Concepts in Information and Knowledge Management for Translational Research

A compilation of practices applied and developed by the eTRIKS consortium

Bergeron, Emam, Fitch, Guitton, Guo, Henderson, Oehmichen, Richard, Saqi, Wagers and Yang

Edited by Jay Bergeron

The *European Translational Research Information and Knowledge management Services* (eTRIKS), an *Innovative Medicines Initiative* (IMI) public private partnership, operated from October 2012 to September 2018 to provide technical products and services to support investigators engaged in elucidating and applying molecular and digital biomarkers associated with complex diseases.

The content of this publication has received support from the Innovative Medicines Initiative Joint Undertaking under grant agreement number 115446 (eTRIKS), resources of which are composed of financial contributions from the *European Union's Seventh Framework Programme* (FP7/2007–2013) and *The European Federation of Pharmaceutical Industries and Associations* (EFPIA) companies' in-kind contributions (www.imi.europa.eu).

Notifications

Consistent with accepted practice for academic publication, the authors have endeavored to thoroughly cite any non-original work or idea. Use of non-original content and images are the sole responsibility of chapter authors who have attested to the appropriate/permitted use of any copyrighted text or image within their specific chapter(s). Neither the editors nor the publisher have reviewed reference sources or confirmed provenance of images or source text. Any concerns should be addressed to the authors, directly or through the publisher.

The content of this book represents the work and opinions of the authors. Knowledge and best practices within the discipline of medical and research informatics and analytics are constantly changing. Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information and methods described herein. While all reasonable efforts have been made to present information that is accurate and reliable at the time of publication the authors, editors, publishers, contributors, funders and copyright holder accept no legal liability for any injury or damage to persons, projects or property from any use or operation of any methods, information. products, instruction, guidance or ideas contained in the material herein.

1st Edition

Edited by Jay Bergeron, Ninja Hoen, Aleksandra Draper and Kyle Bergeron

Biosci Consulting
Scott Wagers, MD. CEO/Founder
Maasmechelen Belgium
+32 89 25 4009
https://www.biosciconsulting.com/

Acknowledgements

This book is a deliverable of the eTRIKS consortium with the initial draft provided and accepted by the IMI at the submission of the consortium's final report in 2019. This first edition was released to the IMI for public use in 2020. The authors express, first and foremost, sincere appreciation to the scientists and technologists across seventeen organizations who worked tirelessly over the course of a decade to propose, champion and realize the goals and achievements of the eTRIKS consortium. Equally important is the recognition of, and gratitude for, the customers and collaborators who considered, adopted and extended eTRIKS products and services during, and following, the term of the collaboration.

The following individuals deserve specific recognition and sincere thanks.

Ninja Hoen and Aleksandra Draper of Biosci Consulting and Kyle Bergeron (Tufts University baccalaureate student) for diligent and careful editing and formatting.

Dr. Ian Dix of AstraZeneca who first conceived of eTRIKS in 2008 and co-led the collaboration through its initial year.

Colm Carroll and Ann Martin, each serving as eTRIKS' IMI Liaisons, for guiding the project through the intricacies of IMI-1 processes/policies and ensuring that the public funds entrusted to eTRIKS were directed solely and productively towards the mission of improving public health through accelerating the development of new medicines.

On behalf of the authors,

Jay Bergeron
eTRIKS Scientific Coordinator and Lead Editor
Pfizer Digital, Pfizer Inc. Cambridge, Massachusetts USA

Yike Guo
eTRIKS Academic Co-Leader
Data Science Institute, Imperial College London, London, UK

December-2020

Table of Contents

Introduction: eTRIKS and Translational Research Information Sciences

Jay Bergeron

eTRIKS and the drive for sustainable translational research information management practices

Translational Research (TR) provides novel insights into disease progression and classification, biomarker discovery and patient stratification through the collection and analysis of traditional clinical observations coupled with corresponding large scale molecular and digital biomarkers. The discipline seeks to reduce the attrition of investigational new drugs during clinical development and accelerate the timelines associated with clinical programs. TR projects depend heavily upon Knowledge Management (KM) capabilities and services that provide study data to project investigators for exploratory analysis.

The *European Translation Research Information and Knowledge management Services* (eTRIKS) was an *Innovative Medicines Initiative* (IMI) consortium that operated between 2012 and 2018 comprised of ten Pharmaceutical companies, four academic institutions, the *Clinical Data Interchange Standards Consortium* (CDISC), *IDBS* (a leading scientific software company) and *Biosci Consulting* (specialists in developing and managing biomedical consortia). eTRIKS was launched to establish information platforms and services to promote data and process harmonization across TR programs operating within the IMI and other *Public Private Partnership* (PPP) frameworks. eTRIKS sought to reduce the operating costs and accelerate information system implementation for TR projects. Additionally, the collaboration sought to maximize the use and value of the research data generated by these projects through harmonized data standards and processes, scientific data analytics, data reuse policies and best practice consulting.

The eTRIKS consortium delivered a core TR KM software platform, TR analytics applications and a wide variety of value-added best practices that impacted over sixty client projects throughout the course of the collaboration. The consortium's assets are available, by and large, under open licensing. The application of eTRIKS best practices continues through the work of the eTRIKS commercial spinoff *Information Technology for Translational Medicine* (ITTM), the *eTRIKS Data Sciences Network* (eDSN) and the many adopters of eTRIKS' products and services.

The information, recommendations and guidelines presented in this book are the direct result of six years of intense efforts by over one hundred individuals affiliated with the eTRIKS consortium. The eTRIKS deliverables, including software development and integration,

analytic method advancement and implementation, data standards consolidation and application, contract management and legal and ethical discourse constitute a comprehensive set of products, services and best practices to expedite TR endeavors while limiting liability and uncertainty with respect to handling the information gathered from study participants.

eTRIKS has created or extended many products and services that have been applied to client translational research programs. The following section describes the major deliverables.

eTRIKS Translational Research Information Platform

The eTRIKS translational research information platform is based on the open source *tranSMART* translational research data warehouse created by *Johnson and Johnson (J&J)* and released open source under the *GNU Public License version 3* in 2012. eTRIKS released five major platform versions throughout the course of the collaboration. The eTRIKS platform incorporates an open source database system which greatly promoted the distribution of the platform within academic and non-profit institutions. Many open license analytical applications such as *Galaxy* for supporting bioinformatic methods and *XNAT* for bioimage management were made interoperable with the eTRIKS environment. The platform was extended substantially through custom software development to enhance the visual and high-performance analytical capabilities of the tranSMART system. The final version of the platform introduced complicated cross study analytic and longitudinal data support capabilities as well as a new user interface to exploit these advanced features.

eTRIKS Public Platform: eTRIKS deployed and hosted a publicly-accessible eTRIKS Translational Research Information Platform (available at https://public.etriks.org/transmart/datasetExplorer at the time of this writing) that exposes roughly 200 clinical studies curated to eTRIKS' standards across a wide breadth of disease areas. Additionally, the Public Platform serves as a demonstration and training environment for investigators interested in evaluating the eTRIKS platform.

eTRIKS Labs

The eTRIKS Labs (https://www.etriks.org/etriks_labs/) concept arose out of a desire to centrally brand and distribute eTRIKS' custom software applications and analytical methods. These applications and methods were developed and deployed to either directly extend the features of the eTRIKS platform with or to complement the platform with cooperative abilities. All eTRIKS Labs are provided open license to the research community. The eTRIKS Labs are comprised of the following assets:

- **eTRIKS Analysis Environment (eAE):** A high-performance compute scheduler that enables investigators, and their applications, to launch analytical jobs against associated compute clusters. Jobs can be launched using integrated *Jupyter Notebooks*. The eAE software deployment is decoupled from specific high-performance compute cluster implementations. As such, the eAE can be installed and operated, in principle, on any

cluster configuration.

- **eTRIKS Harmonization Service (eHS):** A system that facilitates the challenging manual data transformation and mapping process employed to populate the eTRIKS platform with study data. The highly variable nature of data collected during clinical studies complicates its incorporation into structured data warehouses such as the eTRIKS platform. The eHS provides a user interface to accelerate the configuration of clinical data collections and provides certain automated mapping capabilities. The eHS transformations are based on the industry standard ***Clinical Data Interchange Consortium Standards*** (**CDISC,** https://www.cdisc.org/ **)**.

- **Hi Dome:** An application that allows eTRIKS platform users to select cohorts using values of high dimensional datasets, such as gene expression, and to determine statistically significant differences between the cohorts based on these high dimensional results (e.g. as significant differences in the expression of one or more genes between the cohorts). Hi Dome is a natural high dimensional extension to tranSMART's baseline clinical data analysis capabilities.

- **Disease Knowledge Base:** A semantic query application for molecular pathways created using the open source *Neo4J* (https://neo4j.com/) graph database engine leveraging the natural fit of semantic/graph databases to better support the data structure of molecular networks. Molecular pathways structured as triple store relations can be searched using Neo4J's *Cypher* query language and presented visually with the corresponding information associated with each molecular entity that participates in the network.

- **Disease Maps:** eTRIKS extended disease pathway maps related to Asthma and Parkinson's Disease working directly with information known a priori and data generated by client projects. Additionally, supplemental tools were created to accelerate the modeling of these disease maps from underlying disease-associated data.

- **Similarity Network Fusion (SNF):** An ***R-Shiny*** (https://shiny.rstudio.com/) application that was developed to provide an operational user interface for this novel computational method for genomic data integration (developed by Wang et al., in the lab of Anna Goldenberg (http://compbio.cs.toronto.edu/SNF/SNF/Software.html)). SNF constructs patient similarity networks based on a diversity of associated data types and, in a second step, iteratively integrates the individual patient networks until the algorithm converges to a final fused network representing the population.

- **Weighted Gene Co-Expression Network Analysis (WGCNA):** An R-Shiny application was developed to provide an operational interface for performing correlation network gene clustering analysis using the method implemented and

published by Langfelder and Horvath (https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/).

- **Patient Input Platform:** eTRIKS created a discussion game framework, based on the open license *Play Decide* platform (https://playdecide.eu/), to assist patients and legislators in navigating the risks and benefits of consenting their individual health data, or the data of their constituents, to promote biomedical research. A series of game cards were created that present questions aimed at spurring open discussion with respect to the topic of medical data reuse. With the help of a facilitator, groups of people work together to formulate/reassess opinions regarding the risks and potential benefits of health data reuse. Applied in multiple sessions with patients, legislators and medical professionals the favorability of medical data sharing was consistently raised among these key stakeholders as a direct result of these sessions.

eTRIKS Standards Starter Pack

The selection and application of consistent data standards is critical for enabling high quality data review and analysis. Moreover, consistent data standards facilitate meta analyses across studies and increase opportunities for data reuse. The Standards Starter Pack documents the best practices for optimizing the quality and usability of exploratory medical data loaded to the eTRIKS platform. Tailored for project leaders and data managers, the resource provides a comprehensive review of pertinent biomedical data standards including guidance as to which standards platforms are best suited for specific research plans. The Standards Starter Pack documents were made available for all IMI projects to promote consistency in data handling and to raise awareness of the potential advantages of applying consistent standards across translational research projects. The Starter Pack was the basis for eTRIKS project consulting with respect to standards implementation. Multiple extended versions of the Standards Starter Pack were released to the public domain.

eTRIKS Code of Practice on Secondary Reuse of Medical Research Data

The Code of Practice provided multi-partner, multinational scientific research projects with urgently needed practical guidance for conforming to applicable data protection laws, particularly the European Data Protection Directive which was in force at the time the code of practice was developed and initially released in 2014. The Code of Practice was adopted by the IMI, for all new projects, as the base level guideline for the design of ethical practices and policies regarding the appropriate use of patient data. The relevance of the Code of Practice was lessened once the *General Data Protection Regulation* (GDPR) became law in May of 2018. The Code of Practice was the basis of eTRIKS consulting with regards to ethical data use. eTRIKS team members consulting on legal and ethical considerations also became highly knowledgeable with respect to the GDPR statutes to assist clients with necessary process changes once the GDPR became law. The *BBMRI-ERIC GDPR Code of Conduct* (http://code-of-conduct-for-health-research.eu/), has replaced the eTRIKS Code of Practice.

Data Catalogue

eTRIKS created the first broadly applicable *Data Catalogue* (https://datacatalog.elixir-luxembourg.org/) for datasets associated with IMI projects as well as other published sources. The Data Catalogue provides a searchable metadata repository that encompasses a wealth of cross-project study information allowing investigators to quickly find and assess datasets pertinent for their research endeavors. The Data Catalogue is a web-based product implemented using the Open Source *CKAN* (https://ckan.org/) data portal software and affords end users the opportunity to interactively search and display study metadata and summary information across the managed study collection.

Materials Transfer Agreement/Confidential Disclosure Agreement Templates

Material Transfer Agreements (MTAs) are contracts that define policies and responsibilities regarding oversight of the exchange and use of *intellectual property* (IP) between two or more parties. MTAs were generally necessary for eTRIKS to provide comprehensive services to client projects (research data being the pertinent IP for eTRIKS engagements). The MTAs between eTRIKS and the IMI projects *ABI-Risk* and *Oncotrack* each required approximately two years of negotiations to close as all individual partners across the contracting consortia were required to authorize the MTAs as signatories (The ABI-Risk consortium alone required over 40 parties to negotiate MTA terms). The experience of contracting across these large public private partnerships was codified into an MTA template containing the basic collection of terms and clauses pertinent to materials transfer. The template should accelerate the negotiations and subsequent execution of future MTAs. A similar template was created for *Confidential Disclosure Agreements* (CDAs) which were pertinent to eTRIKS project engagements. CDAs for IMI projects can be brokered via the project coordinators (serving as the signatory on behalf of all consortium participants), thus greatly reducing the time and effort to close a CDA relative to an MTA. Nevertheless, the availability of the CDA template should further ease the time and costs associated with commissioning multi-project engagements.

eTRIKS Training Materials

eTRIKS personnel provided many training sessions throughout the course of the collaboration as part of WP6 outreach and promotion efforts. Topics codified into training programs and materials include:

- Platform installation and support
- Introductory guide for new platform users
- In depth training for advanced platform users
- eTRIKS reporting (defect management and support services)
- Building new interfaces with the tranSMART API
- Data Privacy and Reuse
- Application of Data Standards
- Introductory Data Curation and Database Mapping
- Advanced Data Curation and Database Mapping

- Electronic Case Report Form design

The training materials that were developed for these sessions are not distributed under open license. Rather, these materials are distributed through agreement with BioSci Consulting to support commercial services that enable the use of eTRIKS assets.

eTRIKS Website (https://www.etriks.org/**)**
eTRIKS.org provides information pertinent to the consortia as well as access to the eTRIKS assets that are distributed under open license. All assets discussed in this section are available through this website as of the time of this writing.

Enhancing the value of data for medical research

Developing application infrastructure and corresponding best practices to the magnitude accomplished by eTRIKS required a highly focused effort by a large group of diversely skilled individuals. However, the preferences of prospective clients were, of course, critically important. Academic clients, either acting as a single project team or within the context of a public private partnership, were the key customer groups targeted by eTRIKS. The willingness of academic customers to partner with eTRIKS and make use of the open license applications that eTRIKS produced resulted in the large portfolio of community products and services outlined above. These assets are the inspiration for this book, and many will be detailed within the subsequent chapters. The authors hope that readers, especially those clinicians, analysts and technologists who assemble to prosecute translational research programs, will find the content presented to be informative for the design, implementation and execution of their studies and, ultimately, the use of their data to realize medical breakthroughs for patients worldwide.

Readers should note this book has not been directly peer-reviewed. However, much of the content presented herein is published elsewhere within peer-reviewed journal articles. The content of chapters four, six, seven and nine were published prior as part of successful dissertation and Capstone submissions and have, thus, been scrutinized by expert faculty. The content of chapter eight is reprinted from peer-reviewed articles pursuant to terms of licensing for the convenience to the reader.

Chapter 1: Clinicians and Data Science

Yike Guo, Scott Wagers and Mansoor Saqi

1.1 Why should clinicians study data science?

Medicine is increasingly becoming data-centric. Large scale patient datasets, including genetic and molecular profiles capable of assessing hundreds of thousands of markers, were once solely the purview of biomedical researchers. These investigators seek, through carefully designed clinical studies, to understand and, ideally, interrupt the mechanisms and progression of specific human diseases. However, technological developments that have substantially decreased the costs of individual molecular profiling, coupled with the medical knowledge realized from the use of these technologies in disease research, has led to the application of molecular insights for diagnosis and medical intervention with respect to individual patients. Moreover, the advent of digital biomarkers such as those derived from medical images and wearable accelerometers will provide further opportunities to collect large scale medical datasets from patients and use these data to better individual health. Of course, the collection of these data does not, in and of itself, benefit patients. Digital and molecular biomarker analyses depend upon sophisticated mathematical processing methods applied in concert with robust computational environments on which these methods operate. Although it is not reasonable to expect that every practicing physician to also be an expert computational scientist and bioinformatician, clinicians will need to increasingly consider and incorporate digital and molecular biomarker test results into the medical assessments and treatment plans of their patients. A familiarity with pertinent data science methods will promote the effective integration of large-scale biomarker results with traditional medical assessments regardless of whether the biomarker results are delivered to the physician directly via software or through consultation with medical informaticians.

