Big Data Analytics in Healthcare: A Cloud based Framework for Generating Insights

Ashiq Anjum[1], Sanna Aizad[1], Bilal Arshad[1], Moeez Subhani[1], Dominic Davies-Tagg[1], Tariq Abdullah[1], Nikolaos Antonopoulos[1]

[1]College of Engineering and Technology, University of Derby

{A.Anjum, S.Aizad, B.Arshad, D.Davies-Tagg, M.Subhani, T.Abdullah, N.Antonopoulos}@derby.ac.uk

Abstract With exabytes of data being generated from genome sequencing, a whole new science behind genomic big data has emerged. As technology improves, the cost of sequencing a human genome has gone down considerably increasing the number of genomes being sequenced. Huge amounts of genomic data along with a vast variety of clinical data cannot be handled using existing frameworks and techniques. It is to be efficiently stored in a warehouse where a number of things have to be taken into account. Firstly, the genome data is to be integrated effectively and correctly with clinical data. The other data sources along with their formats have to be identified. Required data is then extracted from these other sources (such as clinical datasets) and integrated with the genome. The main challenge here is to be able to handle the integration complexity as a large number of datasets are being integrated with huge amounts of genome. Secondly, since the data is captured at disparate locations individually by clinicians and scientists, it brings the challenge of data consistency. It has to be made sure that the data consistency is not compromised as it is passed along the warehouse. Checks have to be put in place to make sure the data remains consistent from start to finish. Thirdly, to carry this out effectively, the data infrastructure has to be in the correct order. How frequently the data is accessed plays a crucial role here. Data in frequent use will be handled differently than data which is not in frequent use. Lastly, efficient browsing mechanisms have to put in place to allow the data to be quickly retrieved. The data is then iteratively analysed to get meaningful insights. The challenge here is to perform analysis very quickly. Cloud Computing plays an important role as it is used to provide scalability.

Keywords: Big Data, Cloud Computing, Analytics, Healthcare data, Genomics data, Graph models, Tiered data storage,

Introduction

With exabytes of data being generated from genome sequencing, a whole new science behind genomic big data has emerged. Adding to that, the recent advances in storage and processing technologies has enabled the generation, storage, retrieval and processing of exabytes of genomic and healthcare data in electronic form. As technology improves, the cost of sequencing a human genome is going down considerably, and, in turn has increased the number of genomes being sequenced. Handling huge amounts of genomic data along with a vast variety of clinical data using existing frameworks and techniques has become a challenge.

There is a wide interest in genomic data because it can allow meaningful insights to be generated. These insights could range from a variety of things including genomic research as well more practical uses such as personalised medicine for a particular genome. Genomics is producing data size 2-40 EB/year (Stephens, et al., 2015) which is stored in local databases or in cloud storage. Cloud computing is used for storage, distribution and processing of this data so that applications can run on remote machines that already have access to data (Stephens, et al., 2015).

A data platform that integrates genomics/healthcare data while enabling quick and efficient analysis would allow extraction of practical insights in a short frame of time. Developing such a platform poses a number of challenges on its own. These challenges relate to integrating genomics and clinical data sources, ensuring consistency of the integrated data and developing a big data platform that stores and manages the integrated data. An overview of these challenges and a brief description of the proposed framework is provided in the remainder of this section.

With respect to integration of big data, it is imperative to maintain the consistency of data between the data sources and the data warehouse. Since the data is of the magnitude of exabytes the issue converges to Big Data analytics. Infrastructures such as that provided over cloud are required to ensure that the consistency is maintained between the data sources and the warehouse. In a clinical information management environment, data consists of heterogeneous data sources with multitude of data types at distributed locations. Clinicians and scientists generate data which is individually captured at disparate locations and brought together to a warehouse for reporting, decision support and data analysis. This data needs to be correctly integrated in order to ensure the consistency and coherence of the system at large. Any inconsistency may result in breaking the data warehouse, which in turn would affect the reports being generated (examples include quarterly comparisons and trends to daily data analysis) and bio-statistical analysis among other things. Therefore, there is a need for structured migration and integration of data between the sources and the data warehouse to ensure that the integrity of the warehouse can be maintained. In such an environment coherence and consistency of data is imperative in order to protect the integrity of the warehouse. Since the data from heterogeneous sources is in exabytes, it is essential to provide a scalable environment for clinical analytics. A possible solution is the provision of a scalable environment for clinical data integration and system integrity based on graphs. The infrastructure provided for such an environment needs to take the frequent use of data into account. . Large scale graph processing systems such as Giraph (Giraph, 2016) and GraphLab (Low, et al., 2014) provide support for data consistency by providing configurable consistency models.

The infrastructure of the system should be such that it should allow frequently used data to be quickly retrieved when required, whereas the data which is not in much use should be allowed to reside in the system. Technologies such as Hadoop make storing a large scale of data trivial, but Hadoop by itself is often not an ideal platform for working with data and performing the levels of complex analysis and interactive querying often afforded to data warehouses (Borthakur, et al., 2011) (Songting, 2010). Thus, in order to store huge amounts of data in a cost-effective and time-efficient manner and deliver a high standard of analytics performance, Hadoop's scalability may be used to accommodate storing data. On the other hand, there is a need to maintain existing scale-up data warehouses and analytics environments to provide the fast and efficient analysis people expect. But using both technologies can only work if we move data between environments when required.

Generating insights from the integrated data is only possible after developing suitable infrastructure for storing and retrieving the data. Analyzing this data is a user-driven and iterative non-trivial task. In a lot of cases, the data needs to be revisited several times in order to get the required insights. Different challenges and their solutions are discussed.

This chapter proposes a cloud based framework for integrating genomics/ healthcare data in a big data platform which would enable users to generate meaningful insights in their domain. The platform provides a solution to the challenges discussed above. The rest of the chapter is organized as follows: Section 2 introduces genomics and clinical datasets. Section 3 explains the integration of these datasets. An approach for maintaining data consistency during and after the integration is explained in Section 4. The infrastructure for storing the data is explained in Section 5, whereas, Section 6 explains the data analytics approaches for generating insights from the data and Section 7 concludes the chapter.

Genomics and Clinical Data

The cloud based data analytics platform focuses on integrating the genomics and clinical datasets and on generating insights from the integrated data. It is important to understand these dataset before introducing the cloud based data analytics platform.

