

Quality checking

Quality checks are performed only marginally within the process. There is the possibility to define integrity constraints that have to be satisfied by the resulting data. One possible constraint is that the start codon of a gene is `atg`. However, not many helpful constraints are known, many have exceptions, and constraints are often not enforced to avoid information loss or because constraint checking has to be performed by manual inspection or complex programs using additional data sources. Another quality checking method is to mine for errors, i.e., to detect outliers within the feature data, e.g., genes that are abnormally short or long.

4.3 Protein Sequence Determination

The process of protein sequence determination is shown in Figure 5. Computational protein sequence determination translates the predicted gene sequences using the genetic code of the specified organism. The protein sequence can also be determined experimentally making it independent of the two other steps performed before. Protein sequencing is hard to automate and this is why computational sequence translation is the preferred method. Often, a combination is used by determining a starting sequence (prefix) of the protein experimentally and then using it to search existing protein or translated DNA databases for proteins matching the exact prefix.

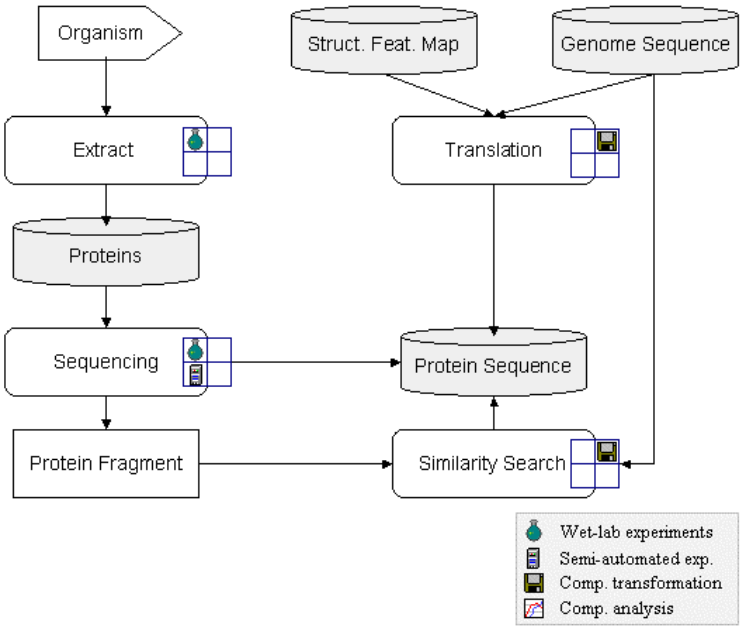

Figure 5: Protein Sequence Determination Process

Errors

The classes of errors resulting from protein sequence determination are experimental errors for experimental sequence determination and transformation errors, propagated errors, and stale data for computational sequence determination.

Experimental errors: As for DNA sequence detection, experimental errors result from poor experimental setup, practices or from failure of chemical reactions within the process.

Transformation errors: Using the wrong genetic code within the translation step, possibly because of an erroneous value within the attribute specifying the organism, results in a string incorrectly representing the protein sequence.

Propagated errors: Incorrect sequences or frame-shifts result in proteins with sequence representations different from the actual amino acid sequence of the protein in the cell. Incorrect feature annotation again yields false positives, i.e., proteins that are non-existent in the organism. Incomplete or partial information results in incomplete protein translations.

Stale data: Changes to the DNA sequence of the translated gene have to be reflected in changes to the resulting protein sequence. As those changes often remain unnoticed, the translated protein sequences become erroneous.

Quality checking
Quality checks can be performed only on the resulting protein sequence. Here, we can search automatically for proteins of uncharacteristic length or by hand for unusual amino acid patterns within the protein sequences. This covers only very few errors. Reliable and efficient checking of correct sequences would require inexpensive, fast, and automated protein sequencing methods.

4.4 Protein Function Annotation

Protein function annotation is performed experimentally or computationally. Computational functional annotation is based on the fact that the protein sequence determines protein function. There are two main techniques for computational annotation of proteins. Often, they are combined to form an annotation pipeline.
The first is based on protein similarity. It is assumed that two proteins with similar sequence very likely possess the same function. By searching databases of already annotated proteins for similar ones, the annotated function is transferred onto the sequence under consideration. The second technique searches for the occurrence of motifs in a protein. Each motif has an annotated function that is assigned to the query protein in case of motif occurrence.

Experimental protein function annotation is much more reliable but also much more time consuming. A typical method is to generate genetically manipulated organisms not containing the gene for the protein under consideration and to observe how the behavior or the phenotype of the organism changes.