In a paper in the NEJM[1], Obermeyer and Lee argue that medicine will need to embrace these developments and clinicians trained in statistics and computer science will have important roles. Physicians will need to appreciate and leverage the following elements of individual big data and information dissemination. By doing so, physicians will be better able to act as stewards of their patient's healthcare and to demystify the confusion that many patients may experience regarding when, and how, such data should impact medical intervention for themselves or their loved ones. Certain factors are pertinent to this work:

- Availability of large-scale databases of molecular profiles and electronic health records, whether provided anonymized to the public domain or restricted to select users through commercial, or other, arrangements

- Complex machine learning methods for analyzing large collections of patient data and using these learnings to inform individual patient diagnosis and treatment
- Patient communities formed to share experiences and information with respect to shared medical conditions
- Patient-directed data collection and interpretation (not ordered by a physician) such as personal genome sequencing
- Internet-based and direct-to-consumer availability/marketing of health information and therapies which may both empower, as well as inappropriately bias, patients

These factors are changing the dialogue between doctors and patients[2] while biomedical data scientists continue to leverage these factors to advance new therapeutics and healthcare protocols. This dynamic broadly impacts healthcare stakeholders including patients, providers, biomedical researchers and funders.

1.2 Towards a new taxonomy

Many medicines sometimes fail to alleviate the medical conditions and associated symptoms experienced by the patients for whom these medicines are prescribed.[3] A recent study of the top ten highest grossing medicines in the US suggests that only between 4% and 25% of patients benefited from drugs[3] that they were prescribed. Complex diseases, such as cancer and inflammatory syndromes, that give rise to similar physical symptoms across patients are often caused by underlying molecular mechanisms that are distinct to individual patients. Thus, medications that are highly effective for certain patients may have limited, or no, efficacy for other patients due to mechanistic heterogeneity.

Consider, for example, Asthma, a chronic lung disease characterized by obstruction of the aveoli due to inflammation that can lead to permanent tissue remodeling. Asthma causes substantial suffering for those afflicted and carries a significant societal disease burden. The disease is usually managed by administration of corticosteroids; yet, some patients do not respond well to this treatment, even at high doses. Asthma is described as a heterogeneous disease as patients diagnosed with the disease can present varied clinical symptoms (a.k.a phenotypes) such as the presence of remodeled aveolar pathways. Additionally, patients can present characteristic molecular profiles, such as differences in the expression of gene sets in tissues pertinent to the disease. Such molecular phenotypes are more likely to definitively describe the ***subtype*** of the disease responsible for the patient's pathology and will more accurately inform a treatment plan tailored for the patient's specific circumstance. The identification of molecular subtypes can also reveal commonalities between diseases not apparent from clinical symptoms alone. This stratification will lead to a new taxonomy of disease.

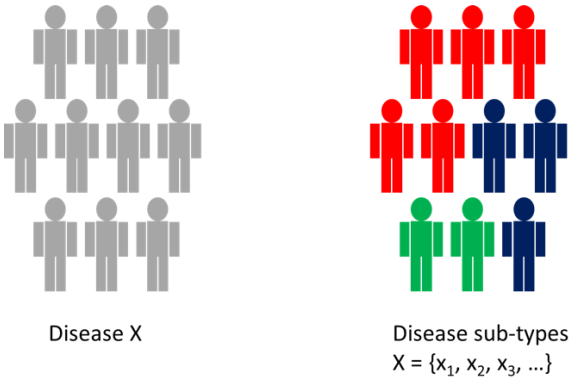


Figure 1.1: Stratification

If mechanistic information can be associated with disease subtypes, novel therapeutics can be designed to specifically modulate the molecular mechanisms that give rise to the subtype. Rather than using a generalized therapeutic approach to a complex disease, patients can be matched to treatments that are known to be efficacious against their diagnosed molecular disease subtype. This individual approach to treatment can lead to higher success rates while minimizing trial and error with respect to medical intervention.

1.3 Precision Medicine

Precision medicine is the practice of developing therapeutics specialized for the treatment of specific disease subtypes and the prescribing of these therapeutics to *only* those patients that exhibit, as demonstrated by robust diagnostic tests, the disease subtype corresponding to the pertinent therapeutic agent. Precision medicine promises to provide "*the right drug for the right patient at the right time at the right dose*". Major precision medicine initiatives aiming to collect genetic data from large numbers of individuals have been launched in several nations[4], including the United States (one million individuals) and the United Kingdom (100,000 individuals) with a focus on better understanding cancer and rare diseases.

Technological developments that have substantially increased the throughput of molecular data acquisition have led to exponential decreases in per subject costs of molecular profiling. The ability to generate massive molecular profile collections matched with corresponding advances in biology, computational methodologies and increases in computing power are transforming the approach to disease research and will most assuredly transform the future practice of medicine[4]. The convergence of these developments in technology, biology, and computing has driven a tremendous amount of activity in ***personalized medicine*** (or P4 medicine as it is also known as, reflecting the four key components, namely *Personalized*, *Predictive*, *Preventive* and *Participatory*)[5]. As large-scale molecular data collection becomes cheaper, the challenge in using these data will shift from data generation (e.g. sequencing) to data analytics[6]. Indeed, several aspects of data sciences including data storage, standardization, integration,

provenance, mining, and analytics are emerging as fundamental challenges in realizing the promise of precision and personalized medicine.

1.4 Historic references

The relevance and importance of data to medicine is, of course, not new. Florence Nightingale in 1869 was aware of the importance of collecting and analyzing, statistically and visually, epidemiological data to assess, justify and promote changes in hygiene and medical practice to benefit public health. She also understood the importance of the graphical presentation of data[7].

John Snow, a physician practicing during the same period as Florence Nightingale, used data analysis to suggest the source of an outbreak of cholera in London. He mapped places where deaths had occurred and identified clusters of deaths, one of which was close to a water pump. His work demonstrated the importance of using data analytics and visualization in responding to public health crises. From such foundational data driven insights, modern medical research has grown to rely on a wide breadth of sophisticated analytic methods applied by highly trained information and analytic discipline specialists.

1.5 Opportunities and challenges: U-BIOPRED as a case example

Translational medicine studies routinely collect multiple types of data from patient cohorts. Traditional ***low dimensional*** clinical descriptors rely on a single, or small collection, of data values to describe a clinical measurement. Low dimensional values include demographic attributes, laboratory measurements and procedural endpoints. ***High dimensional*** (or *Omics*) data, with hundreds, to hundreds of thousands, of data values are also routinely captured from high throughput instrument platforms. These data represent profiles of molecular biomarkers, including genotypes, transcriptomics, metabolomics, proteomics and other profiles.

A recent, large, multi-partner study on severe asthma by U-BIOPRED (Unbiased BIOmarkers in PREDiction of respiratory disease outcomes) illustrates the challenges of data management and data analytics[8]. The U-BIOPRED consortium collected transcriptomics, lipidomics, breathomics (exhaled gases) and other molecular profiles. These molecular features were, and continue to be, used to stratify asthma into disease subtypes by employing data integration approaches and unsupervised learning. A patient's health status at a given time is characterized by the combination of various high and low dimensional data values collected from the patient during the study. Data is collected longitudinally (i.e. at multiple time points during the study) such that changes in a patient's health status as the study progresses can be, ideally, correlated with changes in one or more molecular profiles[9]. Correlations identified in this manner become hypotheses that can subsequently be rigorously studied with the purpose of stratifying disease subtypes through the discovery of causal relationships between endogenous physiological changes and the wellbeing of individual patients. The specialized field of science that applies human clinical and molecular profiles to explore the basis of disease is generally termed ***translational research***. In depth understanding of human pathology promotes the selection of

better therapeutic targets preclinically that increases the probability that promising preclinical therapies will demonstrate efficacy (or *translate*) in phase-2 clinical trials. The similar term, ***translational medicine***, refers to the introduction of promising therapeutics into successful medical practice. However, translational research and medicine are often used interchangeably in scientific discussion and literature.

The U-BIOPRED collaboration surfaced several key data-related issues that impede the progress of interpreting translation research outcomes. A major impediment is the difficulty in standardizing and integrating disparate types associated with translational studies such that these data can be readily used for mathematical or visual analysis. Typically, analysts will need to transform datasets into specialized data structures and formats, an often tedious and time-consuming activity, prior to applying a specific analysis method. Bench biologists and clinicians may not be trained in the data sciences skills necessary to prepare data for analysis. Instead, direct support from data scientists and/or sophisticated custom software packages are often necessary to complete exploratory analysis plans.

A diverse set of analytical methods can be applied to translational datasets once these datasets are prepared for use. Analysis can be performed by computational specialists or through software designed for use by clinicians and scientists not specifically trained in the art of computational method development and operation.

However, clinicians and clinical scientists are crucial to translational data management and analysis. These roles ensure that data fields are properly defined and inter-related, data values are consistent in their format and the meaning of data values are sensible with respect to the scientific observation or description that these data describe. Clinical scientists must collaborate with dedicated technologists and analysts to expedite data processing and assure the quality the study's data assets through participation in planning the data strategy for the study and testing of the data processing methods that are developed.

This book serves as an introduction to data management and analysis concepts for clinical scientists who are not also information technologists. It is hoped that a basic understanding of the information management concepts and processes pertinent to clinical studies will promote strong collaboration between clinical scientists and their technology partners leading to confident and high value interpretations of study data.

It is imperative that a data plan be established for a translational study. Data processing and structure will be critical for the efficient use of the collected data. Moreover, properly structured data, including allowance for, and representation of, incomplete data will ensure that intended research questions can be productively addressed. Adherence to established data standards will have a direct impact on the value of the dataset. One aspect of value being ease of use within the context of the study for which the data was collected. A second aspect of value being the dataset's potential to contribute to research proposals beyond the study for

which the data were collected. This second aspect is, of course, a critical strategy to maximize the study investment and could provide the study investigators, and others, substantial benefit with respect to their pursuit of research interests well beyond the completion of the study itself. The motivation to move quickly from protocol design and approval to study start may be at odds with allocating time to create a data plan prior to study start. However, delays in addressing data management processes will increase risks with regards to achieving the analytic goals of the study and may limit the long-term value of the study data.

For those readers who are medical practitioners it is hoped that this book will spark interest in data driven care as it is expected that all clinicians of the future will have to understand how to apply an expanding wealth of individual and cohort derived information to benefit their patients. Data and software will be increasingly relevant to medical practice as an enabler of efficacious care provided by the physician.

Hawgood et. al. state that the field of precision medicine is at an inflection point. Progress made to date has been promising, although the development of precision medicine tools and techniques will continue to accelerate bringing change to existing research paradigms and the potential for unprecedented understanding of the nature of complex disease. Clinicians, researchers, technologists and patients, all collaborating with the intent of developing safe and effective personalized treatments for complex diseases, will make an enormous difference in alleviating the suffering of patients[10]. Clinicians who do not appreciate and participate in clinical data sciences risk being left behind the precision medicine revolution.

1.6 Scope of this book

This book is meant to be a guide for clinicians beginning their data sciences journey with the aim of increasing their collaborative potential with respect to clinical data management. Think of it as learning a new language to enable conversations about data. Although reading this book will not confer expert level competencies with respect to data science and analysis, it will provide tools to assist in collaborative translational research. The chapters are arranged logically for anyone wishing to explore the subject matter systematically from scientific concepts pertinent to the conduct of translational studies (chapters 2 and 3) to in-depth technical discussions pertaining to data management and analytic processing (chapters 4, 5, 6, 7). Chapter eight provides examples of the application of the technologies discussed in this book. Chapter nine describes Open Source Software fundamentals and the drivers for its adoption. However, each chapter has been written for independent reading and review based on the reader's preference. As such, the reader may find certain redundancies across the chapters, particularly with respect to foundational subject matter.

The book is set up as follows:
Chapter 2 introduces data analysis techniques pertinent for translational study conduct.
Chapter 3 reviews ethical and legal contexts applicable to translational research.

Chapter 4 describes the data life cycle for translational research and corresponding data management strategies.
Chapter 5 reviews analytical techniques commonly applied to translational research data.
Chapter 6 details an open license system, tranSMART, available to support the data management and analytic processing for translational research projects.
Chapter 7 details a high-performance compute environment designed to support high dimensional data analytics for translational research projects.
Chapter 8 provides example projects employing techniques described in prior chapters.
Chapter 9 examines motivations for the production and adoption of Open Source Software.

Chapter 1 References:

[1] Obermeyer Z, Lee TH. 2017. Lost in Thought — The Limits of the Human Mind and the Future of Medicine. NEJM. PMID: 28953443

[2] Nelson and McGuire. 2010. The need for medical education reform: genomics and the changing nature of health information. Genome Med. PMID:20236478

[3] Schork NJ. 2015. Personalized medicine: Time for one-person trials. Nature. PMID: 25925459

[4] Cirillo D and Valencia A 2019 Big data analytics for personalized medicine Current Opinion in Biotechnology 2019, 58:161-167

[5] Hood L, Auffray C. 2013. Participatory medicine: a driving force for revolutionizing healthcare. Genome Med. PMID: 24360023

[6] Mardis ER. 2016. The challenges of big data. Dis Model Mech. PMID: 27147249

[7] McEnroe, N Celebrating Florence Nightingale's Bicentenary Lancet 2020 9-15 May 395(10235) 1475-1478 PMID: 32386583

[8] U-BIOPRED [Internet]. Sheffield (UK): European Lung Foundation; [Date Unknown; Cited Jan 2019]. Available from: https://www.europeanlung.org/en/projects-and-research/projects/u-biopred/home

[9] Hood L, Auffray C. 2013. Participatory medicine: a driving force for revolutionizing healthcare. Genome Med. PMID: 24360023

[10] Hawgood S, Hook-Barnard IG, O'Brien TC, Yamamoto KR. 2015. Precision medicine: Beyond the inflection point. Sci Transl Med. PMID: 26268311.

Chapter 2: The Clinical Research Landscape in the Era of Big Data

Xian Yang and Yike Guo

2.1. Study Designs

One typical way to classify clinical trials is based on how researchers behave.

Observational studies are used to monitor the condition of study participants, assessing their medical status and progress without testing specific medical interventions.
Interventional studies are used to test the safety and/or efficacy of medical therapies comparing the outcomes manifest across different treatment regimens[1].

Figure 2.1 from ("An Overview of Clinical Research: The Lay of the Land" 2002)[2] explains the process of deciding which kind of study design is warranted. If the exposure/treatment is under test by the investigators, then it is an interventional study. With interventional studies, investigators must assign study participants to exposures using an established randomization scheme to ensure confidence in the analysis results[3]. The most popular interventional study is the ***randomized controlled trial*** (RCT). RCTs take a homogenous group of participants and randomly divide them into two groups, ideally with no selection and confounding biases. One group, the treatment group, is exposed to the test therapy while the alternate group acts as a control and is assigned a ***placebo*** (no exposure). Statistical comparison of pertinent medical measurements between the two groups leads to a determination of the effect of the intervention.

For the observational studies, the presence of comparison patient groups is termed an ***analytical*** study while studies having only one cohort are termed ***descriptive***. There are three typical analytical studies: ***cohort study***, ***case-control study*** and ***cross-sectional study***. A cohort study is a longitudinal (duration based) study that samples a cohort to investigate the cause of a disease or pathology[4]. A cohort is a group of people sharing a common characteristic or experience within a defined period (e.g., same birth date, same treatment strategy). An example question to be answered by the cohort study could be whether smoking is associated with lung cancer. The cohort study starts with an exposure (e.g., smoking) and follows people for a few years to measure outcomes (e.g., lung cancer). A case-control analytical study begins with outcome (e.g., lung cancer) and seeks to determine a statistical association of the outcome to an exposure (e.g. smoking). The case-control study is only used to detect factors that would result in a medical condition by comparing people with the condition to people having, ideally,

very similar physical traits but do not manifest the condition. If the study determines exposure and outcome at the same time, then it is a cross-sectional study. A cross-sectional study assesses the prevalence of medical conditions across a target population providing a snapshot of the distribution of a disease in a population at a given time.