Genomics Data

The genetic makeup of an organism is responsible for coding its different characteristics. A complete set of genetic information is contained in the genome, which consist of genes. The genes are a sequence of four different molecules known as nucleotide bases: Adenine (A), Guanine (G), Thymine (T) and Cytosine (C). Different combinations and frequency of these nucleotides generate a huge variety of genes within a genome. Understanding the constitution of these genes was a mystery until development of sequencing methods. The 1970s and 1980s saw manual DNA sequencing methods such as Maxam-Gilbert sequencing (Maxam & Gilbert, 1977) and Sanger sequencing (Sanger & Coulson, 1975). Automated sequencing methods such as Shotgun sequencing were introduced in the 1990s. Over the next decade, scientists were able to sequence unicellular and multicellular organisms using these methods. It wasn't until 2001 that the human genome was completely sequenced. By 2005, next-generation sequence (NGS) technologies (Metzker, 2010) were introduced.

Before sequencing, other techniques such as genome-wide association studies between thousands of individuals were used because genome sequencing was an unthinkable thing to do. However, as technologies advanced, the sequencing market has become very competitive in recent years. Many platforms, such as Illumina (16Il), 454 Life Sciences (1645) and Complete Genomics (16Co) to name a few, are available commercially for research and clinical use.

Sequencing is now the first step for research investigating the genome at the basic level. Genome sequencing technology takes a sample of the genetic material in a test tube and converts it to a string of As, Gs, Ts and Cs representing the genome and stores it into a text file. A human genome consists of 3 billion bases. The size of a text file containing these is, on average, 6 X 10^9 bits.

As the cost of sequencing is decreasing (Fig. 1.), more and more genomics data is becoming readily available sparking several initiatives such as 1000 Genomes Project (1000 Genomes Project Consortium, 2010) and the 100,000 Genome Project (Brierly, 2010). One of the aims of initiatives like these is to discover medical insights especially for more serious diseases such as cancer.

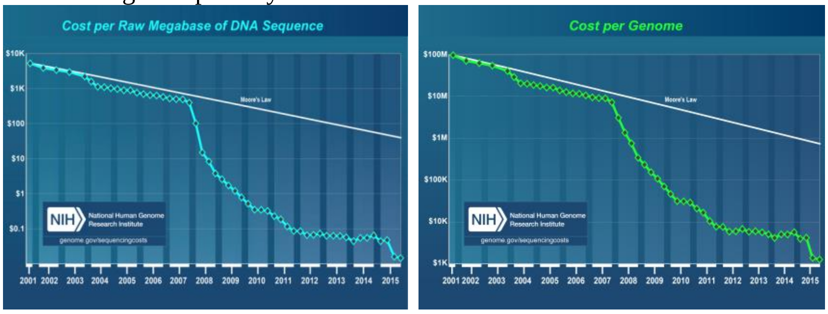

Fig. 1. Cost per Raw Megabase of DNA Sequence & Genome over the years. Published by National Human Genome Research Institute (NHGRI) (National Human Genome Research Institute, 2016)

Clinical Data

Clinical data sets are generated during the course of ongoing patient care or as part of normal clinical trial program. Major sources include electronic health records, claims data, disease registries, health surveys, clinical trials data and administrative data. These are a vital source for health and medical research.

Data Integration

Data integration is the first challenge while developing a cloud based data analytics platform. The data sources in clinical research domain are much diversified, such as health records, clinical trials, disease records etc. On the other hand, the genomics data sets are generally very data intensive such as genome sequences, variants, annotations and gene expressions data sets etc. Due to the massive size of genomics data sets, the problem of integration enters into the domain of big data problems. Integrating these data intensive genomics sources with a set of diversified clinical sources is a considerable challenge that chiefly implies building a database capable of containing heterogeneous data types.

The clinical data ranges from patients health records, diagnostics tests results including laboratory reports and imaging scans, disease history, to hospital administration and finance data. These data sets are captured in different repositories, such health records maintained by each hospital or clinical trials conducted by state or different pharma or non-profit organizations. The clinical data sets within these repositories are comprised of a large variety of parameters within a single study, and then there are further variations among parameters across different studies, as per the requirement of underlined research. Integrating this large variety of parameters of various data types across multiple studies is a challenging problem in itself because the integrated clinical data should have an intuitional output.

The next challenge is to integrate these parameters with genomics data sets. Traditionally, the information about genomics is not captured in the clinical data sets. Therefore, the genomics data is only available from separate genomics sources, mainly the repositories such as NCBI, Ensembl or 1000 genome projects. These data types are, therefore, different from those of clinical data sets. Hence, in order to integrate them with clinical data sets, the challenge is to make the data types compatible with each other so that they can be consolidated within a single warehouse.

Combining data sets from different clinical sources with genomic data can help understanding a clinical problem at a deeper level by empowering it with genomics background information. This big data integration may help to delve into genetic background of clinical problems, which will ultimately aid various users of these data sets. The major benefit, that can be foreseen from clinical and genomics data integration, will be to design personalized treatments for patients.

Pharmacogenomics industry can also gain the advantage to provide more personalized solutions to healthcare, such as designing drugs with improved efficacy. Researches from both clinical and genomics domains can also use the integrated data to discover the insights of complicated biological problems, such as finding new biomarkers. Hence, it can be estimated that data integration could help every academic or industrial institution related to these dimensions of medical science.

There exist some clinical data integration solutions, such as those provided by SAS (SAS CDI), Edifecs (Edifecs CDI), Lumeris (Lumeris CDI) etc., but they are only focused on data management and administration purposes and are not targeted for clinical research. These solutions target combining various clinical data sets from different sources and providing them from a single platform. However, there are no solutions for clinical and genomics data integration available hitherto. Due to absence of any data model that can accommodate both clinical and genomics data sets, there is a need to design and construct such a data model which provides a single platform access to both domains.

In the last decade, increasing trend has been observed in this direction of research. Researchers have studied and proposed various integration models for integrating multi-omics data. The two most common approaches that can be found in literature are multi-stage analysis and meta-dimensional analysis.