Errors
Experimental errors, analysis errors, propagated errors, and stale data are the classes of errors within this step, again depending on the method used.

Experimental errors: Due to the numerous experimental techniques for experimental protein function annotation there are correspondingly oodles of possible errors that yield improper annotation of protein function.

Analysis errors: A major problem within protein annotation using sequence similarity is to define the degree of similarity between proteins, so they can be considered to have identical function. In [6] it is stated that sequence similarity above 25% for proteins having minimum length of 100 amino acids is sufficient, while a similarity below 15% does not allow annotation transfer. In the interval between 15% and 25%, the proteins may very well be related, but additional studies must be performed to achieve higher confidence. Furthermore, similarity might not be present in the region that is responsible for the actual function of the annotated protein. Unfortunately, the responsible region is not always explicitly annotated. Thus, transferring function is error-prone. Insufficient inspection and careless usage of similarity search results very easily and very often lead to erroneous annotations.

Propagated errors: The huge amount of protein data produced by computational translation requires the application of computational annotation. Often, annotations are not marked as putative and used carelessly by other biologists. This causes a large degree of propagated erroneous annotations.

Stale Data: As for propagated errors, stale data causes problems because of the high degree of data dependency for the results of the annotation process and because of changes to the annotation of proteins. This happens frequently because computational annotations are reproduced and changed due to experimental verification.

Quality checks
Quality checks involve performing additional studies to collect arguments for or against the correctness of the annotation. Unfortunately, these checks are often not performed, and thus most annotations have low confidence. Verification of protein function annotation requires the tedious documentation of the annotation process and the results of the operations performed and decisions made, which also is missing in most of the databases managing protein function annotation.

5 CHALLENGES FOR DATA CLEANSING

As we have shown, genome data is erroneous by nature – due to its production process. The data is produced by experiments that are error prone and by domain experts who analyze the data in a subjective manner using uncertain knowledge and in turn invalid, uncertain, or incomplete data. Despite syntactic errors, such as format inconsistency, duplicates (synonyms), homonyms and syntax errors and vocabulary usage in textual annotation, as reported in [2], these problems lead to semantic errors, i.e., the resulting information does not represent the real-world facts correctly. Data dependencies inherent to the production process and to the usage of the data make genome data predestined for propagated errors. Also, there are frequent changes in the data and knowledge that in many cases remain unnoticed to systems storing derived data.

Two approaches could eliminate many of the data quality issues raised in the previous sections *at production time*. First, to keep pace with the analysis of the huge amounts of data produced, reliable methods for genome data production could be employed, i.e., using repeated experimental methods and less automation for data analysis. Second, quality checks within the production process could be employed. Quality checks are often omitted, because usually they require manual inspection and the huge amount of data makes them time-consuming and expensive. Within the domain of genome data there also exist only few reliable constraints and a multitude of exception hindering effortless verification of data correctness. The multitude of different sources necessary for result comparison and verification poses another problem with genome data. There is no standard format for genome data storage and no commonly accepted vocabulary. This hampers integrated access and makes data transformations for standardization and normalization necessary.

The afore-mentioned reasons make data cleansing a necessity for genome data *after the data is produced*. Most existing work [1,7,11,14,19,24,29] focuses on data transformation, enforcement of simple integrity constraints, and duplicate elimination. Existing cleansing approaches are mainly concerned with producing a unified and consistent data set, i.e., addressing primarily syntactical problems and ignoring the semantic problem of verifying the correctness of the represented information. The problem of duplicates is also existent in genome data, but duplicates are less interfering than in other application domains. Duplicates are often accepted and used for validation of data correctness. In conclusion, existing data cleansing techniques do not and cannot consider the intricacies and semantics of genome data, or they address the wrong problem, namely duplicate elimination. We see three concrete and reasonable challenges for genome data cleansing in the near future.

Credibility checking and re-annotation: The most reliable way for semantic error correction in genome data is to re-perform experiments and computational analysis under careful control by domain experts. As already mentioned, this is time-consuming and expensive and it is also performed for already correct values, yielding a large amount of unnecessary computation and experiments. We therefore perform credibility checking on the data to identify those yielding evidences for being erroneous and re-annotated them. The correction of erroneous data still has to be performed by re-computation, unless one is able to choose from a set of alternatives the one that is considered as correct. Credibility checking is also a very important technique for genome data production if the correctness of data is verified before it is used within other processes.