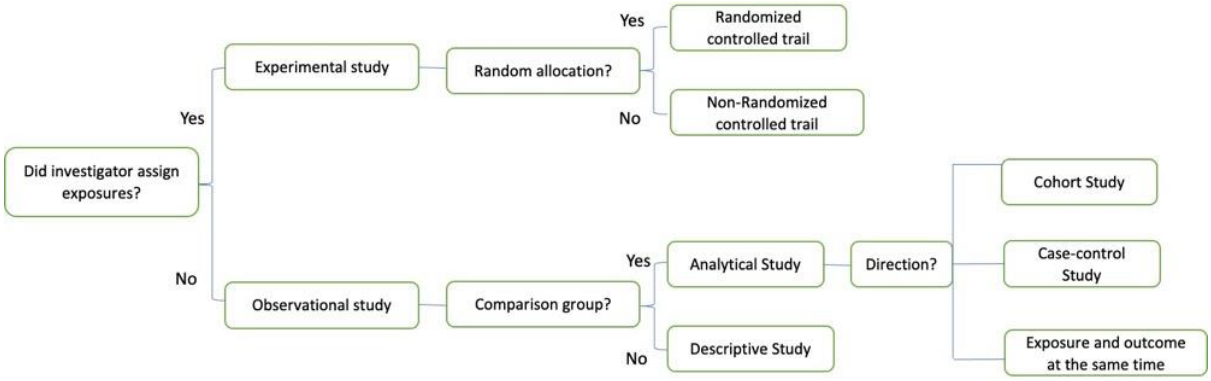


Figure 2.1: The process of choosing an appropriate study design

The advantages and disadvantages of different designs are summarized in Table 2.1[5] [6]. Selection of a design is based on the nature of the research questions of interest. The feasibility, cost, length of time, risk and benefits to the participants must also be considered[7]. Complicated research questions may require multiple studies.

Table 2.1: Advantages and disadvantages of some typical study designs.

Study design	Strengths	Weakness
Cohort study	● Temporality demonstrated Individualized data ● Ability to control for multiple confounders ● Can assess multiple exposures Can assess multiple outcomes	● Expensive ● Time intensive ● Not good for rare diseases
Case-control	● Inexpensive ● Timely ● Individualized data ● Ability to control for multiple confounders ● Good for rare diseases ● Can assess multiple	● Cannot calculate prevalence ● Can only assess one outcome ● Poor selection of controls can introduce bias ● May be difficult to identify enough

	exposures	cases ● Prone to recall bias ● No demonstrated temporality
Cross sectional	● Inexpensive ● Timely ● Individualized data ● Ability to control for multiple confounders; Can assess multiple outcomes	● No temporality ● Not good for rare diseases ● Poor for diseases of short duration ● No demonstrated temporality
RCT	● Unbiased distribution of confounders ● Blinding more likely ● Randomization facilitates statistical analysis	● Expensive: time and money ● Volunteer bias ● Ethically problematic at times

2.2 Statistical power and the clinical study

2.2.1 Sample size calculations

Sample size calculations determine the number of participants needed to detect a clinically relevant treatment effect and are usually the first step in a clinical study design[8]. The sample size should be optimized by considering both the costs associated with recruiting patients and the likelihood of obtaining significant findings[9]. There are three plausible approaches for estimating the sample size during the pre-study phase:

1. Use a comparable dataset from the public domain or a previous study
2. Carry out a pilot study
3. Base estimations on the minimum clinically meaningful difference

Finding public datasets having the same population and experimental conditions as those of the proposed study will likely be difficult. However, discovering comparable datasets collected from a similar population under analogous circumstances may be more likely. A pilot study could be performed if no comparable dataset is identified. The pilot study can be used to estimate effect size, test recruitment, study procedures, and follow-up strategies. Basing the sample size on estimations of the smallest clinically meaningful difference can be used if there are no comparable datasets and a pilot study is not feasible.

2.2.2 Basic statistical methods for calculating sample size

Simple hypothesis testing must be understood to calculate the sample size.

1. **Null hypothesis and alternative hypothesis.** Statistical tests can be used to check whether the difference in means between two populations is significant or not. Null hypothesis H_0 is defined with the statement saying that there is no difference, while the alternative hypothesis H_a is with the opposite statement of H_0. Rejection of null hypothesis means the acceptance of the alternative hypothesis. For instance, if we are investigating the protein concentration levels between healthy people and asthma patients, the null hypothesis can be defined as "the protein has the same concentration level across two groups of people" while the alternative hypothesis would be "the protein is of significantly different levels between two study groups".
2. **Type I error (alpha).** Rejection of the null hypothesis when the null hypothesis is true is known as Type I error[10]**.** Type I error is also called false positive, which occurs when we observe a difference when there is none. The probability of getting Type I error with rejection region R is $P(R|H_0\ is\ true)$**.** It is denoted by the Greek letter α and is also called alpha level. The significant level, which is usually set to 5%, indicates the acceptable probability of getting Type I error.
3. **Type II error (Beta).** Not rejecting a null hypothesis when the alternative hypothesis is true is known as Type II error. Type II error is called as false negative, occurring when we fail to find a difference when in truth there is one. The probability of getting Type II error in a test with rejection region R is $1 - P(R|H_a\ is\ true)$. It is often denoted by the Greek letter β. Conventionally, the β value is set at 20%, meaning that the false negative rate is controlled at the level of 20%. The calculations of different error types are shown in Table 2.2.
4. **Power (1-Beta).** The power of a study reflects the probability of rejecting the null hypothesis when the alternative hypothesis is true. In the case of β equal to 20%, the power is set to 80%, showing the probability of avoiding false negative conclusion. Power analysis is used to calculate the minimum sample size required to detect an effect of given size.
5. **Minimal clinical difference.** The minimum clinical difference is the smallest numeric difference between a study attribute, measured across study groups, that constitutes a distinct physiological response. The minimal clinical difference is set by the investigator, for example, if heart rate is the outcome of a trial, the investigator could choose the difference of 20 beats per minute as indicative of a difference in physiologic response between two individuals.
6. **Effect size.** The effect size is the quantitative measure of the difference in response for a measured attribute between two groups. The effect size is calculated as the difference between the means in two groups divided by the population standard deviation. Effect

size and sample size are inversely related (i.e. a larger the effect size reduces the required sample size for the study)[11].

Table 2.2: Relations between truth/falseness of the null hypothesis

Table of error types		**Null hypothesis (H_0) is**	
		True	**False**
Decision About Null Hypothesis (H_0)	**Reject**	Type I error (False Positive)	Correct inference (True Positive)
	Fail to reject	Correct inference (True Negative)	Type II error (False Negative)

2.2.3 Specific sample size calculations

Each type of statistical analysis requires different elements (e.g., expected proportion, standard deviation) to determine the needed sample size. There is no single method for estimating sample size for all kinds of analysis.

2.2.4 Sample size calculation for univariate analysis

Sample size calculations for single clinical attributes (univariate analysis) are used to determine the precision of estimates, such as proportion and mean.

The population proportion describes the prevalence of a clinical attribute across a defined population [12]. For example, a study may calculate the prevalence (a.k.a. proportion) of diabetes across the adult UK population to be 6%. Proportion is simply count of positive observations (x) in a total population of *N* size. Assuming each observation is independent in the population proportion is commonly estimated using a confidence interval known as a one-sample proportion in the Z-interval [13].

The margin of error is calculated. $E = Z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\mathrm{p})}{N}}$, where $Z_{\alpha/2}$ is the z-value having a tail area of $\alpha/2$to its right.

With a desired error value of E the sample size can be obtained from

$N = \frac{(Z_{\alpha/2})^2\hat{p}(1-\hat{p})}{E^2}$, where $\hat{p}$ is the educated guess of the population proportion. With $0 < \hat{p} \leq 1$, the maximum value of $\hat{p}(1-\hat{p})$is 0.25. Hence, we could get a simplified form for sample size calculation as $N = \frac{(Z_{\alpha/2})^2}{4E^2}$.

Sample sizes for other descriptive statistics, such as mean, are estimated in a similar manner assuming individual values of the attribute under study are normally distributed across the population.

For the mean and standard deviation of a sample population.

The statistic follows the t-distribution as $Z = \frac{\bar{\mathrm{x}}-\mu}{s/\sqrt{N}}$.

The sample size can be estimated by sample t-interval for μ. The margin error is of the form $E = \frac{t_{\alpha/2}S}{\sqrt{N}}$. With a desired margin error, the sample size can be calculated through $N = (\frac{t_{\alpha/2}S}{E})^2$.

One issue we need to pay attention is that $t_{\alpha/2}$ is sample size dependent. Therefore, we need to carry out an iterative process for solving N. That is, we start with an initial guess of N to get its corresponding $t_{\alpha/2}$. Then, using this $t_{\alpha/2}$value, we estimate N. We keep updating N and $t_{\alpha/2}$ in this way until the estimated N is consistent with the value used for $t_{\alpha/2}$

2.2.5 How to calculate sample size for bivariate analysis?

Sample size calculation for bivariate analysis involving two clinical attributes is more complicated. Table 2.3 [14] lists out the required elements for sample size calculation for some typical bivariate analyses. The details of sample size estimation for different bivariate analyses are as follows:

Table 2.3: Required elements for sample size determination for bivariate analyses

Comparison of the two proportions	Comparison of two means	Association of two normally distributed interval variables
Expected percentage in group 1	Effect size	Effect size
Expected percentage in group 2	Standard deviation of interval variable	
Ratio of number of subjects in group 1 to number of subjects in group 2	Alpha	Alpha
Alpha	Power	Power

Power

- Comparison of the two proportions: In the study with null hypothesis of $H_0: p_1 = p_2$ and alternative hypothesis of $H_\alpha: p_1 \neq p_2$, where p_1 and p_2 are proportions in group 1 and group 2, the sample size can be estimated through the following formula: $N = \frac{(z_{\alpha/2}\sqrt{2p(1-p)}+z_{1-\beta}\sqrt{p_1(1-p_1)p_2(1-p_2)})^2}{(p_1-p_2)^2}$, where p equals to $\frac{p_1+p_2}{2}$, $z_{\alpha/2}$ and $z_{z_{1-\beta}}$ are the normal deviates for Type I error and power of study[15 16 17]
- Comparison of two means: In the study with null hypothesis of $H_0: m_1 = m_2$ and alternative hypothesis of $H_\alpha: m_1 \neq m_2$, where m_1 and m_2 are means for group 1 and group 2, the sample size can be estimated by: $N = \frac{(r+1)(z_{\alpha/2}+z_{1-\beta})^2\sigma^2}{rd^2}$, where $r = \frac{n_1}{n_2}$ is the ratio of sample size required for two groups, σ and d are the pooled standard deviation and difference of means of two groups[18].
- Correlation: In the study with null hypothesis of $H_0: r = 0$ and alternative hypothesis of $H_\alpha: r \neq 0$, where r is the correlation between two groups, the sample size can be estimated through the following formula: $N = \frac{(z_{\alpha/2}+z_{1-\beta})^2}{\frac{1}{4}[log_e(\frac{1+r}{1-r})]}$, where r is the correlation between two variables[19].

2.2.6 Calculating sample size for multivariate analysis

Conventionally, the minimum sample size required for most multivariate analyses involving sets of more than two clinical attributes is determined using the rule-of-thumb, which is mostly derived from multiple linear regression (MLP). Some sample size guidelines suggest the ratio between the number of independent variables and subjects is 1 to 10[20] or 1 to 30[21]. In Knofczynski and Mundfrom 2007, the minimum sample size for using MLR for prediction is suggested to be varied according to the effect sizes[22]. In Wilson Van Voorhis et al. 2007, an overview of the sample size rules of thumb is shown as it is in Table 2.4[23].

Table 2.4: Sample size rules of thumb

Relationship	**Reasonable sample size**
Measuring group differences (e.g., t-test, ANOVA)	Cell size of 30 for 80% power, if decreased, no lower than 7 per cell[24].
Relationships (e.g., correlations, regression)	~50

Chi-square	At least 20 overall, no cell smaller than 5.
Factor Analysis	~300 is "good"

2.2.7 Sample size feasibility

There may be the case that the estimated sample size is larger than the maximum number of study participants that could be enrolled into a study. In this case, the following should be considered before abandoning the study[25]:

1. Is there more sensitive marker of the outcome? For example, death due to lung disease is a more sensitive marker of smoking than other diseases such as heart attack.
2. Can we repeated measurements be collected? Repeated measurements can increase the number of observations without increasing the sample size.
3. Can power be relaxed? For example, we could decrease the power from 90% to 80%.
4. Can the population be limited to individuals more likely to experience the outcome under study? For example, setting a minimum age and BMI enrollment criteria could foster insight regarding the impact of elevated cholesterol on the risk of myocardial infarction using a smaller study population.

If the above strategies are not applicable, the researcher must consider pragmatic solutions, such as seeking additional funding, or reassess the scope of the proposed investigation.

2.3. The N-of-1 trial

2.3.1 The value of N-of-1 trials

Heterogeneity with respect to the underlying mechanisms of complex disease results in inconsistent efficacy of generalized treatment across patients regardless of whether treatments of been demonstrated to be effective on average[26]. Therefore, large scale randomized controlled trials (RCTs) cannot comprehensively address all clinical problems across patient populations [27] [28].

An N-of-1 (single subject) clinical trial investigates the optimal medical treatment for an individual patient using the objective data-driven criteria. As reported in Gabler et al. 2011 [29], more than 2,154 N-of-1 trials across 108 studies have been conducted between 1985-2010 addressing various clinical conditions, such as neuropsychiatric, musculoskeletal and pulmonary. Such trials are becoming more prevalent due to increased appreciation of disease heterogeneity, the availability of datatypes capable of assessing individual physiologic variations and data processing methods that allow individual trials to be conducted efficiently.

The indications and contraindications for N-of-1 trials are summarized in [30].

2.3.2 What are typical designs for the N-of-1 trial?

In the N-of-1 trial, datasets are collected longitudinally as frequently as possible for an individual patient[31]. The basic design principles include randomization, blinding, replication and carryover [32] ("Design and Implementation of N-of-1 Trials: A User's Guide | Effective Health Care Program" n.d.):

- **Randomization/Counterbalancing:** To maintain the variability of experiments, patients are given a sequence of treatments, which can be either randomly generated or definitively assigned [1,2]. Most commonly, treatments are administered consecutively but separated by wash out periods (also termed ***carryover***) in which the study participant receives no treatment to return the subject to their baseline physiology. Assuming two treatment protocols, A and B, a randomized four-period trial could be conducted longitudinally according to the following designs: ABAB, BABA, ABBA and BAAB. ABAB and BABA are ***unbalanced design*s** as the treatments are rotated such that no single treatment is administered in consecutive periods. Conversely ***counterbalancing designs*** alternate such that one or more treatments are administered repeatedly (i.e. ABBA and BAAB).
- **Blinding:** Patients should be kept blinded to the treatment design although there is a risk that patients may deduce the treatment regime based on; for example, their responses to the treatments, which could confound outcomes.
- **Replication:** Sample size in N-of-1 studies refers to the number of periods and measurements collected during the periods.

2.3.3 What are the analysis methods for the N-of-1 trial?

The analysis methods in N-of-1 trial fall into the following categories: visual inspection, statistical analysis, time series analysis and Bayesian methods[33] [34] [35] [36] [37] [38] [39] Table 2.5 from Gabler et al. 2011 [40] has summarized these analysis methods. The analysis methods are designed to compare treatment results accounting carryover and randomization effects.

Table 2.5: Analysis methods for the investigated 108 N-of-1 trials

Method	Number of studies
Pooled analysis	26

Bayesian	6
Other	20
Nonparametric analysis	24
Wilcoxon signed-rank test	8
χ^2	4
Mann-Whitney	3
Fisher exact test	3
Sign test	4
Other	7
Graph or visual examination	56
T test	48
Regression model	18
ANOVA	13
Other	5

Let us discuss some example methods as follows:

- **Visual inspection:** Many studies do not have formal analysis except visual inspection. People use graphs to compare the outcomes of two treatments. This method can only work well for simple datasets with obvious differences between treatments
- **Statistical methods:** The simplest statistical test in N-of-1 trials is the sign test. Suppose under the two-treatment case, treatments are randomized in blocks of two periods. The difference in response for each block is calculated and assigned a sign (+/-) depending on the difference. Binomial analysis can be used to determine relative treatment efficacy although insightful inferences with respect to effect size may be ignored.
- **Bayesian methods:** Bayesian (conditional probability) methods incorporate results from the same N-of-1 study design conducted across many patients. Bayesian inference may challenge individual treatment responses that may appear significant in isolation but are not significant when analyzed across many similar patient trials.
- **Time series analysis:** Attribute values collected consecutively throughout a trial may not appear to be independent measures. This bias associated with time-consecutive measures must be corrected during analysis and models exist that adjust the values of a present measurement based on that prior measurement[41].