Multi-stage analysis is a stepwise or hierarchical analysis method. It helps to reduce search space by stage wise analysis (Ritchie, Holzinger, Li, Pendergrass, & Kim, February 2015). It essentially analyses and integrates only two data types at a time while analysing across the data space. Triangle method is the most common method under this approach which has been widely used for association studies. This method is more commonly used for SNP (single nucleotide polymorphism) associations with expression data and genes themselves (Ritchie, Holzinger, Li, Pendergrass, & Kim, February 2015) (Lee, Cho, Kim, & Park, 2009). Some clinical phenotypes can be result of interaction between different genes and multiple clinical parameters. Due to step-wise analysis, this approach cannot capture those phenotypes which are determined by factors acting from various sources. It is a robust and rather simple approach, however, it is not recommended when multiple different sources are required to be integrated (Ritchie, Holzinger, Li, Pendergrass, & Kim, February 2015) (Hamid, et al., 2009).

Meta-dimensional studies involve simultaneous analysis of all the data sources to produce complex models (Ritchie, Holzinger, Li, Pendergrass, & Kim, February 2015). There are various methods under this approach, each of which is based on a different data model. The approach can be selected according to the underlining research goals. Either the multiple data sets are integrated prior to building a common model on them, or an individual model is built on each data set before integrating them together, as illustrated in Fig. 2. Bayesian networks and neural networks have been more commonly observed in the integration based research (Fridley, Lund, Genkins, & Wang, 2012) (Akavia, et al., 2010). Meta-dimensional approach facilitates the capability to search across various data types among multiple data sets. This vast search capability aids to detect those phenotypic traits which are caused by mutual interaction of multiple factors from different clinical and genomics sources. Although this integration using meta-dimensional approach leads to a rather complex and less robust models, but it helps to search across a wider spectrum of data types (Ritchie, Holzinger, Li, Pendergrass, & Kim, February 2015) (Hamid, et al., 2009).

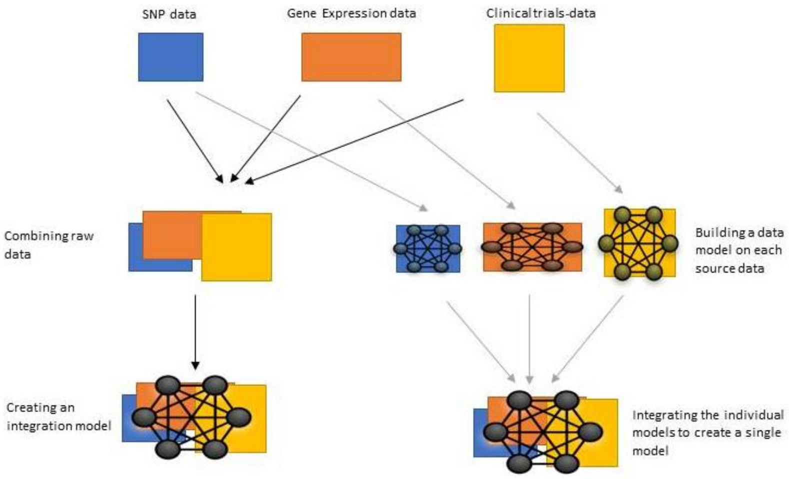
SNP data
Gene Expression data
Clinicaltrials-data
Combining raw data
Building a data model on each source data
Creating an integration model
Integrating the individual models to create a single model

Fig. 2. An illustration of meta-dimensional approach

Due to the huge size of genomics data sets, and large variability of clinical parameters, it is not viable to integrate all parameters. Only those parameters should be integrated which may provide deeper intuition after integration. Most researchers have identified gene expression data and SNP data sets to be most relevant to integrate with the clinical data. Since, determining the gene expression of SNPs can help to find out ultimate effects of a gene on a phenotype, therefore, these parameters have been widely seen to be integrated with clinical data in research (Nevins, et al., 2003) (Louie, Mork, Martin-Sanchez, Halevy, & TarczyHornoch, 2005) (Ritchie, Holzinger, Li, Pendergrass, & Kim, February 2015) (Lee, Cho, Kim, & Park, 2009). For future prospects, the research can be further extended to incorporate additional genomics parameters for integration, such as annotations data.

A promising solution to integrate the clinical and genomics data will be to design a relational data model based on meta-dimensional approach and implement it within a data warehouse. Since meta-dimensional approach provides a wider search spectrum, therefore, this approach seems more promising to be implemented for clinical and genomics data integration where a wide variety of parameters and large data sets are required to be integrated. Out of various meta-dimensional approaches, graph based models seems more promising such as Bayesian networks (Fridley, Lund, Genkins, & Wang, 2012) (Holzinger & Ritchie, 2012 January). A probabilistic schema can be designed to implement on this data model. Some previous work shows that star-based schema can be designed for biomedical data (Wang, Zhang, & Ramanathan, 2005) (Salem & Ben-Abdallah, 2015). These schema designs can be adopted and modified to meet the requirements of the data sets and data warehouse under consideration. The performance and scalability of the integration model will be a critical factor to be controlled in this case. If the model is not capable of scaling to larger data sets, or it fails to provide same performance with larger data sets, then such a model will not be sustainable for a futuristic model.

Data Consistency

Ensuring consistency of integrated data is a crucial part of the big data analytics platform. Data coming from heterogeneous sources requires to be effectively integrated to ensure the coherence of the source data and the warehouse (Salem & Ben-Abdallah, 2015). A change in one of the data sources not only affects the data in that data source but also affects the inter-relationships between the multiple data sources. As the structure of the data warehouse is defined based on the structure of the individual data sources and based on the inter-relationships between the sources, a single change has the potential to significantly impact the warehouse. More importantly, the data in the warehouse may not be consistent with the data in the data sources when a change occurs in the data source. This is turn means that the inconsistent changes might result in breaking the data warehouse. Evolution of clinical data results is one such example of inconsistent source change that needs to be reflected in the data warehouse. Since the data from these sources is of the magnitude of petabytes the challenge of data consistency emerges as a part of the Big Data domain. Furthermore in context of big data applications, it is imperative to maintain data consistency across the entire spectrum of application to ensure correct results and traceability of individual elements in the system.

One of the prime issues in an evolving data warehouse environment is the dynamic nature of sources. The evolving nature of sources can lead to breaking the data warehouse which is a major issue in maintaining data consistency. Inconsistent changes can lead to generation of inaccurate reports such as those based on personalised patient analysis further leading to incorrect diagnosis. In order to prevent the system from breaking due to inconsistent changes, this endeavour aims to explain a possible solution to ensure consistency between the heterogeneous data sources and the clinical data warehouse. As explained in the previous section once the data has been integrated, consistency mechanisms need to ensure that the sources and data warehouse are consistent and reflects the evolving data from clinical data sources.