A first idea is to use the huge amount of additional and redundant data in other sources, assuming the possibility of integrated access to them. The information is then clustered into sets of related information, i.e., information about the same real-world fact produced by different research groups. By detecting and highlighting contradictions or accordance, one collects arguments for or against the correctness of data items and makes them usable for further processing and credibility checking. The main challenge here is to efficiently and reliably identify the overlapping information.

Another way is to perform integrity constraint checking to collect arguments for or against the correctness of a certain data value. These arguments are generated by domain dependent evidence functions. These are functions that operate on existing data and check or assess known biological facts and rules. For example certain amino acid combinations are known to be non-existent in proteins or the evidence for a predicted coding region is high if similar regions exists within other organisms, i.e., the sequence region is conserved. There is a difference between hard and weak constraints, i.e., constraints where violation is a clear indication for an error or only for the possibility of an error. There have to be rules that define when to assume a data item to be erroneous and ready for re-processing. The methods used within the constraints are the same as those within the annotation pipelines, e.g. sequence alignment, pattern search, statistical methods about sequence composition, etc.

Metadata management: One of the main problems in verifying the correctness is the missing metadata about how the information was gained and what other information and interpretations it is based upon. For derived data the information used within the production process is called the *data lineage*. Data lineage can be used for keeping annotation up-to-date in a changing environment without a re-annotation every time parts of the base data changes. A data cleansing framework for genome data has to be able to detect and react on changes in the base data without re-performing the complete and expensive data cleansing process. By using the data lineage the data items that depend on the changing data are easily identified for re-annotation.

Alternative solution management: In data cleansing it is often impossible to find the correct solution immediately. Instead, there often exists a set of alternative solutions. These solutions have to be managed up to a point in time where one is able to decide which is the correct value. Until then the alternatives have to be included within the process of further data production. After deciding which is the correct solution from a set of possible solutions, one has to be able to undo decisions that were based on data that has become obsolete now. For this purpose the data lineage collected is used to identify the depending data items.

6 CONCLUSIONS AND OUTLOOK

Genome data is dirty and this state is caused by inadequacies of the data production process. We presented typical cases and classes of errors and gave reasons why errors cannot be avoided simply by changing parts of the production process. Only prohibitively expensive quality checking within the process, using quality checking modules, can increase quality during data production. This leads to the problem of how to eliminate existing errors through data cleansing methods. We have shown that existing methods are not applicable for the major errors found in genome data.

Cleansing of genome data is closely related to genome annotation. Both require domain dependent evidence functions. The definition of a set of general evidence functions for the domain of genome annotation will enable us to build a formal model to specify the annotation and cleansing process. The intrinsic properties of these individual functions can then be used to detect erroneous annotations without the necessity of complete re-annotation.

In those cases where alternative solutions and evidence values for them are managed it is desirable to include them within the annotation and cleansing process to receive results of higher quality. Some of the genome databases are also beginning to manage such evidences for their entries. Credible annotations can be derived by excluding invalid or unreliable entries from the processing. The formal model for genome annotation has to take these evidences into account.

Including the management of cleansing lineage within the model further enables efficient detection and re-annotation of affected annotations when changes in external data sources occur.

REFERENCES

[1] R. Ananthakrishna, S. Chaudhuri, V. Ganti, *Eliminating Fuzzy Duplicates in Data Warehouses*, Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002

[2] P. Bork, A. Bairoch, *Go hunting in sequence databases but watch out for the* traps, Trends in Genetics, Vol 12, No. 10, 1996, 425-427

[3] S.E. Brenner, *Errors in genome annotation*, Trends in Genetics, Vol. 15, No. 4, 1999, 132-133

[4] J. Cheung, X. Estivill, R. Khaja, J.R. MacDonald, K. Lau, L.-C- Tsui, S.W. Scherer, *Genome-wide detection of segmental duplications and potential assembly errors in the human genome sequence*, Genome Biology 2003, 4:R25

[5] D. Devos, A. Valencia, *Intrinsic errors in genome annotation*, Trends in Genetics, Vol. 17, No. 8, 2001, 429-431

[6] R.F. Doolittle, *Of URFs and ORFs – A Primer on How to Analyze Derived Amino Acid Sequences*, University Science Books, 1987

[7] H. Galhardas, D. Florescu, D. Shasha, E. Simon, C.-A. Saita, *Declarative data cleaning: Language, model, and algorithms*, Proceedings of the 27 th VLDB Conference, Roma, Italy, 2001

[8] R. Guigó, P. Agarwal, M. Burset, J.W. Fickett, *An Assessment of Gene Prediction Accuracy in Large DNA Sequences*, Genome Research, Vol. 10, 2000, 1631-1642

[9] L.M. Haas, P.M. Schwarz, P. Kodali, E. Kotlar, J.E. Rice, and W.C. Swope. *DiscoveryLink: A system for integrated access to life sciences data sources*. IBM Systems Journal, Vol. 40, No. 2, 2001,489-511

[10] S. Hensley, *Death of Pfizer's 'Youth Pill' Illustrates Drugmakers Woes*, The Wall Street Journal online, May 2, 2002.