2.4. Data types associated with translational research

2.4.1 Molecular datasets

Many types of molecular and physiological data should be collected and integrated during a translational research study[42]. Molecular data types that are typically collected include gene sequences, gene expression (measured by the ***microarray*** or ***RNA sequencing*** technologies) and protein expression (measured by the mass spectrometry)[43] with each data type profiling different aspects of an individual's molecular physiology. Analyzing molecular datasets *can* lead to the discovery of biomarkers predictive of disease or indicative of a specific pathological state.

The U-BIOPRED project[44] used samples and medical information from hundreds of severe asthmatics to stratify disease subtypes. This work accelerated the discovery of novel diagnostic and therapeutic targets for asthma. U-BIOPRED generated various high dimensional Omics datasets, including genome wide association (GWAS), Transcriptomics, Proteomics, Lipidomics and Breathomics. Moreover, this project also generated low dimensional histological, morphological, clinical and patient reported outcome datasets to comprehensively model asthma phenotypes.

Figure 2.2 shows typical steps of carrying out the U-BIOPRED project:

1. The first step is to collect patient samples and construct biobanks for sample storage and management. Cross-sectional and longitudinal cohort studies for both adult and pediatric healthy controls and asthma patients are well designed.
2. The second step is using data-driven approach to stratify patients of different groups using the "handprint", which are extracted from high-throughput 'Omics' data and patient clinical data[45].
3. The third step is to validate handprints and investigate the asthma phenotypic features.
4. The fourth step is to refine phenotype 'handprints' with pre-clinical and human exacerbation models.

The most important step for generating handprint is step 2, which adopts unbiased approach to generating disease 'handprints' based on comprehensive and integrative analysis of various Omics data.

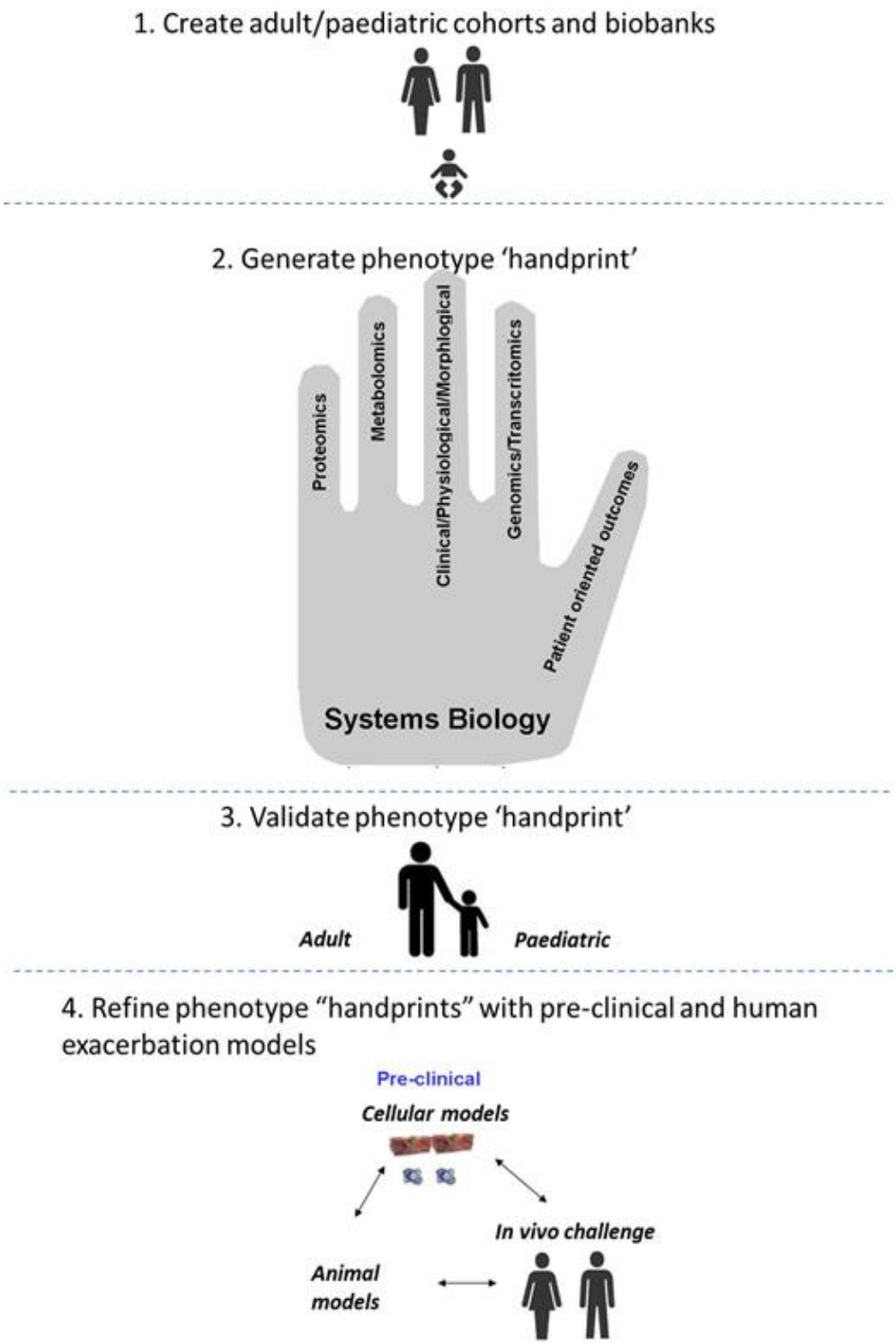


Figure 2.2: The steps of carrying out the U-BIOPRED project.

2.4.2 Using molecular datasets

Molecular data profiles can be used for a variety of purposes. The following are examples pertinent to U-BIOPRED.

Mapping molecules to pathways

Omics datasets can be integrated with traditional clinical datasets to generate a *handprint* (patient profile) for asthma phenotyping. Omics feature profiles can be matched to specific subgroups of severe asthma. These Omics features, genes, proteins and other molecules, can be mapped to molecular pathways to generate hypothesizes as to the causative nature of the disease subtypes. These hypotheses can be tested experimentally in the laboratory or by using in silico models of biological processes (see Figure 2.3). Molecular pathway modelling a time-consuming process given the massive number of potential molecular interactions. However, identifying disease-related pathways is critical to understanding the biological processes of disease.

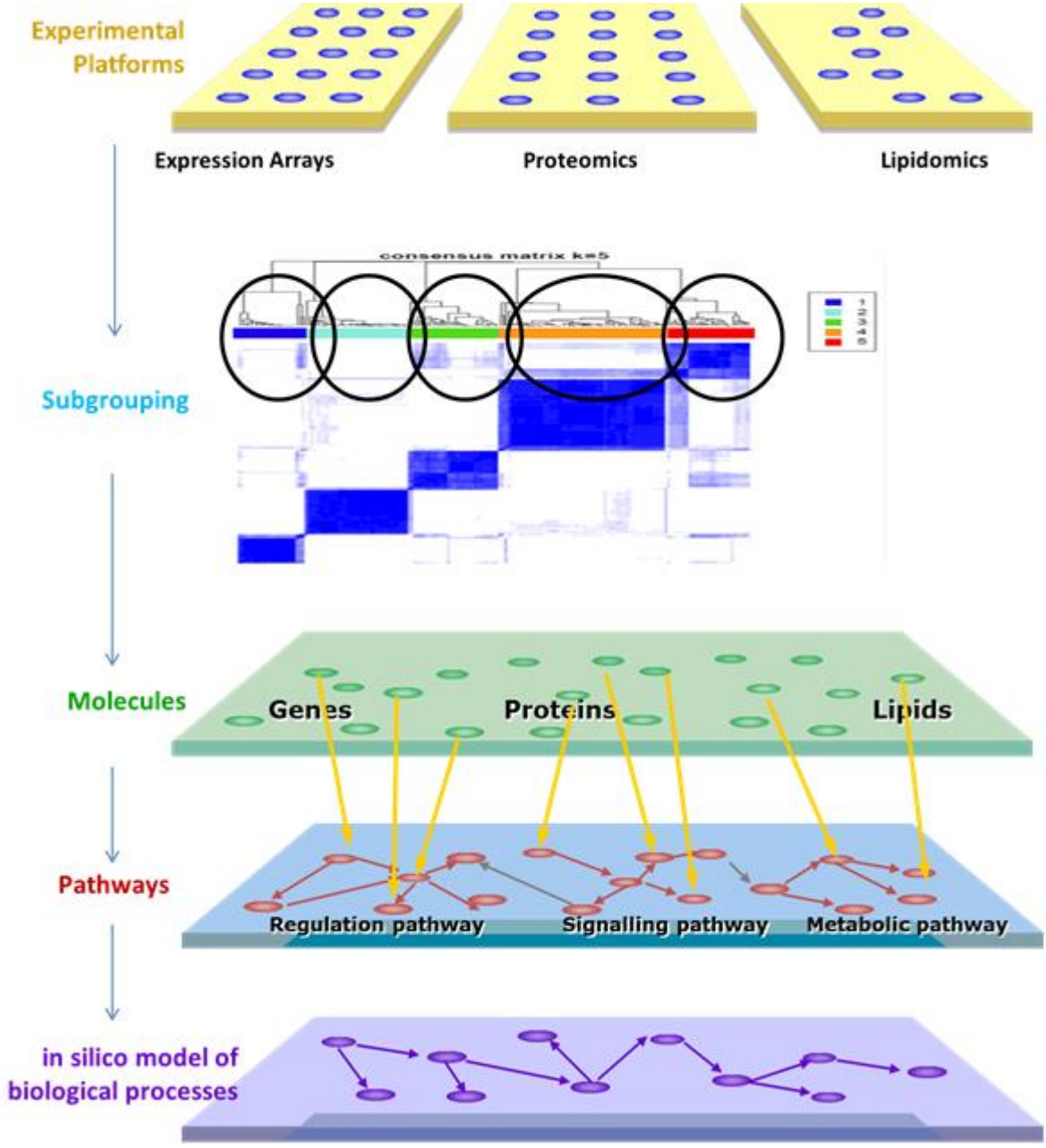


Figure 2.3: Process of obtaining pathway models for understanding sub-phenotypes.

Modelling pathways

Having detected potential disease related pathways, the next step is to understand disease mechanism by dynamic modelling of pathways[46 47]. The UBIOPRED team worked with clinicians and biologists to construct an integrated MAPK-38 and GR pathway model to explain corticosteroid resistance in patients suffering from severe asthma [48]. ***Corticosteroids*** (CS) are essential stress hormones that regulate many physiological processes including immune function and cell proliferation. These hormones are used as treatments for asthma due to their anti-inflammatory and immunosuppressive properties. The binding between CSs and ***glucocorticoid receptor*** (GR) results in nuclear translocation. Subsequent attachment of activated GR to DNA in the nucleus leads to gene expression modulation via ***transactivation*** [49]. In the meantime, to suppress the proinflammatory cytokine transcription through ***transrepression***, activated GR can interact with other transcription factors. However, for asthma patients the anti-inflammatory of CS is impaired. The impaired suppression of pro-inflammatory cytokines by dexamethasone has been found to be related to augmented activation of ***p38 MAPK*** [50 51]. Therefore, it is necessary to study interactions of p38 MAPK pathway with the GR-induced signaling pathway.

Constructing a mechanistic model is a plausible way to understand the mechanism of corticosteroid responsiveness in inflammatory diseases. However, most studies only focus on mechanistic models of isolated pathways such that integrated models of various interconnected pathways are rarely investigated. For example, a model for LPS-induced p38 pathway can be found in the online pathway databases [52]. This p38 model was used to construct a novel mechanistic model of the GR pathway based on the known biological reactions. The work led to a proposed interaction (crosstalk model) between these pathways (see Figure 2.4). Potential entities that crosstalk could happen are ***TGF kinase-1*** (TAK1), ***MAPK phosphatase-1*** (MKP-1) and ***phosho-p38*** itself. However, this proposed interaction model is found to be difficult to be validated by wet-lab experimental observations. The main challenge in constructing pathway models is determining accurate and complete model parameters (e.g., kinetic rates and initial concentrations) from limited time series measurements. Converging to a unique parameter set solution is therefore difficult[53]. Pathway models can be simplified at the risk of ignoring important and insightful interactions. Therefore, it is better to develop and maintain models that are as detailed and accurate as achievable, available data and resources.

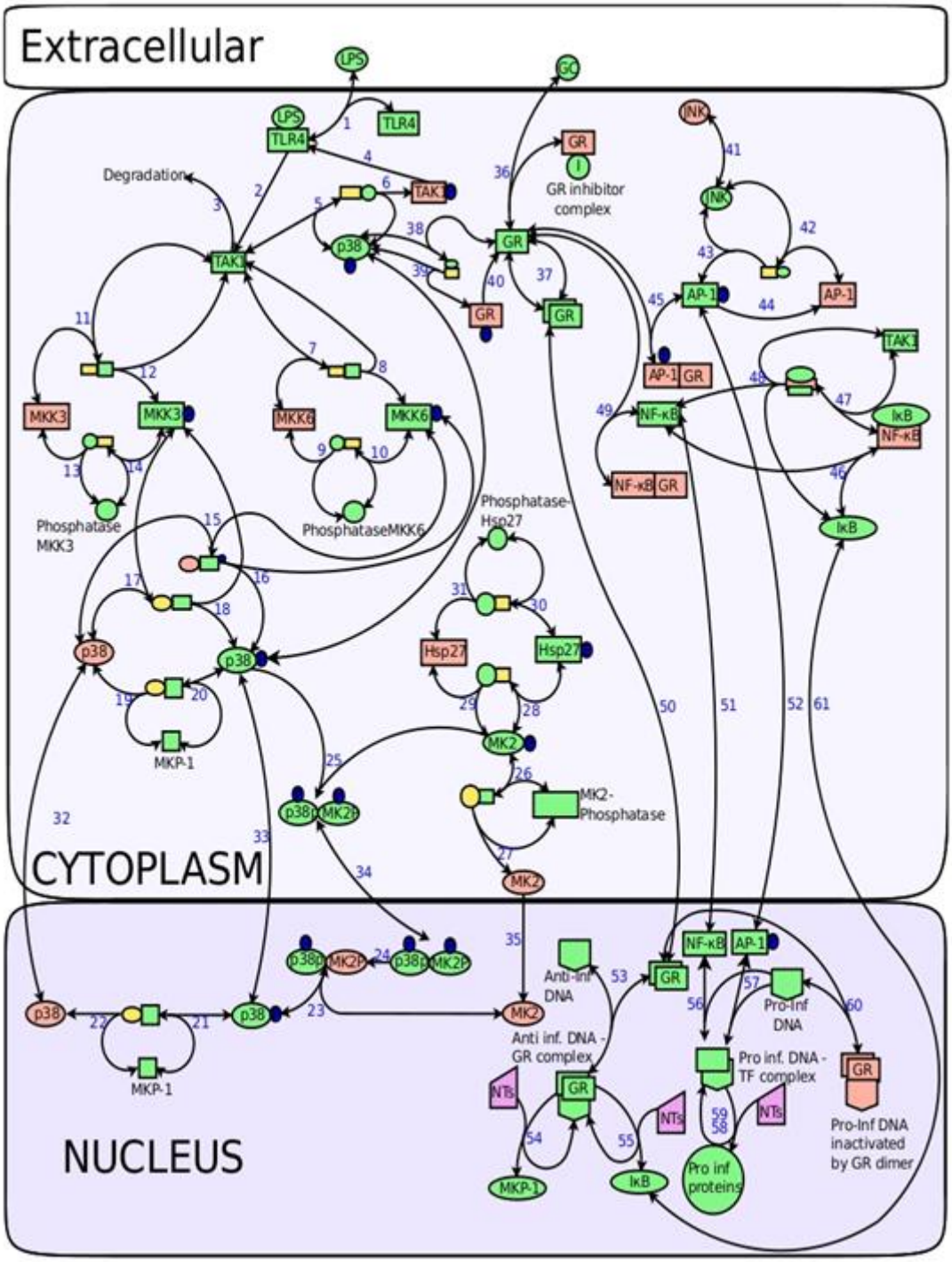


Figure 2.4: The integrated pathway of p38 MAPK and GR from[54] Holehouse et al. 2012

2.4.3. Electronic health Records

Electronic health records (EHR) are another critical source of data. The application of EHRs in the clinic can improve the quality of patient care in various ways, such as supporting

pragmatic clinical trials and big data driven discovery[55] [56]. Recent studies have shown that secondary use of EHRs has enabled data-driven prediction of drug effects and interactions[57], identification of type-2 diabetes subgroups[58], discovery of comorbidity clusters in autism spectrum disorders[59] and improvements in recruiting patients for clinical trials[60].