In order to prevent the breaking of data warehouse from the evolving changes in the data sources, a possible solution is the use of graphs to ensure the coherence and consistency of data between the sources and the warehouse. Graphs can scale well to represent millions of entities in a clinical domain (Rodriguez & Neubauer, 2010) thus allowing to ensure the scalability of the system. This is of particular interest in the domain of clinical data since integrating data from disparate sources will be of a much higher magnitude compared to the data coming from sources. Graphs are governed by graph models that allow a flexible and uniform representation of data originating from heterogeneous sources. This study aims to investigate suitable graph data models for accurate representation of data both at the source and data warehouse level. Furthermore graph models provide the ability to predict functional relationships between heterogeneous data sources in order to ensure the correctness of source data with respect to the data warehouse. Thus the need for a scalable environment for clinical analytics arises to ensure the integrity of a data warehouse without compromising the integrity of the clinical data warehouse. Existing state of the art graph analytical systems do not fully encompass the needs for such a system.

In conjecture with source data, another key component in a data warehouse environment is metadata (Harris, et al., 2009). Metadata describes the context in which the data was collected and hence means to query the sources. Since the data comes from distributed sources a lot of research deals with capturing metadata at the source level. Any change occurring at the source needs to be reflected in the metadata repository by updating it, leading to generation of new metadata. Both the updated and prior metadata are essential to aid in the replication and integration of sources. For the purpose of our research work we will be looking at the metadata repository knows as Semantic Manager (Akana, 2016) by Akana. Semantic Manager enables enterprises to define, understand, use and exchange data by managing standards and metadata as organizational assets.

Several approaches have been investigated for clinical data integration that help to ensure data consistency such as integration engines (Karasawas, Baldock, & Burger, 2004), (Sujasnsky, 2001) or ontology based data integration (Lapatas, Stefanidakis, Jimenez, Via, & Schneider, 2015). Integration engines provide a useful way of solving the basic communication problems between systems, but they do nothing to address true integration of information particularly in the context of data consistency (Karasawas, Baldock, & Burger, 2004), (Sujasnsky, 2001). This approach works well and has been effective, but when the number of possible interactions between systems increases, the limitations of scalability becomes apparent. The use of graph based integration of data being generated from multiple data sources is a viable option to address this issue (Rodriguez & Neubauer, 2010).

Graphs (Rodriguez & Neubauer, 2010), (Park, Shankar, Park, & Ghosh, 2014), are particularly useful for the description and analysis of interactions and relationships in a clinical domain. Graphs provide useful features such as analytical flexibility, in particular to evaluate relationships, integration of data and comparison of results to name a few. Graphs are currently being used to analyse social networks, knowledge bases, biological networks and protein synthesis etc. (Rodriguez & Neubauer, 2010). A graph consists of a set of nodes and a set of edges that connect the nodes. The nodes are the entities of interest and the edges represent relationships between the entities. Edges can be assigned weights, directions and types. This is particularly useful in a clinical domain, the directions in edges help to represent causality between nodes, while the edges themselves can be annotated to represent the relationship between entities.

In order to ensure that the changes have been integrated consistently, source graphs need to be correctly replicated. This leads to the need to investigate and implement models that allow quick generation, integration and replication of graphs so that the source data can be quickly and effectively integrated. Furthermore, in order to replicate and integrate graphs, powerful graph models such as the Property Graph Model (Property Graph Model, 2016), Bayesian Networks (Nielsen & Jensen, 2009) or Markov Models (Nielsen & Jensen, 2009) are required. These graph models allow efficient inference of clinical data (Nielsen & Jensen, 2009) essential to determine relationships between disparate clinical data sources. Graph models can be divided into two classes: undirected and directed graph models. Markov Models (Nielsen & Jensen, 2009) are an example of undirected graph model, while Property Graph Model are an example of directed graph model. Bayesian Networks can accommodate a variety of knowledge sources and data types, they are computationally expensive and difficult to explore previously unknown network. Bayesian Networks do not have feedback loops due to the acyclic nature of Bayesian network graphs. In contrast to Bayesian Networks, Property graph model (Property Graph Model, 2016) represents data as a directed multi graph consisting of finite (and mutable) set of nodes and edges. Both, vertices and edges can have assigned properties (attributes) which can be understood as simple name-value pairs, shown in Fig. 3. A dedicated property can serve as a unique identifier for vertices and edges. In addition to this, a type property can be used to represent the semantic type of the respective vertex or edge. Properties of vertices and edges are not necessarily determined by the assigned type and can therefore vary between vertices or edges of the same type. Vertices can be connected via different edges as long as they have different types or identifiers. Property graph model (Property Graph Model, 2016) not only offers schema flexibility but also permits managing and processing data and metadata jointly. Graphs are generated by the graph engine based on the graph models.

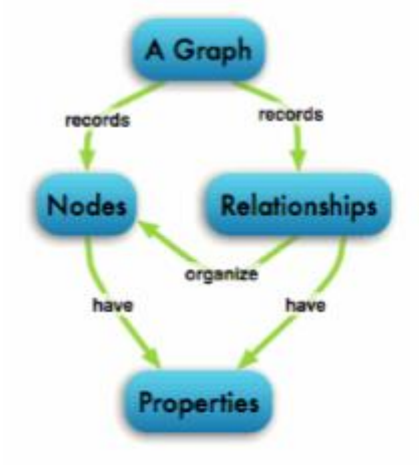

Fig. 3. Property Graph Model (Property Graph Model, 2016)

The property graph model provides the following key characteristics that differ from the classical relational data model:

- Relationships as first class citizens - With the property graph model relationships between entities are promoted as first class citizens of the model with unique identity, semantic type, and possibly additional attributes;
- Increased Schema Flexibility - In a property graph edges are specified at the instance and not at the class level, i.e. they relate two specific vertices, and vertices of the same semantic types can be related via different edges;
- No Strict separation between Data and Metadata - Vertices and edges in a graph can have assigned semantic types to indicate their intended meaning. These types can be naturally represented as a tree (taxonomy) or graph themselves. This allows their retrieval and processing as either type definitions, i.e. metadata or (possibly in combination with other vertices) as data.