[11] M.A. Hernández S.J. Stolfo. *Real-world data is dirty: Data cleansing and the merge/purge problem*. Data Mining and Knowledge Discovery, Vol. 1, 1998, 9-37

[12] T. Hubbard, et al., *The Ensembl genome database project*, Nucleic Acids Research, Vol. 30, No. 1, 2002, 38-41

[13] I. Iliopoulos, et al., *Evaluation of annotation strategies using an entire genome sequence*, Bioinformatics, Vol. 19, No. 6, 2003, 717-726

[14] M.L. Lee, T.W. Ling, W.L. Low, *IntelliClean: A knowledge-based intelligent data cleaner*, Proceedings of the ACM SIGKDD, Boston, USA, 2000

[15] J. Long, C Seko. *A new method for database quality evaluation at the Canadian Institue for Health Information*. In Proceedings of the International Conference on Information Quality (IQ), Cambridge, MA, 2002, 238-250

[16] S. Madnick, Wei Zhang, R.Y. Wang. *A framework for corporate householding.* In Proceedings of the International Conference on Information Quality (IQ), Cambridge, MA, 2002, 36-46

[17] F. Meyer, et al., *GenDB – an open source genome annotation system for prokaryote genomes*, Nucleic Acid Research, Vol. 31, No. 8, 2003, 2187-2195

[18] S. Mohan, M.J. Willshire, C. Schroeder. *DataBryte: A proposed data warehouse cleansing framework*. In Proceedings of the International Conference on Information Quality (IQ), Cambridge, MA, 1998, 283-291

[19] A.E. Monge, C.P. Elkan, *An efficient domain-independent algorithm for detecting approximately duplicate database tuples*, Workshop on Data Mining and Knowledge Discovery, Tucson, USA, 1997

[20] H. Müller, J.-C. Freytag, *Problems, Methods and Challenges in Comprehensive Data Cleansing*, Technical Report HUB-IB-164, Humboldt-Universität zu Berlin, Institut für Informatik, 2003.

[21] C.A. Ouzounis, P.D. Karp, *The past, present and future of genome-wide re-annotation*, Genome Biology, Vol. 3, No. 2, 2002, comment2001.1-2001.6

[22] F. Piontek, H. Groot, *Healthcare informatics: Data quality, warehousing and mining applications*. In Proceedings of the International Conference on Information Quality (IQ), Cambridge, MA, 2002, 256-260

[23] E. Rahm Hong Hai Do. *Data cleaning: Problems and current approaches*. IEEE Data Engineering Bulletin, Vol. 23, No. 4, 2000, 3-13

[24] V. Raman, J.M. Hellerstein, *Potter's Wheel: An Interactive Framework for Data Transformation and Cleaning*, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

[25] P. Richterich, *Estimation of Errors in „Raw" DNA Sequences: A Validation Study*, Genome Research, Vol. 8, 1998, 251-259

[26] G. Shankaranarayanan, R.Y. Wang, M. Ziad: *IP-MAP: Representing the Manufacture of an Information Product.* In Proceedings of the International Conference on Information Quality (IQ), Cambridge, MA, 2000, 1-16

[27] A.P. Sheth, J.A. Larson, *Federated database systems for managing distributed, heterogeneous, and autonomous databases*, ACM Computing Surveys, Vol. 22, No. 3, 1990, 183-236

[28] G. Stoesser, et al., *The EMBL Nucleotide Sequence Database: major new developments*, Nucleic Acids Research, Vol. 31, No. 1, 2003, 17-22

[29] P. Vassiliadis, Z. Vagena, S. Skiadopoulos, N. Karayannidis, T. Sellis, *ARKTOS: towards the modeling, design, control and execution of ETL processes*, Information Systems, Vol. 26, 2001, 537-561

[30] W. Winkler. *Methods for evaluating and creating data quality*. In Proceedings of the International Workshop on Data Quality in Cooperative Information Systems (DQCIS), Siena, Italy, 2003