Using EHRs to construct predictive models is challenging due to the highly multivariant nature, noise, sparseness and lack of completeness associated with EHR data [61] [62] [63]. Data driven approaches have been proposed to overcome these problems [64] [65] [66]. These methods include supervised [67], unsupervised models, including the use of recurrent neural networks (RNN) by Choi et al. 2016 [68]. As EHRs are being increasingly generated, high performance compute environments such as ***Hadoop*** will be necessary to train automated classification methods to identify patterns within EHR databases[69].

2.4.4 Medical imaging

Medical imaging, capable of characterizing morphological and functional properties of tissues, can also be used to construct personalized models of disease. ***Magnetic resonance imaging*** (MRI) [70] , ***computed tomography*** (CT) [71] , ***positron emission tomography*** (PET) [72] and ***ultrasound*** are common medical image modalities available to medical researchers capable of discriminating anatomical features, infusion of molecules within tissues and material properties[73]. As a non-invasive tool, imaging enables the study of tissues that are difficult to sample through biopsy, such as lesions in cancer patients ("The Evolution of Medical Imaging In Clinical Research" n.d.[74]). The functional nature of many imaging methods, such as those based on time-lapsed MRI and ultrasound, greatly facilitate diagnostics for many conditions and have a tremendous impact in medical practice [75] [76]. The availability of automated image processing and analysis software promotes the use of imaging for translational research.

2.4.5 Wearable biosensors

Wearable biosensors have emerged in both consumer and medical markets. The application of medical images and wearable biosensors to assess medical conditions, i.e. ***digital biomarkers***, are increasingly being incorporated into clinical trials. Continuous monitoring of patients outside of medical facilities is a great benefit of the clinical application of wearable sensors[77]. There are now many wearable biosensors designed to monitor patients having specific conditions. Examples include devices that track the physiologic and kinetic parameters associated with disabilities resulting from multiple sclerosis[78] and accelerometers worn on the wrists of Parkinson's disease patients to monitor circadian sleep patterns as well as motor and autonomic disruptions.

Wearable biosensors are starting to decouple some elements of medical examination from office visits with certain medical abnormalities detectable in real time without burdening the schedule of the patient. Applications, leveraging both standard and augmented capabilities of the ubiquitous mobile phone, are available to measure blood pressure [79], identify cervical cancer[80], and even perform an eye exam [81]**.** Besides the mobile phone, personal items such as smart helmets and clothing-embedded sensors are used for health monitoring, disease treatment and detection [82].

2.4.6 Social media

Social media, such as Facebook, Twitter, blogs and Wikipedia, are web-based tools for people to create, share, comment upon or modify content[83] [84]. Data available from these sources are often documents that do not conform to consistent formats (***unstructured data***) and, therefore, are typically difficult to incorporate into medical analyses. However, these data can contain information useful to the clinical researcher such as lifestyle preferences and medically related experiences[85]. Social media provides patients a way to communicate with fellow patients and clinicians through online communities that are not limited by geographic boundaries.[86]

2.4.7 Data standards

Data standards in clinical research refer to methods, protocols, terminologies and specifications for collecting, exchanging, storing and retrieving clinical information. Table 2.6 from Bioinformatics for Omics Data[87] lists the major reporting standards for various Omics data

types. Reporting standards promote consistency in the representation of experimental designs, clinical measurements and analysis results. Consistent data representations promote and expedite research data exchange, data storage and software data processing. Table 2.7 from Bioinformatics for Omics Data[88] lists the most popular exchange standards for Omics datasets. The ***DICOM*** (Digital Imaging and Communications in Medicine) format is, by far, the most commonly used data standard for medical images.

Table 2.6: Existing reporting standards for Omics.

Acronym	**Domain**
CIMR	Metabolomics
MIAME	Transcriptomics
MIAPE	Proteomics
MIGS-MIMS	Genomics
MIMIx	Proteomics
MINIMESS	Metagenomics
MINSEQE	Genomics, Transcriptomics (UHTS)
MISFISHIE	Transcriptomics

Table 2.7: Data exchange and modelling standards for Omics.

Data format	**Object model**	**Domain**
FuGE-ML	FuGE-OM	Multiomics
ISA-TAB		Multiomics
MAGE-ML	MAGE-OM	Transcriptomics
MAGE-TAB	MAGE-OM	Transcriptomics
MIF (PSI-MI XML)		Proteomics
MzML		Proteomics
mzIdentML		Proteomics
PML	PAGE-OM	Genomics
PML	SDTM	Healthcare

2.5. Big Data Analytics

2.5.1 Big data analytics in clinical research?

The term ***Big Data*** traditionally referred to datasets that consume at least one terabyte of memory on a computer storage device. Datasets at this volume tended to require data processing systems more powerful than a standard personal computer, involvement of computer professionals for data management and specialized software for data processing. Although there is no precise definition of Big Data, the term heralded a new era of computing potential in which large scale datasets, interrogated by sophisticated mathematical pattern recognition methods, could be leveraged to obtain transformative insights beyond those intended at the time such datasets were originally conceived and collected.

The criteria that classically describe big datasets are *volume* (memory consumption), *velocity* (rate of data generation), *variety* (number of data attributes), *veracity* (data quality, correctness) and, sometimes, *value* (interest with respect to data consumers). For translational research, molecular and digital biomarker data, especially raw data generated by instruments and devices, will generally meet the volume criteria of big data. The rate of data generation from wearable devices will likely constitute a big data challenge for clinicians. Low dimensional clinical study datasets, which can number hundreds to thousands of attributes, and EHRs are highly variable dataset. Reuse of translational information is a key value proposition justifying the creation of such data and careful management of these data with respect to availability and standardization will enhance the value of these data to the research community.

Big data analytics projects require specialized data processing environments that are typically not necessary for conducting traditional clinical research programs. A common strategy for accelerating data analysis for large scale data sets is to subdivide and distribute these datasets across many computers for parallel processing. The results of these independent processing events are then aggregated into a derived reduced dataset. Open source platforms such as ***Hadoop/Map Reduce***, which provide specific implementations of distributed processing models, are increasingly used for big data analytics in clinical research. End users must approach Big Data analytics different from local data processing. For example, distributing standard statistical methods, that may be available to scientists through personal computer or server-based applications such as SAS and R, might require intensive, complicated programming efforts to design corresponding algorithms that operate efficiently on a distributed compute environment. Figure 2.5 from Raghupathi and Raghupathi 2014[89] presents a conceptual architecture of big data analytics. In this figure, big data in clinical research emanates from various resources and in various formats. Transformation tools need to efficiently process, modify and store raw big datasets in preparation for subsequent analysis. Following data processing, big data analytics platforms and tools are used to analyze these

data. Common big data platforms and tools can be found in Raghupathi and Raghupathi 2014[90]. Typical applications of big data analytics are high performance queries, report generation, online analytical processing (OLAP), and data mining/pattern recognition.

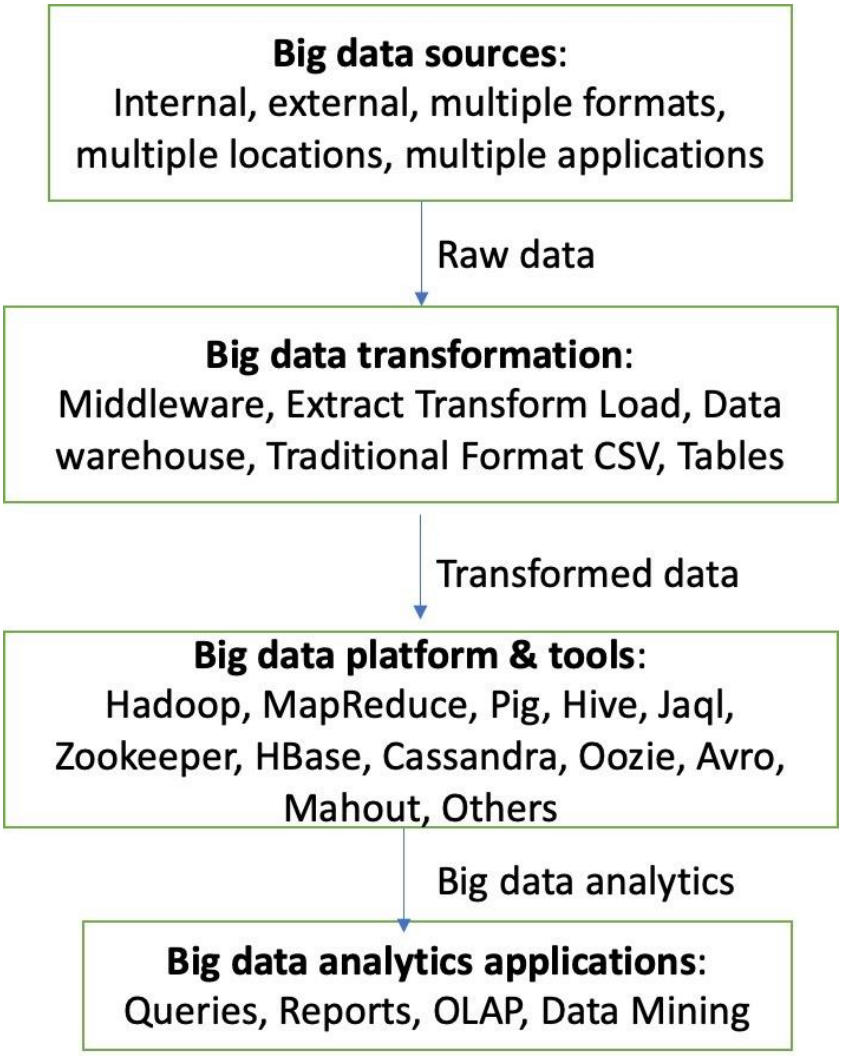


Figure 2.5: The conceptual architecture of big data analytics.

2.5.2 Stages of big data analytics in clinical research

The main stages of applying big data analytics in clinical research are summarized in Figure 2.6 (adapted from Raghupathi and Raghupathi 2014[91]). The initial stage builds a conceptual goal that, if of a large enough scale, establishes the need for big data analytics. The next stage probes the significance of the project through activities such as compiling a priori knowledge (literature review) and background materials such as pertinent existing datasets and implementations of potentially useful analytic methods. The third stage defines the structure of the datasets, collects and transformed the data and selects, builds and applies the analysis methods. The core part of this stage is the platform/tool evaluation and selection as listed in

Table 2.9. The fourth stage evaluates, validates and tests the analytic model and its derived results, following which, the analysis can be confidently used to generate insights.

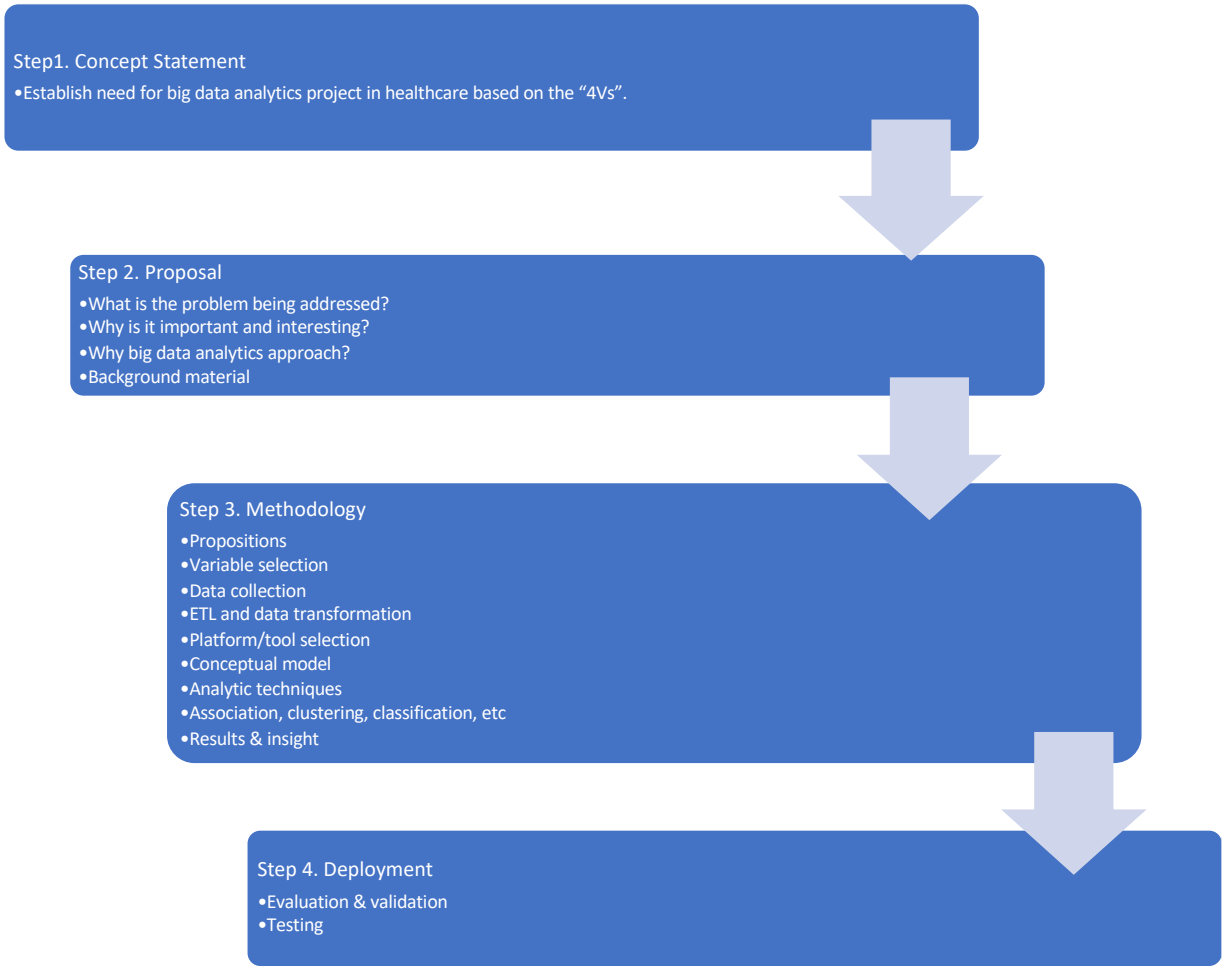


Figure 2.6: Key stages of big data analytics methodology

Summary

This chapter introduced the clinical study and the variations likely to be encountered when prosecuting translational research. Basic clinical study design was examined including biomarker extensions pertinent to the pursuit of individualized treatments for subtypes of complex diseases. Hypothesis generation through the association of experimentally derived molecular profiles with biological pathways was explored. Digital biomarkers including medical images and wearable biosensors were reviewed as elements of large-scale datasets which require specialized systems for data management and processing.

Designing, conducting and analyzing clinical studies requires exceptionally specialized training and substantial experience. This chapter was written to provide readers with limited familiarity with clinical data management and analysis an appreciation of the types of activities that are necessary to operationalize precision medicine investigations. The following chapter will detail data management procedures for translation research.

Chapter 2 References:

[1] Hannan, EL. 2008. Randomized Clinical Trials and Observational Studies: Guidelines for Assessing Respective Strengths and Limitations. JACC. PMID: 19463302

[2] Grimes DA, Schulz KF. 2002. An Overview of Clinical Research: The Lay of the Land. The Lancet. PMID: 11809203

[3] Schulz, KF. 1997. Assessing the Quality of Randomization From Reports of Controlled Trials. JAMA. PMID: 8015122

[4] Song, JW, and Chung KC. 2010. Observational Studies: Cohort and Case-Control Studies. Plast. Reconstr. Surg. PMID: 20697313

[5] Thiese, MS. 2014. Observational and Interventional Study Design Types; an Overview. Biochemia Medica. PMID: 24969913

[6] Study Designs [Internet]. Oxford (UK): Centre for Evidence Based Medicine; [April 3, 2014]. Available from: http://www.cebm.net/blog/2014/04/03/study-designs/

[7] Skinner, H. Study Design and Statistical Analysis: A Practical Guide for Clinicians. Cambridge (UK). Cambridge University Press, 2006.

[8] Noordzij M et al. 2010. Sample Size Calculations: Basic Principles and Common Pitfalls, Nephrology, Dialysis, Transplantation. European Dialysis and Transplant Association - European Renal Association. PMID: 20067907

[9] Skinner, H. Study Design and Statistical Analysis: A Practical Guide for Clinicians. Cambridge (UK). Cambridge University Press, 2006.

[10] Bhattacharya R, Lin L, Patrangenaru V. Multiple Testing and the False Discovery Rate. In: A Course in Mathematical Statistics and Large Sample Theory. Springer, New York, NY, 2006.

[11] Prashant K, Supriya B. 2010. Sample Size Calculation, International Journal of Ayurveda Research. PMID: 20532100

[12] Tintle N. Introduction to Statistical Investigations. Hoboken (NJ). Wiley, 2016.

[13] Starnes DS, Tabor J. The practice of statistics: for the AP exam. New York: Bedford, Freeman, & Worth, high school publishers; 2018.