In order to process large graphs such as those generated in clinical domain, there is a need for systems that can scale well over hundreds and thousands of nodes and edges at a single point in time. To ensure that this requirement can be achieved several large scale graph processing systems have been designed such as Apache Giraph (Giraph, 2016) , GraphLab (Low, et al., 2014)] and Pregel (Malewicz, et al., 2010). Apache Giraph is an iterative graph processing framework, built on top of Apache Hadoop. The input to a Giraph computation is a graph composed of vertices and directed edges. GraphLab is a graph based, high-performance, distributed framework written in C++. The GraphLab framework is a parallel programming abstraction targeted for sparse iterative graph algorithms. It provides a high level programming interface, allowing a rapid deployment of distributed machine learning algorithms. Pregel is Google’s scalable and fault tolerant API that is sufficiently flexible to express arbitrary graph algorithms. . Giraph is a suitable choice for applications where scalability is essential (Giraph, 2016), in contrast to that GraphLab is effective in applications where processing time is critical (Low, et al., 2014) In order for the system to scale well, these systems can be deployed over cloud to ensure the scalability of the system at large.

A proposed solution (Fig. 4.) is a graph based system that ensures coherent integration of data from heterogeneous clinical data sources for consistency and scalable analytics. In order to ensure consistency in the disparate clinical data sources and data warehouse graphs can be used based on the property graph model. In order to accommodate the overarching requirement of the amount of data large scale graph processing engines such as Giraph (Giraph, 2016) can be used since it is based on the property graph model. The proposed system can be designed based on the gather-apply-scatter (GAS) programming paradigm (Low, et al., 2014). This will allow an incremental graph problem to be reduced to a sub-problem that operates on a portion, or sub-graph, of the entire evolving graph. This sub-graph abstraction will aim for the solution to substantially out-perform the traditional static processing techniques. There are multiple heterogeneous clinical data sources with varying data (clinical trials data, genomics data, EHR data etc.). The proposed solution shall incorporate a metadata repository that ingests the metadata from the disparate clinical data sources in order to ensure the correctness of the data once it resides in the clinical data warehouse. The wrapper ingests the clinical source data and passes it on to the Graph Processing Engine that will generate graph and then allows it to push into the clinical data warehouse. If the source data changes/evolves e.g. over the course of the clinical trial, metadata repository detects the change and automatically alerts the data warehouse to update the graph in it, the changes are then made to the subset of the graph where the source has evolved so the overhead of generating new graph every time a changes occurs is omitted, reducing the computational workload on the graph engine.

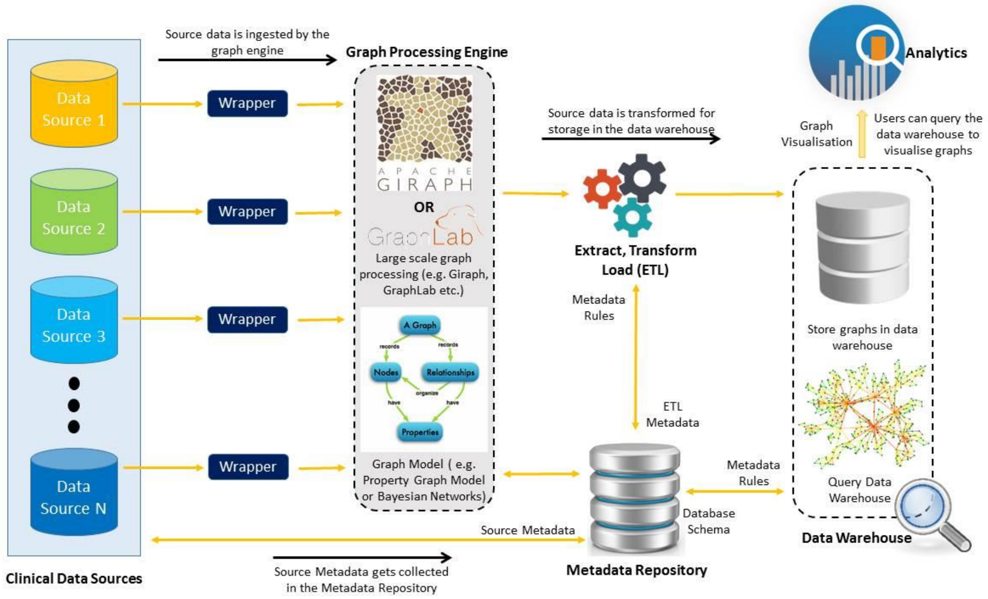
Source data is ingested by the graph engine
Graph Processing Engine
Data Source 1
Data Source 2
Data Source 3
Data Source N
Wrapper
Wrapper
Wrapper
Wrapper
APACHE GIRAPH
OR
GraphLab
Large scale graph processing (e.g. Giraph, GraphLab etc.)
A Graph
records
records
Nodes
Relationships
organize
have
have
Properties
Graph Model (e.g. Property Graph Model or Bayesian Networks)
Source data is transformed for storage in the data warehouse
Extract, Transform Load (ETL)
Metadata Rules
ETL Metadata
Analytics
Graph Visualisation
Users can query the data warehouse to visualise graphs
Store graphs in data warehouse
Query Data Warehouse
Data Warehouse
Metadata Rules
Database Schema
Source Metadata
Metadata Repository
Clinical Data Sources
Source Metadata gets collected in the Metadata Repository

Fig. 4. Proposed Solution Architecture to maintain data consistency in a Big Data Environment

Data coming from heterogeneous sources requires to be effectively integrated to ensure the coherence of the source data and the warehouse. Compared to traditional approaches for data integration, graphs promise significant benefits. First, a graph like representation provides a natural and intuitive format for the underlying data, which leads to simpler application designs. Second, graphs are a promising basis for data integration as they allow a flexible and uniform representation of data, metadata, instance objects and relationships. Graphs are well suited for data integration since they can model highly interconnected entities where other NoSQL alternatives and relational databases for short. Graphs can scale well over millions of nodes hence suitable for integration of data for clinical data. Metadata works as a governance framework in such an environment.

Data Infrastructure

Data integrated from diverse genomics and clinical sources requires a cloud based platform for storage and retrieval. We explain the infrastructure for data storage, retrieval and data movement on an on-demand basis.