[14] Skinner, H. Study Design and Statistical Analysis: A Practical Guide for Clinicians. Cambridge (UK). Cambridge University Press, 2006.

[15] Suresh K, Chandrashekara S. 2012. Sample size estimation and power analysis for clinical research studies. PMID: 22870008

[16] Cohen JW. Statistical power analysis for the behavioral sciences. New York: Academic Press; 2013.

[17] Guenther WC. Power and sample size for approximate chi-square tests. Laramie: Division of Business and Economic Research, College of Commerce and Industry, University of Wyoming; 1977.

[18] Suresh K, Chandrashekara S. 2012. Sample size estimation and power analysis for clinical research studies. PMID: 22870008

[19] Ibidem.

[20] Miller DE, Kunce JT. 1973. Prediction and Statistical Overkill Revisited.

[21] Pedhazur EJ, Schmelkin LP. 2013. Measurement, Design, and Analysis.

[22] Knofczynski GT, Mundfrom D. 2007. Sample Sizes When Using Multiple Linear Regression for Prediction.

[23] Van Voorhis W et al. 2007. Understanding power and rules of thumb for determining sample sizes. TQMP. Vol. 3: p 43-50

[24] Cohen J. 2013. Statistical Power Analysis for the Behavioral Sciences. Cambridge (MA): Academic Press.

[25] Skinner, Halcyon. 2007. Study Design and Statistical Analysis: A Practical Guide for Clinicians. Ann. Intern. Med.

[26] Jørgensen JT. 2008. Are We Approaching the Post-Blockbuster Era? Pharmacodiagnostics and Rational Drug Development. Exp. Rev. Mol. Diag. PMID: 18999921

[27] Brook RH. 2009. Possible Outcomes of Comparative Effectiveness Research. JAMA. PMID:19584350

[28] Tanenbaum SJ. 2009. Comparative Effectiveness Research: Evidence-Based Medicine Meets Health Care Reform in the USA. J. Eval. Clin. Pract. PMID: 20367695

[29] Gabler NB et al. 2011. N-of-1 Trials in the Medical Literature: A Systematic Review. Medical Care. PMID: 21478771

[30] Duan N et al. 2013. Single-Patient (n-of-1) Trials: A Pragmatic Clinical Decision Methodology for Patient-Centered Comparative Effectiveness Research. J. Clin. Epidemiol. PMID: 23849149

[31] Schork NJ. 2015. Personalized Medicine: Time for One-Person Trials. Nature. PMID: 25925459

[32] Design and Implementation of N-of-1 Trials: A User's Guide | Effective Health Care Program [Internet]. 2018. U.S. Department of Health and Human Services. [Date Unknown; Accessed January 4]. Available from: https://effectivehealthcare.ahrq.gov/topics/n-1-trials/research-2014-5

[33] Lillie EO et al. 2011. The N-of-1 Clinical Trial: The Ultimate Strategy for Individualizing Medicine. Personalized Medicine. PMID: 21695041

[34] Gabler NB et al. 2011. N-of-1 Trials in the Medical Literature: A Systematic Review. Medical Care. PMID: 21478771

[35] Nikles J et al. 2011. Aggregating Single Patient (n-of-1) Trials in Populations Where Recruitment and Retention Was Difficult: The Case of Palliative Care. J. Clin. Epidemiol. PMID: 20933365

[36] Zucker DR et al. 1997. Combining Single Patient (N-of-1) Trials to Estimate Population Treatment Effects and to Evaluate Individual Patient Responses to Treatment. J. Clin. Epidemiol. PMID: 9179098

[37] Zucker DR et al. 2006.Lessons Learned Combining N-of-1 Trials to Assess Fibromyalgia Therapies. J. Rheumatol. PMID: 17014022

[38] Zucker DR. 2010. Individual (N-of-1) Trials Can Be Combined to Give Population Comparative Treatment Effect Estimates: Methodologic Considerations. J. Clin. Epidemiol. PMID: 20863658

[39] Guyatt G et al. 1998. A Clinician's Guide for Conducting Randomized Trials in Individual Patients. CMAJ. PMID: 3409138

[40] Gabler NB et al. 2011. N-of-1 Trials in the Medical Literature: A Systematic Review. Medical Care, PMID: 21478771

[41] Schmid CH. 2001. Marginal and Dynamic Regression Models for Longitudinal Data. Stat. Med. PMID: 11746319

[42] Chen R. et al. 2012. Personal omics profiling reveals dynamic molecular and medical phenotypes. Cell. PMID: 22424236

[43] Schneider M V. Orchard S. 2011. Omics technologies, data and bioinformatics principles. Methods Mol. Biol.

[44] Shaw DE et al. 2015. Clinical and inflammatory characteristics of the European U-BIOPRED adult severe asthma cohort. Eur. Respir. J. PMID: 26357963

[45] De Meulder B et al. 2018. A computational framework for comples disease stratification for multiple large-scale datasets. BMC Syst Biol. PMID: 29843806

[46] Wolkenhauer O. 2014. Why Model?. Front. Physiol. PMID: 24478728

[47] Kholodenko BN. 2006. Cell-Signalling Dynamics in Time and Space. Nat. Rev. Mol. PMID: 16482094

[48] Holehouse A et al. 2012. Developing a Novel Integrated Model of p38 MAPK and Glucocorticoid Signalling Pathways. CIBC. PMID: 24084075

[49] Nelson HS et al. 2003. Update on Glucocorticoid Action and Resistance. J. Allergy Clin. Immunol.

[50] Bhavsar P et al. 2010. Effect of p38 MAPK Inhibition on Corticosteroid Suppression of Cytokine Release in Severe Asthma. Eur. Respir. J. PMID: 19840967.

[51] Hew M et al. 2006. Relative Corticosteroid Insensitivity of Peripheral Blood Mononuclear Cells in Severe Asthma. AJRCCM. PMID: 16614347

[52] Hendriks BS et al. 2008. Analysis of Mechanistic Pathway Models in Drug Discovery: p38 Pathway. Biotechnol. Prog.

[53] Ibidem

[54] Holehouse A et al. 2012. Developing a Novel Integrated Model of p38 MAPK and Glucocorticoid Signalling Pathways. CIBCB.

[55] Taglang G and Jackson DB. 2016. Use of ‘big Data’ in Drug Discovery and Clinical Trials. Gynecol. Oncol. PMID: 27016224

[56] Hersh WR. 2007. Adding Value to the Electronic Health Record through Secondary Use of Data for Quality Assurance, Research, and Surveillance. Am. J. Manag. Care. PMID: 17567224

[57] Tatonetti NP et al. 2012. Data-Driven Prediction of Drug Effects and Interactions. Sci. Transl. Med. PMID: 22422992

[58] Li, Li, Wei-Yi Cheng BS et al. 2015. Identification of Type 2 Diabetes Subgroups through Topological Analysis of Patient Similarity. Sci. Transl. Med. PMID: 22422992

[59] Doshi-elez F et al. 2014. Comorbidity Clusters in Autism Spectrum Disorders: An Electronic Health Record Time-Series Analysis. Pediatrics. PMID: 4323995

[60] Miotto R and Chunhua W. 2015. Case-Based Reasoning Using Electronic Health Records Efficiently Identifies Eligible Patients for Clinical Trials. J. Am. Med. Inform. Assoc. PMID:25769682

[61] Jensen PB et al. 2012. Mining Electronic Health Records: Towards Better Research Applications and Clinical Care. Nat. Rev. Gen. PMID: 22549152

[62] Weiskopf NG and Chunhua W. 2013. Methods and Dimensions of Electronic Health Record Data Quality Assessment: Enabling Reuse for Clinical Research. J. Am. Med. Inform. Assoc. PMID: 22733976

[63] Weiskopf NG et al. 2013. Defining and Measuring Completeness of Electronic Health Records for Secondary Use. J. Biomed. Inform. PMID: 23820016

[64] Huang SH et al. 2014. Toward Personalizing Treatment for Depression: Predicting Diagnosis and Severity. J. Am. Med. Inform. Assoc. PMID: 24988898

[65] Lyalina S et al. 2013. Identifying Phenotypic Signatures of Neuropsychiatric Disorders from Electronic Medical Records. J. Am. Med. Inform. Assoc. PMID: 23956017

[66] Wang X, Sontag D, Wang F. 2014. Unsupervised Learning of Disease Progression Models. KDD ’14.

[67] Miotto R et al. 2016. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. Scientific Reports.

[68] Choi E et al. 2016. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. JMLR Workshop and Conference Proceedings, PMID: 28286600

[69] Ng, Kenney, A Ghoting et al. 2014. PARAMO: A PARAllel Predictive MOdeling Platform for Healthcare Analytic Research Using Electronic Health Records. J. Biomed. Inform. PMID: 24370496

[70] Hollingworth W et al. 2000. The Diagnostic and Therapeutic Impact of MRI: An Observational Multi-Centre Study. Clin. Radiol. PMID: 11069736

[71] Herman GT. 2009. Fundamentals of Computerized Tomography: Image Reconstruction from Projections. Springer.

[72] Bailey DL et al. 2004. Positron Emission Tomography: Basic Science. Springer.

[73] Sensen CW, Hallgrimsson B. 2008. Advanced Imaging in Biology and Medicine: Technology, Software Environments, Applications. Springer Science & Business Media.

[74] The Evolution Of Medical Imaging In Clinical Research. 6 Feb 2017. Erie (PA): Clinical Leader; [Date Unknown; Accessed January 8, 2018] Available from https://www.clinicalleader.com/doc/the-evolution-of-medical-imaging-in-clinical-research-0001

[75] Khan NL et al. 2005. Mutations in the Gene LRRK2 Encoding Dardarin (PARK8) Cause Familial Parkinson's Disease: Clinical, Pathological, Olfactory and Functional Imaging and Genetic Data. Brain. PMID: 16272164

[76] Killiany RJ et al. 2000. Use of Structural Magnetic Resonance Imaging to Predict Who Will Get Alzheimer's Disease. Ann. Neurol. PMID: 10762153

[77] Ajami S, Teimouri F. 2015. Features and application of wearable biosensors in medical care. J. Res. Med. Sci. PMID: 26958058

[78] Bradshaw MJ et al. 2017. Wearable Biosensors to Monitor Disability in Multiple Sclerosis. Clinical Practice. PMID: 29185551

[79] Misra S. 2015. Best Bluetooth and Wireless Smartphone Connected Blood Pressure Cuffs. iMedicalApps.

[80] MHEALTH: MOBILEOCT BRINGS CERVICAL CANCER SCREENING, n.d. MedicalExpo. Accessed January 8, 2018.

[81] Metz R. 2015. Smartphone Eye Exam Fits in a Suitcase. MIT Technology Review.

[82] Sima A, Fotooheh T. 2015. Features and Application of Wearable Biosensors in Medical Care. J. Res. Med. Sci.

[83] Ahlqvist T et al. 2010. Road-mapping the Societal Transformation Potential of Social Media. Foresight.

[84] FJ, et al. 2014. Social Media: A Review and Tutorial of Applications in Medicine and Health Care. J. Med. Internet Res. PMID: 24518354

[85] Taglang G and Jackson DB, Use of 'big Data' in Drug Discovery and Clinical Trials, 2016, Gynecologic Oncology, PMID: 27016224

[86] Attai DJ et al. 2016. Social Media in Cancer Care: Highlights, Challenges & Opportunities. Future Oncology. PMID: 27025657

[87] Mayer, B. 2011. Bioinformatics for Omics Data. Methods and Protocols, ISBN: 987-1-61779-027-0

[88] Ibidem

[89] Rughupathi W, Rughupati V. 2014. Big data analytics in healthcare: promise and potential. Health Inf. Sci. Syst. PMID: 25825667

[90] Ibidem

[91] Ibidem

Chapter 3: Ethical and Legal Considerations for Medical Data Reuse

David Henderson, Fabien Richard and Neil Fitch

3.1 Introduction

The secondary use of data in biomedical research is becoming a major theme pertinent to discussions regarding the legal and ethical frameworks that supports scientific data processing. This trend is driven to a large extent by increasing scientific and technological capacities to collect and analyze large scale *"Big"* datasets. The potential for expanding the scientific understanding of the mechanisms of complex diseases is unprecedented with medical records available in electronic format and molecular biomarker assessments increasingly performed for both medical and research purposes. However, these datasets contain sensitive personal information. Electronic health records contain overtly identifiable information regarding patients including names and geographical locations as well as health information that could be used illicitly against not only the individual but also their close relatives. Moreover, high-dimensional molecular profiles, especially genomic sequences, are fundamentally identifiable through software-based comparisons.

The *General Data Protection Regulation* (GDPR)[1] has been in force in the European Union since May 2018 and attempts to define more precisely the scope of personal data: for example, by recognizing pseudonymized data as personal (identifiable) data and including genetic data in the category of 'sensitive data'. The GDPR defines personal data as follows.

'Personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person (GDPR, Article 4, paragraph 1)

The GDPR is pertinent to all European Member States including, at the time of this writing, the United Kingdom (UK) pending the UK's exit from the European Union. The GDPR will be enacted as 'UK GDPR' and remain in force after the Brexit transition period, but data controllers and processors should be aware that it may be subject to revision either as part of the withdrawal negotiations or at a subsequent time point.

Researchers face an ethical dilemma when using medical data. Medical data can benefit the health of patients, both communally and individually, when reused for translational research

studies; however, exposure of these sensitive data may lead to emotional trauma, e.g. embarrassment leading to depression and isolation, and tangible economic loss should health data be used inappropriately to determine employment or insurance eligibility. Therefore, investigators must limit use of these data and take precautions to reduce the possibility of inadvertently disclosing this sensitive information (see for example ref. 2)[2]. Informed consent, in which patients formally agree to allow the use of their data for research purposes, and data anonymization, in which personally identifiable information is removed, by deletion or unrecoverable alteration, from patient datasets are two widely employed methods for reasonably protecting patient privacy while reusing their medical data.

Patients and clinical study participants have often been allowed the option to share certain medical data under carefully developed informed consent clauses approved by internal review boards (groups of people that approve and govern the conduct of clinical studies within an institution). Investigators are able, under circumstances consistent with the patient's consent, to use data for research purposes not conceived of at the time that the patient's data were collected. To protect the individual from the risks of health information reuse their data is often anonymized by removing or altering identifying data elements such that these data elements can no longer reference the patient. However, demographic information such as gender and ethnicity, which are typically important co-variants with respect to exploratory medical research and cannot be removed or anonymized, can lead to identification of patients when considered together with corresponding health measurements and published data such as social media posts. People with rare conditions would, of course, be at higher risk for identification based on their medical information alone. However, high performance statistical and cognitive computing association methods, many of which are discussed in chapters five and seven of this book, place people within the general population at risk of being identified should their medical data be misused by unscrupulous agents or stolen by cyber criminals.

The specific definitions of anonymized data are important. It should be noted that these definitions presented here are those pertinent to the European Union (EU), especially as defined in the GDPR. These definitions are not necessarily concordant with those used in nations outside of the EU, although the concepts that these specific definitions describe will be relevant to medical research regardless of the location of conduct.

'Anonymization' means the processing in such a manner that the personal data can no longer be attributed to a specific data subject (personal data are rendered anonymous).

'Anonymous information' means information that does not relate to an identified or identifiable natural person or personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable (GDPR Recital 26).

'Pseudonymization' means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional

information, provided that such additional information is kept separately and is subject to technical and organizational measures to ensure that the personal data are not attributed to an identified or identifiable natural person (GDPR Article 4, paragraph 5)

Genetic data is specifically defined in the GDPR as a type of personal data.

Since genetic data contains unique information about the data subjects and their blood relatives, complete anonymization may not be technically feasible. GDPR Recital 26, however, states *"To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly. To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments.*" The 'consideration of all objective factors' leads to the concept of 'de facto anonymization', which may be applied to the processing of sensitive data without undue risks for the data subject (Ref. 3 and see section 3.3 below).

'Genetic data' means personal data relating to the inherited or acquired genetic characteristics of a natural person which give unique information about the physiology or the health of that natural person and which result, in particular, from an analysis of a biological sample from the natural person in question (GDPR Article 4, paragraph 13)

Sharing of health data seeks to optimize re-use of available data resources. This reuse reduces the overall costs of conducting research by avoiding duplicated efforts and improving the efficiency and reproducibility of research programs. With the Innovative Medicines Initiative of the EU commission alone valued in the billions of Euro, reuse of data is a compelling investment for taxpayers.

3.2 Data Life Cycle

The concept of a 'Data Life Cycle' (discussed in detail in the following chapter) represents the comprehensive flow of data from creation to destruction including all data manipulations, copies, derivations and system transfers. The Data Lifecyle applies to all information domains, including biomedical datasets.