When planning a multi-storage data warehouse environment the data needs to be understood and evaluated to determine whether a specific data set needs storing within a high performance legacy warehouse or on a commodity Hadoop cluster. A method to accomplish this is through assigning data with a “Data Temperature”.

“Hot” represents the in-demand and mission critical data in direct need for quick decision making, through to “Frozen” data which is accessed very infrequently and often is represented as archived. In between these two extremes are “Warm” data which is commonly used but does not have a huge amount of urgency, and “Cold” data which is infrequently accessed (Subramanyam, 2015).

The assigned temperature of data is used to determine its storage location. The frequently accessed “Hot” data is stored within fast storage such as high performance main-memory systems (scale-up) and the infrequently accessed “Cold” is stored on the large amount of cheap commodity storage such as Hadoop (scale-out) (Levandoski, Larson, & Stoica, 2013).

To make informed decisions about the data and where it should be moved, it is vital to identify what data is hot, and what is cold. Factors that are commonly used to establish data temperature are the frequency of access and age, so the more frequent the access and the more recent the data then the hotter the data ranked. These factors can be used separately or collectively.

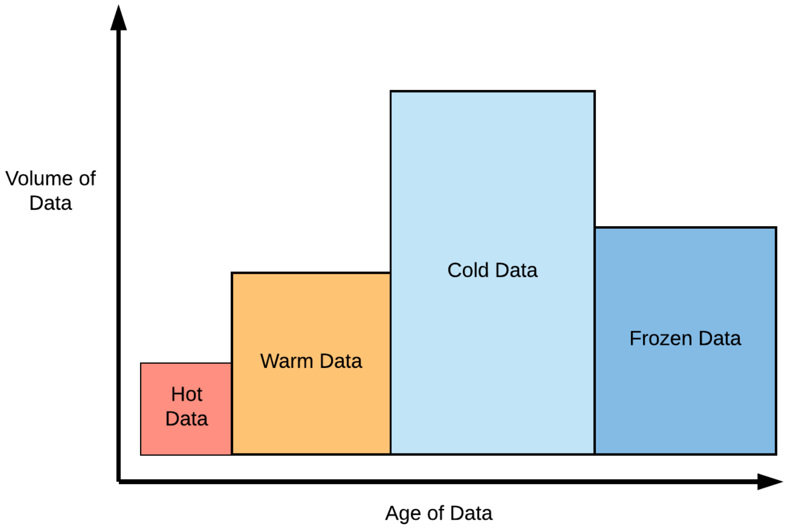

Fig. 5. Data Temperatures with age of data

In evaluating the data, certain workloads and data tasks may be identified that would be more suitable for batch-type work upon cold Hadoop storage. Usage and age are common factors for data temperature, but it is also important to consider data could have a priority based upon a specific task or alternatively based upon around a specific group of user requirements for the data, so it is important to consider business operations and other influencing factors when establishing a data temperature. Another example of this could be a set of data that remains unused for long periods of time but becomes incredibly important at a single point of the year the age and usage values would not be able to account for this but incorporating business logic or machine-learned knowledge would.

Read and write operations are expensive operations so where possible they are best avoided (Lin, Ma, Chandramohan, Geist, & Samatova, 2005), but with “Hot” storage being in short supply and high demand, it is inevitable that data will be moving in and out of this storage layer frequently. When planning to implement a multi-temperature storage environment, it is vital to plan how frequently and at what scale data will move. If it was based purely on the temperature, then you could potentially have data moving in and out of the hotter storage tiers constantly through the day which would be a considerable drain on resources and considerably impact system performance (Crago & Yeung, 2016).

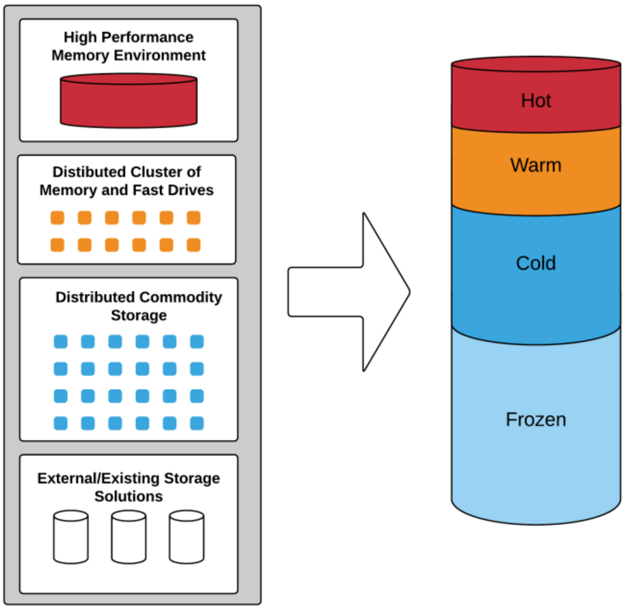
High Performance Memory Environment
Distibuted Cluster of Memory and Fast Drives
Distributed Commodity Storage
External/Existing Storage Solutions
Hot
Warm
Cold
Frozen

Fig. 6. Multi-tiered Data Storage

To prevent such a problem movement operations to rebalance the temperature need to be scheduled at opportune times but also need to be relatively frequent to ensure the benefits of a multi-temperature system are maintained and so that you are not moving huge amounts of data at one time.

Data Analysis

The main aim of data analytics is to provide quick healthcare. The available genomics data, and the new data that is being generated on almost a daily basis needs to be explored in a meaningful way. As a result, new insights, such as different relationships between disease and genome may be identified. Furthermore, this could be a significant step towards personalised medicine based on an individual's genome. This is a very difficult challenge given the size of genomics data. Add to it the integrated clinical data and the complexity of the problem increases several folds. There are many challenges along the way starting with finding an effective way of storage and retrieval of this huge amount of data. Once the data can be accessed quickly, insights could be found by generating useful data models.

The existing frameworks and platforms carry out genomic data analysis using SQL, NoSQL and high throughput approaches. For example, (Rohm & Blakeley, January 2009) looks at genome data-management by storing the data files and importing data into a relational database system for analysis using SQL. Another platform called Genome Analysis Toolkit integrates data access patterns with MapReduce to allow analysis (McKenna, et al., 2010). The HIG platform makes use of in-memory technology and distributed computing to increase the speed of processing by intelligent scheduling (Schapranow, October 2013). The SQL approaches are not appropriate for scalable analytics. NoSQL approaches are not optimized in reading data. MapReduce approaches are scalable but do not support iterative analytics. Most of the time, data integration as well as storage is not taken into account.

One way to address the scale of data and latency of accessing integrated genomic data is to introduce an in-memory Warehouse. The genomic data can be analysed on its own as well as in combination with clinical data. Genomics data can be pushed into the warehouse, but in order to store it efficiently, state of the art approaches such as tiling may be used (Guthrie, et al., 2015). The tiling approach breaks down the genomic data into short overlapping segments called "tiles", adds unique tags before and after each tile, along with a hash table of variants and its position in the genome. These tiles are then stored in a library. Gene Variants are stored as a new tile in the library at the same position in the genome as the reference genome. The genome is represented by a file containing pointers to the tiles in the library, thus reducing the size of the genome file from around 200GB to a few kilobytes (KB). Tiling could be integrated with the warehouse so that genomic data is efficiently stored in the warehouse in parallel to clinical data.

For analysis, the stored data should be quickly retrievable by addressing the computational cost associated with genome data browsing. Traditional methods for searching genome databases compare a sequence from a query to all the sequences (i.e. several GBs of data) present within the database being searched, with thousands of queries being processed a day. This method is, however, computationally expensive. With exabytes of genomic data, this creates a limitation to query and browse data quickly. To address this, approaches which will allow genomic data to be browsed within the least possible time should be explored. The current warehouse architecture is not scalable, but it can browse finite amount of data very quickly. Efficient memory and storage management models and innovative algorithms for processing large amounts of data should be investigated to offer high speed iterative analytics.

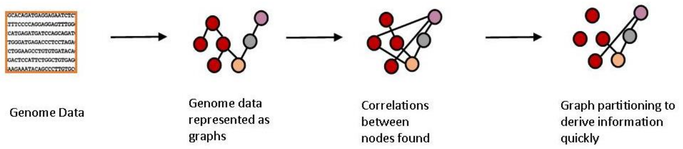
Genome Data
Genome data represented as graphs
Correlations between nodes found
Graph partitioning to derive information quickly

Fig. 7. Genomic data represented as graphs. Correlations are found between nodes and useful information is extracted using algorithms such as graph partitioning.

Analytics on genome data predict disease risks, drug efficacy and other outcomes. This requires integration of data from external sources. Several iterations of the data should sift through the data. To allow for fast and intelligent processing of data using the approaches such as machine learning, the stored genomic data could be represented as machine readable graphs (Fig. 7). Different graph models should be investigated and a suitable one, which could support high performance iterative analysis, should be selected. Previously, genome data has been represented as graphs (Ritchie, Holzinger, Li, Pendergrass, & Kim, February 2015). This could be extended to exploit the graph model for newer ways of processing genomic data that is structured into the tiling approach. Using a graph model will overcome the problem of processing the data iteratively because a graph-like representation will offer opportunities to rapidly generate and compute graphs using emerging hardware architectures and computing platforms.

The information associated with a genome and its variants will be linked within the graph model. Graphs will ensure that the genomics data they are representing is functionally correct and results being produced are consistent with stored data. Using a graph model will also ensure the correctness of the analytics being performed on the data because of their capabilities to be mathematically and statistically verified. Hundreds of associations between genes and variants could be deduced by linking the nodes in the graph model (Fig. 7). However, not all the correlations deduced within the data-sets would be of importance in different analytical studies of the genome. In order to extract the required information only, approaches and algorithms such as graph partitioning should be investigated (Fig. 7). This way a few meaningful correlations from hundreds of associations could be extracted using several iterations.

Hosting the warehouse in a cloud environment will provide the infrastructure for scalable analytics. As the warehouse is based on distributed, in-memory architecture hosted on a cloud environment, both performance and scalability will be addressed in the resulting infrastructure.

Conclusions

In this chapter, we presented a cloud based data analytics platform. It provides an infrastructure for integrating diverse sources of genomics and clinical data. The approaches for maintaining consistency of the integrated data are also explained. It is ensured that data is in consistent state before and after integration. Analytics approaches for generating insights from the integrated data are discussed towards end of the chapter.

References

(n.d.). (Illumina) Retrieved October 2016, from http://www.illumina.com/

(n.d.). (454 Life Sciences) Retrieved October 2016, from http://www.454.com/

(n.d.). (Complete Genomics) Retrieved October 2016, from http://www.completegenomics.com/

1000 Genomes Project Consortium. (2010). A map of human genome variation from population-scale sequencing. *Nature, 467*(7319), 1061 - 1073.

(2016, August). Retrieved from Akana: https://www.akana.com/products/semantics-manager

(2016, 09 01). Retrieved from Property Graph Model: https://github.com/tinkerpop/blueprints/wiki/Property-Graph-Model

(2016, September). Retrieved from Giraph: http://giraph.apache.org/

Akavia, U. D., Litvin, O., Kim, J., Sanchez-Garcia, F., Kotliar, D., Causton, H. C., . . . Pe'er, D. (2010). An integrated approach to uncover drivers of cancer. *Cell*, 1005 - 1017.

Apache Hadoop Goes Realtime at Facebook. (n.d.). *Facebook.*

Borthakur, D., Muthukkaruppan, K., Ranganathan, K., Rash, S., Sarma, J. S., Spiegelberg, N., . . . Aiyer, A. (2011). Apache Hadoop Goes Realtime at Facebook. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (pp. 1071-1080). Athen, Greece: ACM.

Brierly, C. (2010, Jun). *Press release for UK10K.* Retrieved from http://www.wellcome.ac.uk/News/Media-office/Press-releases/2010/WTX060061.htm

Crago, S. P., & Yeung, D. (2016). Reducing data movement with approximate computing techniques. *2016 IEEE International Conference on Rebooting Computing (ICRC)* (pp. 1 - 4). IEEE.

Edifecs CDI. (n.d.). Retrieved from https://www.edifecs.com/downloads/Clinical_Data_Integration_Solution_Brief_2015.pdf

Fridley, B. L., Lund, S., Genkins, G. D., & Wang, L. (2012). A Bayesian integrative genomic model for pathway analysis of complex traits. *Genetic epidemiology*, 352 - 359.