Source: https://www.nexor.com/white-papers/enabling-secure-information-exchange-in-cloud-environments/

The idea that data may endure, and retain value, beyond the project or study period in which these data were generated, and indeed may outlive the system from which (or, in the case of medical data, the individual from whom) the data were collected, is central to justifying the investment of establishing a project 'Data Life Cycle'. While there is at present no agreed, unified concept as to what constitutes the perfect 'Data Life Cycle' https://www.bloomberg.com/professional/blog/7-phases-of-a-data-life-cycle/), there are, as presented above, at least six distinct steps, or phases, that are essential for successful data management. These steps are carried out by *"data stewards"* who are responsible for the management of data collections in a manner that is client-serving, ethically sound, and legally compliant. Manipulation of data throughout the data lifecycle constitutes an act of data processing. Data processing must be performed in such a manner as to sustain the integrity of data as it progresses through various intermediate states and is written to various storage repositories. Data processing is as defined by the GDPR as follows.

'Processing' means any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction (GDPR, Article 4, paragraph 2)

The person(s) or agency(ies) that perform(s) acts of data processing is/are also explicitly defined.

'Processor' means a natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller (GDPR Article 4, paragraph 7)

Person(s) or agency(ies) responsible for specific datasets are defined.

'Controller' means the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data (GDPR Article 4, paragraph 7)

Under the GDPR, processing of data is defined in exceedingly broad terms such that any agent interacting with personal data, in any way (i.e. at any step of the data lifecycle), are data processors and must be aware of, and comply with, GDPR-pertinent responsibilities. Both processors and controllers share responsibility for proper processing of data relative to the law. Moreover, the GDPR assigns individual rights that were not material to prior EU data protection statutes. As such, legacy data processes that pre-date May-2018 should be carefully reviewed with respect to the additional protections afforded by the GDPR. These individual data protections (defined in Articles 12-23) include.

The right to be informed: Data must be obtained and processed fairly and lawfully. To this end, study participants (also called data subjects) must be fully informed regarding the purposes of a study, how their samples and/or data will be used, the identity of the data controller(s)[4] and how study participants can exercise their rights under the law.

The right of opposition: Study participants have the right to refuse the processing of their data. They also have the right to withdraw their consent for the processing of their personal data while their data are identifiable.

The right of access: Study participants have the right to know what elements of their personal data are, or have been, processed and to access these data while their data remain identifiable.

The right of correction: Study participants have the right to have their personal data corrected if inaccurate or obsolete. Study participants can demand that their data be deleted while their data remain identifiable.

Data confidentiality: The data controller/processor must guarantee the confidentiality of the data and are responsible for implementing appropriate security measures.

Data integrity: The data controller/processor must maintain the integrity of the data (i. e. avoid corruption, accidental loss or destruction).

Proportionality: Personal data cannot be processed for a period longer than necessary for the purposes of the study (i.e. study retention period). Personal data should be removed, destroyed, or made anonymous after that period.

Data transfer: Personal data that have been collected or processed in the European Economic Area (EEA) cannot be transferred to an organization that is based in a country outside the EEA unless:

1. the organization provides equivalent levels of protection for personal data or
2. the country where the organization is located is recognized by the European Commission as providing adequate data protection or
3. unless such transfer is expressly permitted by a data transfer agreement or the informed consent under which the data were collected.

Education: The data processors must be aware of the requirements of the Regulation and must comply with the Regulation and with the principles and requirements of the Framework during the processing of personal data.

The anonymization of personal data: The application of anonymization methods is an act of processing of personal data and, as such, it must follow requirements of the applicable data protection law and, where required, be authorized. Properly anonymized data is not considered personal data under the Regulation.

Data re-identification: Attempts to bypass protection measures to identify a study participant are a violation of the Regulation.

Data confidentiality and integrity principles apply to both personal and anonymized data. Moreover, integrity principles apply to personal data regardless of whether it is maintained privately or disclosed publicly.

All the above principles must be followed regardless of the purposes for which the personal data is processed. This is true regardless of whether data are used for the purposes for which these data were originally collected or for purposes secondary to the original intent[5]. The ramifications for non-compliance may be severe with corporations that generate sizable revenues at risk for up to 4% of their yearly gross revenue should a court of law rule that such defendants have violated the GDPR and elect to assign penalties. Given the substantial financial risk, corporations have attempted to assess their existing information systems relative to the GDPR and remediate noncompliance as deemed necessary. Unfortunately, remediation efforts conducted to date (Feb 2019) which indicate that data processing systems should be reconfigured or replaced may not be justified given that GDPR case law has yet to emerge. However, identifying systems at risk with respect to the GDPR and reassessing information protection and risk management controls applied to these systems are likely valuable activities.

By reassessing information protection controls, including software development lifecycle artifacts such as design documents, organizations can demonstrate that deliberate actions were taken to address key GDPR tenets such as *"privacy by design"* and *"privacy by default"*.

The GDPR states that information systems managing personal data must incorporate data privacy elements while designing the system. As the security model for most systems is delineated early in the design phase, this tenet will most likely be met, at least from the perspective of intent, even if a security measure may fail under certain operational scenarios. Privacy by default is intended to ensure that security measures are active when a system is operating in its baseline configuration. Both tenets can be demonstrated, at least partially, through software lifecycle requirements, design and qualification documentation and may be material to defense with respect to GDPR litigation.

How EU courts will apply the GDPR, including the level of consistency of rulings across member states, is, of course, of great interest and will be closely monitored as case law emerges. To what extent courts will hold data processors and controllers responsible in cases of data exposure, or loss, predicated by the acts of malicious third parties is highly anticipated. Data theft and other related criminal activities can be difficult, if not impossible, to prosecute given the difficulty in identifying perpetrators. Even if perpetrators are confidently identified, jurisdictional impediments may prevent bringing cyber criminals to judicial proceedings. However, data controllers and data processors are culpable for data loss caused by criminal activity even if these controllers and processor were not party to the crime.

It is important to note that the GDPR must often be interpreted in the interdependent legal framework generated by other EU and Member State statutes. An often discussed, and highly pertinent, example is the conflict between an individual's right under the GDPR to have their data deleted (see Recitals 65 and 66, and Article 17 GDPR) versus non-GDPR regulations that mandate long term retention of data and processing of that data for a variety of purposes. This applies for example, to data generated during clinical trials, especially for market authorization of a medication. More generally, erasure may be denied if the data processing is required *"for compliance with a legal obligation which requires processing by Union or Member State law to which the controller is subject or for the performance of a task carried out in the public interest or in the exercise of official authority vested in the Controller"*. Although the GDPR sets out a number of clearly defined examples where the right to erasure applies and, equally, the situations where the data controller can deny such a request: for details, see: https://gdpr.eu/right-to-be-forgotten/. Clarification of these provisions may prompt the legislation of legal exceptions or, if challenged, will require litigation.

3.3 Specific recommendations for maintaining regulatory compliance

Biomedical researchers can either generate/collect and/or process personal data. There are several practical considerations to be addressed with respect to performing these activities in a compliant ethically-responsible manner. The data will, of course, need to be generated by a

legally and ethically conducted clinical trial that has been reviewed and approved by pertinent government authorities and internal review boards. Investigators must ensure that all sites are operating in compliance with national policies regarding data collection and transfer across national borders, these laws vary across nations within the EU (GDPR ch2 article 4 clause j).

Investigators must assess their data collection, storage, and processing systems with regard to the requirements of the GDPR and ensure that their data processing activities comply with the applicable informed consent. Investigators must select an appropriate anonymization method if anonymization will be conducted.

The GDPR does not explicitly define the term anonymization although recital 26 deals with the question of whether a subject is identifiable.

'To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly. To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments'.

The data controller may consider the data to be de-facto anonymised if deanonymisation cannot be ruled out completely but is only possible with an unreasonable effort in terms of time, cost and manpower. De facto anonymization may be achieved through employing several methods that decrease the likelihood of re-identification. Anonymized data is not considered personal data within scope of the GDPR. Unfortunately, since anonymization is not an absolute process and depends on the information content of the dataset in question, there remain risks of potential GDPR non-compliance.

At the time of this writing, the European Medicines Agency and the Heads of Medicine Agencies-led joint *Big Data Task Force*[4] will likely address anonymization with respect to use/reuse of biomedical data, hopefully leading to unambiguous guidance regarding methodologies that, if followed correctly, will yield anonymized datasets that are out of scope of the GDPR.

Investigators must appoint data controllers, establish systems for data processing and carry out risk assessments to determine the appropriate security measures for processing and retaining data. In the best-case scenario, institutions to which investigators belong will provide specific guidance and policies with respect to GDPR conformance. In addition to the rights of study participants noted prior, the GDPR mandates 72-hour notification in cases of data breach, data portability (i.e. the right of the individual to access and share their data which, in the case of clinical trials, may conflict with statues governing trial conduct) and the right of individuals to

demand that Data Protection Officers (DPO) be involved with data processing activities. Services such as breach notification, data portability and the availability of DPOs may be most efficiently provided as institutional services.

Individual investigators would have the responsibility of creating data flows that minimize the processing of personal data, pseudonymizing data as early as possible (e.g. at the location of data generation/collection as is typically done for clinical trials), documenting the data flow plan, performing risk assessment and creating, and improving as necessary, the implementation of security measures (GDPR Recital 78). Completing a *Data Protection Impact Assessment* (GDPR Article 35, Recital 90) that formally documents the risks and mitigation strategies is recommended, at least, for large projects with complicated data flows.

DPO appointment will be mandatory should regular and systematic monitoring of a large study participant population be required (GDPR Article 37). Select data categories, such as data relating to criminal convictions, will require the involvement of a DPO. A DPO that participates in a personal data project must be qualified to serve in the role based on established credentials, such as professional certifications, with respect to their expertise in implementing best practices associated with enabling dataflows that conform to legal constraints. The DPO, regardless of whether the role is fulfilled by an institutional employee or contractor, must be provided with pertinent resources to successfully meet the expectations of their role and are required to maintain their expertise, for example, by remaining professionally certified. An individual serving a project in the role of DPO should report to the senior level manager of the project.

In circumstances in which the data of EU citizens will be processed outside of the EU, an individual must be appointed to oversee and approve such data transfer processes (GDPR Recital 80). Moreover, specific controls must be enacted should data be sent to organizations within a nation that has been identified by the EU as having not legislated adequate data protection statues.

3.4 Data de-identification

Data de-identification is the process of rendering personal data anonymous by removing identifying data elements or replacing these identifying data elements with unique alpha-numeric codes such that knowledge of these codes cannot expose the value of the corresponding original identifiable data element. These deidentification codes allow datasets that were originally cross-referenced by personal identifiers to remain cross-referenced, and thus analysis-ready, following deidentification.

As noted previously, pseudonymized data retains an intermediate reference, or key value, that allows personal identifiers in the original dataset to be discovered indirectly from the corresponding deidentification codes of pseudonymized datasets. If this intermediate key value, also an alpha-numeric code, is destroyed the pseudonymized datasets become

anonymized. The intermediate key value and deidentification codes are typically generated using a well understood hashing algorithm, such as MD5. These hashing algorithms ensure that manufactured codes are unique and, to a pragmatic impossibility, prevent the original identifier from being computationally reconstructed from the intermediate key. To limit the chances of improper re-identification of pseudonymized datasets, the GDPR (Article 4, paragraph 5) mandates that intermediate keys be maintained separately from the identified datasets that these keys reference. Furthermore, the chance for reidentification must be further limited by establishing organizational measures to properly manage re-identification of pseudonymized datasets when warranted. Certain anonymization applications (e.g. DWISE Blur, http://www.dwise.com) can, if desired, generate a *risk of reidentification* probability and ensure that datasets are anonymized to attain a user-defined risk of reidentification threshold.

Data access limitations applied in concert with data deidentification constitute reasonable control measures for clinical data. Data that is distributed to the public domain must, of course, be anonymized.

The following are examples of direct identifiers that must be deidentified to realize an anonymized dataset.

Names of persons or relatives
Addresses (post, email, url, etc.)
Telephone Number
Social Security
Driver's License
Vehicle License plate
Professional Certificates/Licenses
Any Account (e.g. bank account)
Any Record (e.g. medical records)
Photos, Facial and/or Body
Biometric Identifiers (e.g. finger or voice prints)

The following are examples of indirect identifiers that must be removed or transformed to realize an anonymized dataset.

Absolute dates and times (e.g. dates of birth, disease onset, treatment start) can be replaced by relative values such as age, days post baseline visit, etc. If absolute times are required these should be approximated, to the closest hour or time-period (e.g. between 8:00 and 8:30) if possible.

Birth date conversions to age, typically in years, can be specialized for pediatric studies (age in months) and geriatric studies (> certain age if the age itself is rare, e.g. >90). Continuous values can be replaced by ranges, e.g. 10-20, 21-30, etc., if possible.

Addresses can be converted to geographic regions having population sizes of at least 10 million.

Names of study sites should be coded.

Genetic data is a special case. The investigators must determine whether genetic datasets are complete enough to be identifiable. For example, low density genotype profiles (e.g. Taqman Low Density Arrays) having a limited number of genes may not uniquely identify the individual associated with the profile. However, microarray genotyping, whole exome sequencing and whole genome sequencing represent progressively greater risks for reidentification with each of these techniques carrying substantial risk for reidentification should copies of these datasets, or separately generated genetic assessments, be discovered with corresponding personal data. Investigators should reduce the content of these datasets if possible but must otherwise protect against dataset exposure. The potential to computationally obfuscate genetic datasets for use in standard genetic analyses is limited at the time of this writing.

Clearly, some directly or indirectly identifiable data elements may need to be retained to perform analysis (e.g. ethnicity, gender). Investigators will need to deidentify values as best as possible and further rely on limiting exposure.

3.5 Data use consent and ethical considerations

Several constructs have emerged regarding the sharing and reuse of human data. The popular *FAIR* (Findable Accessible, Interoperable and Reusable) data principles, discussed more fully in Chapter 4, promote long term valuation of medical data through management ideals. The *TRUST* principles provide a corresponding ethical framework with respect to data distribution.

Transparency. Data subjects are informed of data users' requests if they wish, and data breaches when required
Reciprocity and reward. The contribution of stakeholders (data subjects, data providers, and data users) is acknowledged or rewarded in a study
Universality. The use of data is open to any registered data users if that use is authorized by a national law and/or a data subject
Security. Data are processed in a controlled environment. Data users and their requested processes are recorded for auditing purposes
Tiered data use. The authorization of data use depends on the data type, the analysis purpose, the data user's profile, the analytical algorithm that a data user wants to use, and the data subject's will.

The GDPR mandates action supporting the *Transparency* principle, specifically with respect to disclosure in the event of a data breach. The GDPR also addresses the *Security* principle by mandating security strategies be included during design and security implementations be

operational by default. *Reciprocity* and *Universality* are reasonable ideals to promote within the scientific community but do not address patient privacy concerns. The *Tiered* data principle would stratify the protections of data based on the applied data summarization methods. For example, individual genotypes clearly carry higher risks with respect to patient identification if disclosed than do genome wide summary statistics, which no longer have an individual affiliation. The former must be managed in compliance with the GDPR, the latter are typically released to the public domain. Although proponents of the TRUST paradigm may envision specific auditing processes with respect to reuse of personal data, the historical control for ensuring data is used in an appropriate manner is via informed consent. Additionally, data use contracts between data controllers and data processors typically forbid redistribution by data processors.

The precise text of informed consent clauses is exceptionally difficult to draft. Base templates of these clauses often need to be modified not only on a per study basis, but often on a per site basis within a single study to accommodate local legal restrictions and preferences of internal review boards. These clauses must declare explicitly the types of research for which reuse of patient data is proposed in a form that can be easily understood by study participants such that an informed decision can be made. Although consents are often formulated to permit potential secondary data use, the breadth of the text must be modified to limit data reuse to, for example, specific biological specimen types and disease-specific research (GDPR Recitals 32, 33).

Given the diversity of consent clauses across, and potentially within, studies it is imperative that consents be reviewed in detail prior to taking actions of the following types.

1. Transfer of a study participant's personal data to a third party. With respect to EU citizens, if the third party is based in a country that is outside the EEA and **not** recognized by European Commission as providing adequate data protection, a controller or processor may transfer personal data only if the controller or processor has provided appropriate safeguards, and on condition that enforceable data subject rights and effective legal remedies for data subjects are available (GDPR Article 46, Recitals 108 and 109).
2. Collection and use of special categories of data. In addition to medical and health related data strata such as genetic and biometric data, demographic data including ethnicity/race, political opinions, religious/philosophical beliefs, trade union membership and sexual orientation should be carefully considered before collecting or disclosing to a third party.