Guthrie, S., Connelly, A., Amstutz, P., Berrey, A. F., Cesar, N., Chen, J., . . . Zaranek, A. W. (2015). Tiling the genome into consistently named subsequences enables precision medicine and machine learning with millions of complex individual data-sets. *PeerJ Preprints*, 3:e1780. doi: https://doi.org/10.7287/peerj.preprints.1426v1

Hamid, J. S., Hu, P., Roslin, N. M., Ling, V., Greenwood, C. M., & Beyene, J. (2009). Data integration in genetics and genomics: methods and challenges. *Human Genomics and Proteomics*.

Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., & Conde, J. G. (2009). Research electronic data capture (REDCap) - a metadata-driven methodology and workflow process for providing translational reserach informatics support. *Journal of biomedical informatics*, 377 - 381.

Holzinger, E. R., & Ritchie, M. D. (2012 January). Integrating heterogeneous high-throughput data for meta-dimensional pharmacogenomics and disease-related studies. *Pharmacogenomics, 13*((2)), 213–222. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3350322/pdf/nihms357046.pdf

Karasawas, K., Baldock, R., & Burger, A. (2004). Bioinformatics integration and agent technology. *J Biomed Inform*, 205 - 219.

Lapatas, V., Stefanidakis, M., Jimenez, R. C., Via, A., & Schneider, M. V. (2015). Data Integration in Biological Research - an overview. *Journal of Biological Research - Thessaloniki*, 1 - 16.

Lee, E., Cho, S., Kim, K., & Park, T. (2009). An integrated approach to infer causal associations among gene expression, genotype variation, and disease. *Genomics*, 269 - 277.

Levandoski, J. J., Larson, P.-A., & Stoica, R. (2013). Identifying Hot and Cold Data in Main-Memory Databases. *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)* (pp. 26 -27). Washington, DC, USA: IEEE Computer Society.

Lin, H., Ma, X., Chandramohan, P., Geist, A., & Samatova, N. (2005). Efficient Data Access for Parallel BLAST. *19th IEEE International Parallel and Distributed Processing Symposium* (pp. 72 - 82). IEEE.

Louie, B., Mork, P., Martin-Sanchez, F., Halevy, A., & TarczyHornoch, P. (2005). Data integration and genomic medicine. *Journal of biomedical informatics*, 5 - 16.

Low, Y., Gonzalez, J. E., Kyrola, A., Bickson, D., Guestrin, C. E., & Hellerstein, J. (2014). Graphlab: A new framework for parallel machine learning. *arXiv preprint arXiv: 1408.2041.*

Lumeris CDI. (n.d.). Retrieved from http://lumeris.com/wp-content/uploads/2014/05/Lumeris-SOL.CDI_.05-14.v1.pdf

Malewicz, G., Austern, M. H., Bik, A. J., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010). Pregel: a system for large-scale graph processing. *In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data* (pp. 135 - 146). ACM.

Maxam, A. M., & Gilbert, W. (1977). A new method for sequencing DNA. *Proc Natl Acad Sci USA, 74*(2), 560 - 564.

McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., . . . DePristo, M. A. (2010). The Genome Analysis Toolkit: A MapReduce framework for analiyzing next-generation DNA sequencing data. *Genome Research*, 1297-1303.

Metzker, M. L. (2010). Sequencing technologies - the next generation. *Nature Reviews Genetics, 11*, 31 - 46.

National Human Genome Research Institute. (2016, July). *National Human Genome Research Institute.* Retrieved from https://www.genome.gov/27565109/the-cost-of-sequencing-a-human-genome/

Nevins, J. R., Huang, E. S., Dressman, H., Pittman, J., Huang, A. T., & West, M. (2003). Towards integrated clinico-genomic models for personalized medicine: combining gene expression signatures and clinical factors in breast cancer outcomes prediction. *Human Molecular Genetics*, R153-R157.

Nielsen, T. D., & Jensen, F. V. (2009). Bayesian networks and decision graphs. *Springer Science & Business Media*.

Park, Y., Shankar, M., Park, B. H., & Ghosh, J. (2014). Graph databases for large-scale healthcare systems: A framework for efficient data management and data services. *In Data Engineering Workshops (ICDEW)* (pp. 12 - 19). IEEE.

Ritchie, M. D., Holzinger, E. R., Li, R., Pendergrass, S. A., & Kim, D. (February 2015). Methods of integrating data to uncover genotype-phenotype interactions. *Genetics, 16*, 85-97.

Rodriguez, M. A., & Neubauer, P. (2010). Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 35 - 41.

Rohm, U., & Blakeley, J. A. (January 2009). Data Management for High-Throughput Genomics. *Conference on Innovative Data Systems*.

Salem, A., & Ben-Abdallah, H. (2015). The design of valid multidimensional star schemas assisted by repair solutions. *Vietnam Journal of Computer Science*, 169 - 179.

Sanger, F., & Coulson, A. R. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J Mol Biol., 94*(3), 441 - 448.

SAS CDI. (n.d.). Retrieved from [16] B. Louie, P. Mork, F. Martin-Sanchez, A. Halevy and P. TarczyHornoch, “Data integration and genomic medicine,” Journal of biomedical informatics, pp. 5 - 16, 2005.

Schapranow, M. (October 2013). HIG - An In-memory Database Platform Enabling Real-time Analyses of Genome Data. *IEEE International Conference on Big Data*, 691 - 696. doi:10.1109/BigData.2013.6691638

Songting, C. (2010). Cheetah: A High Performance, Custom Data Warehouse on Top of MapReduce. *Proc. VLDB Endow*, 1459 - 1468.

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., . . . Robinson, G. E. (2015). Big Data: Astronomical or Genomical? *PLOS Biology*.

Subramanyam, R. (2015). HDFS Heterogeneous Storage Resource Management Based on Data Temperature. *2015 International Conference on Cloud and Autonomic Computing* (pp. 232-235). ICCAC.

Sujasnsky, W. (2001). Heterogeneous database integration in biomedicine. *J Biomed Inform*, 285 - 298.

Wang, L., Zhang, A., & Ramanathan, M. (2005). BioStar models of clinical and genomic data for biomedical data warehouse design. *International journal of bioinformatics research and applications*, 63 - 80.