If a consent restricts data use with respect to a certain purpose, there may be allowable exceptions based on local pertinent law. However, it is imperative that data controllers fully understand the applicability of such exemptions before using or sharing data for purposes not authorized by the consent. In the absence of exemptions, reuse of data for purposes restricted by consent will require a new or supplementary consent form to be authorized by the study

participant. There are tools for expediting new or supplementary consent forms provided by certain scientific organizations or large-scale projects such as the *Public Population Project in Genomics* and the *International Policy Interoperability and Data Access Clearinghouse* (P3G, IPAC).

Of course, data controllers and processor must clearly understand how their intended research plans align with pertinent data protection statutes. Establishing that the data of interest are indeed personal in nature is the crucial first step. Anonymization of the study participant's personal data for which the participant has consented for reuse removes the classification of personal data with respect to the GDPR. However, consent with respect to anonymized datasets may still be pertinent given alternative national or organizational criteria under which the researcher must operate.

If the data are determined to be personal in nature, there may be multiple legal and organizational entities that govern the use of data for distinct purposes. Therefore, it is important for data controllers to itemize their research intentions to determine which statues apply to each aspect of their intended use of individual data. For collaborative situations involving sharing of data across multiple data controllers, it is important to formally delineate the controller and processor roles for each collaborative dataset. The data controllers are responsible for determining and governing allowable use by data processors. Data processors are responsible for the technical implementation of the data flow provided such implementations conform to the governance set forth by legal statutes as well as any supplemental instructions mandated by the data controller. European data processors are bound to operate in compliance with the data protection statutes of their country of residence regardless of the country of residence of the associated data controller. Figure 3.1 shows a decision tree that can be used to ensure ethical and legal compliance when granting access for data re-use

Data controllers and processors can of course, at their discretion, mandate and govern processing rules that are more restrictive than those required by applicable law.

With respect to the European Economic Area (EEA), if the data controller is European, their corresponding data processors must comply with the data protections laws of the country in which the data controller is based. If a data controller is not European but uses data processors within the EEA the data processing must operationally comply with the data protection laws of the country in which the data processor is based. European study participants must consent specifically to their data being transferred outside of the EEA should the data controller and processor entities not be based within the EEA and the country in which the controller resides is not recognized by the European Commission as providing adequate data protection.

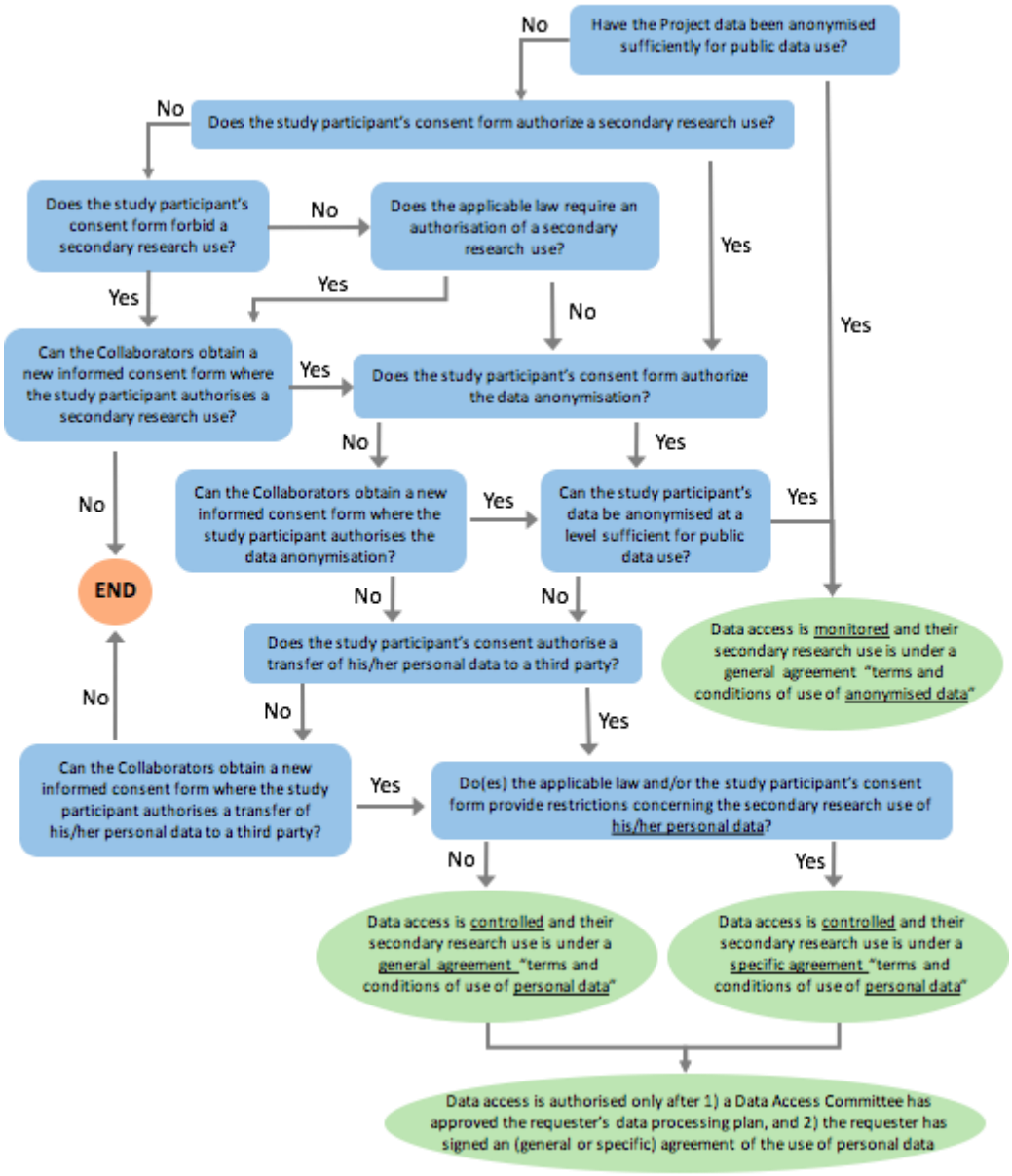


Figure 3.1: Decision tree to ensure legal and ethical compliance when granting data access in biomedical research projects.

3.6 Summary

Data protection is a critical topic with respect to maximizing the value of data collected in the clinic. Patients participate in clinical trials based, at least in part, on the understanding that their data may benefit others, and these benefits are more likely to occur if the data are widely available. Indeed, the *International Committee of Medical Journal Editors* considers responsible sharing of clinical trials data to be a moral obligation. Medical technologies are

greatly facilitating the collection of a wide breadth of complex medical data. Computing and communications technologies are enabling the rapid and broad distribution of these data for use in novel translational research proposals. The potential to substantially better the lives of current and future patients is extraordinary, however, disclosure of sensitive personal information may have traumatic effects on individuals. Investigators, and the scientific associations to which investigators belong, must balance the medical potential of data reuse with the respect for an individual's right to determine when, and for what purpose, their personal data may be (re)-used.

Legal frameworks such as the General Data Protection Regulation provide broad operational and governance constraints to guide data reuse policies. However, such policies do not explicitly inform the development of project-level dataflow implementations. Case law in which these statues are, and will continue to be, tested will slowly emerge and help codify conforming implementations. Nonetheless, research organizations have no choice but to assume compliance risks to progress their scientific endeavors. Adherence to an ethical system based on demonstrating respect for individuals through responsible data sharing is foundational. Such adherence will promote confidence and resolve for researchers operating within unfamiliar, uncertain and sometimes contradictory legal contexts. The moral tenets of the *Global Alliance for Genomics and Health*[5] are representative of the pillars of such an ethical framework.

Respect individuals, families and communities
Advance research and scientific knowledge
Promote health, wellbeing and the fair distribution of benefits
Foster trust, integrity and reciprocity

Chapter 3 References:

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) is a European Union directive adopted in 1995 which regulates the processing of personal data within the European Union. It is an important component of EU privacy and human rights law and became legally binding in all Member States on 25 May 2018 . The web site "Complete guide to GDPR compliance" (https://gdpr.eu/) is a resource for individuals or organisations seeking information on GDPR compliance.

[2] Shabani, M and Borry, P (2018) Rules for processing genetic data for research purposes in view of the new EU General Data Protection Regulation. European Journal of Human Genetics 26, 149–156. https://doi.org/10.1038/s41431-017-0045-7

[3] Butler, J, van Speybroeck,M, Druml, C, et al., SOP HARMONY Anonymization Procedure https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bf59eb2f&appId=PPGMS

[4] HMA/EMA Joint Big Data Steering Group: https://www.hma.eu/509.html

[5] The Global Alliance for Genomics and Health: https://www.ga4gh.org/

Chapter 4: Data Management

Ibrahim Emam

4.1. Translational research data management

Translational research (TR) is often described as a data intensive discipline. The granularity, scale and diversity of data collected and observed during a TR study proves intrinsically challenging to process, analyze, and interpret. Phenotypic data, such as demographics, diagnosis, lab tests, clinical events and medications are collected during clinical studies and hospital encounters. Moreover, the generation of high dimensional molecular profiles including genomics, transcriptomics, proteomics, and metabolomics datasets from physical biospecimen are becoming routine.

Integration and analysis of such diverse high-volume data presents an *informatics* challenge which has led to the emergence of translational bioinformatics as a discipline [1,2]. However, integration and analysis are elements of a broader TR data life cycle[3], an elaborate process to collect, curate, store, integrate, find, retrieve, analyze, and share data (Figure 4.1). Together, these stages form a linear data pipeline which often depends upon communication and feedback between multiple colleagues serving in various roles, including data curators, data managers, clinicians, and bioinformaticians. Conducting research data assets throughout this data pipeline presents a *data management* challenge with respect to improving the efficiency of the research process, data reuse and long-term preservation of data. Thus, effective research data management is critical for enabling TR data analysis and, as such, is an essential cornerstone of successful TR studies.

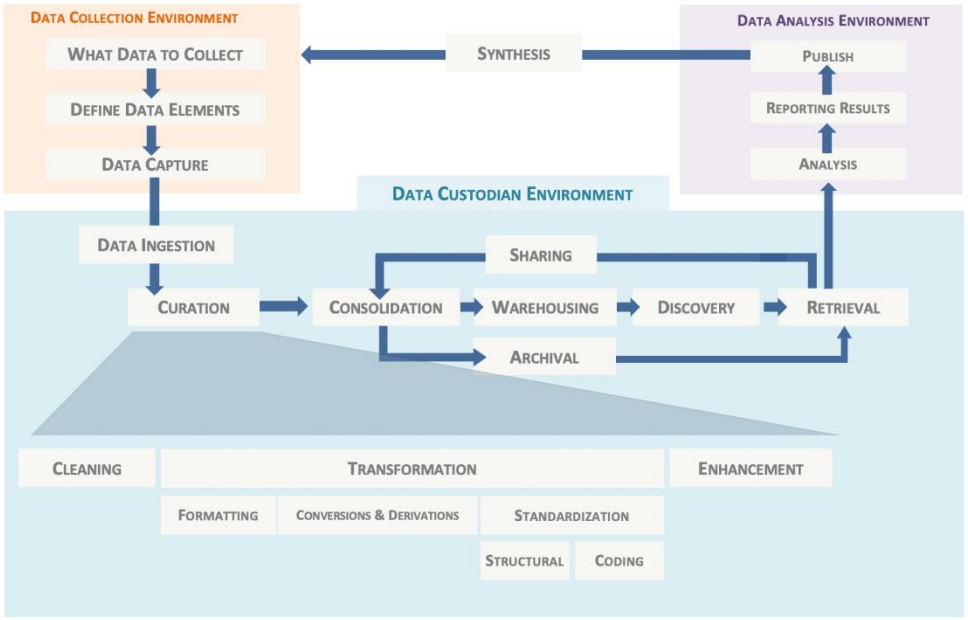


Figure 4.1: Research data life cycle

Over the past few years, translational research platforms have been developed to tackle the challenges of integrating and analyzing combined sets of clinical and omics data. Recent reviews of non-commercial solutions[4,5] demonstrate their relative success in providing;

1. storage and integration of clinical and molecular data
2. analysis context for researchers to investigate, visualize and explore their data
3. data enrichment through integration with external information resources.

These platforms provide solutions for TR studies by enabling integrative data analysis, generation and validation of physiological hypotheses, data exploration, and cohort discovery[6]. However, these platforms focus on supporting the analytical stage of a project ensuring that intended scientific goals are met during study conduct. Other platforms such as dbGap[7] and ImmPort[8] offer an archive solution to preserve data following the termination of the project for which these data were generated. However, such solutions do not play a role while the study is ongoing.

Figure 4.1 illustrates a typical research data life cycle, highlighting in detail the different stages of the pipeline between data collection and data analysis. A data custodian is responsible for the technical control and infrastructure that supports such a pipeline. Many translational

research platforms, while addressing the needs of data storage, integration and analysis, lack data custodianship, which is critical. Establishing a data governance policy, and allocating a data custodian to implement this policy, is essential to promote data as a primary asset of a TR study and to maximize the value of this asset. Diligent data governance not only enhances the efficiency of the research process but also facilitates sharing and re-use of the resulting data asset thereby increasing the return on investment of the study. Thoughtful governance processes also promote conformance to the FAIR data principles[9], discussed later, that seek to create a culture of data reuse within the scientific community. These principles are established by decisions and actions taken at each phase of the data pipeline.

One of the key functions of a data custodian is to implement and enforce data and metadata standards, touched on in chapter two and to be elaborated further in this chapter. Metadata describes data values. For example, a meta data label of "Heart Rate" describes a physical value that is a numeric data type that measures 60 units of beats per minute for the average human being. The data standard for heart rate may be defined to be a whole number, defined in a data system with the label "Heart Rate", that is equal to or greater than zero with the number -9999 representing a missing value. Various stages of the data life cycle in TR research, such as discovering, reusing, sharing, and analyzing data rely on the use of metadata and data standards to be efficient[10]. TR data standards refer to the selection of coding terminologies or ontologies such as SNOMED-CT, LOINC, MedDRA and ICD, against which data is annotated. Standards, however, also pertain to consistent metadata labels, formats and data types as well as common data elements (CDE). Data provenance, recording a chain of custody for data values as these progress through the processing pipeline, is established by capturing events, such as data transfer between systems, by referencing the data value, its corresponding metadata and the event that processed the data value. Several recent reviews suggest that the failure to implement data standards is major challenge of translational bioinformatics and is often due to a lack of understanding regarding how to select and use applicable data standards[11] [12] [13]. In the domain of clinical research, the ***Clinical Data Interchange Standards Consortium*** (CDISC)[14] offers several standards that describe data at different stages of the clinical research pipeline. Similarly, in the domain of molecular assays ('omics), the ***Investigation, Study and Assay*** (ISA) model [15] offers community driven standards for describing assays across different technologies. The eTRIKS standards starter pack[16] provides guidance on the adoption and use of data standards relevant to TR including those for preclinical, clinical and 'omics research.

4.2 Data asset management

An asset is an economic resource that can be controlled or managed and that holds or produces value. Data is widely recognized as the main driver of research, the main currency that feeds the analytical processes and frameworks to derive knowledge from data. However, managing research data as assets is an uncommon undertaking in the scientific community. Viewing data as an asset assigns value to data when it is consumed or applied. The corresponding return on investment, a.k.a. data exploitation, increases as the cost of data planning, acquisition, maintenance, and enablement is minimized.

A data asset management framework ensures that definition, documentation, collection, storage, and processing of data results in consistent, predictable, and appropriate data quality to drive scientific analysis. The FAIR data principles are intended to promote data distribution and use by advocating that research data should be Findable, Accessible, Interoperable, and Reusable (FAIR). However, realizing the FAIR principles in practice for real-world datasets requires data lifecycle management and thoughtful transformation of data into readily useable formats.

This section provides an overview of some of the key foundational ideas and principles needed to establish a data asset management framework. The enterprise data management approach for expediting scientific data analysis, including data classification, transformation and management will first be detailed. The second part of the chapter will present a comprehensive general data model that defines metadata pertinent to translational research and describes the relationships between these metadata. This data model can be used to support a wide breadth of exploratory biomarker projects.

4.2.1 Data categories

Managing data is made more complicated by the fact that there are different types of data that have different life cycle management requirements. Data categories are groupings of data with common characteristics or features and are useful for managing the data because certain data may be treated differently based on their classification. In an enterprise data management system, data can be classified by function (e.g. transactional data, reference data, master data, metadata), by content (e.g. data domains, subject areas), by format or by how and where the data is stored and accessed.

The four most commonly described data categories are described subsequently.

- **Transactional data** describes business events. These are the common transactions that take place as an organization conducts its business. Transactional data always has a time stamp, often is, or includes, numerical data elements and can refer to one or more objects (i.e. combinations of data elements that describe a real word concept). Examples include sales order, purchase order, or a ticket purchase.

- **Master data** describes tangible business concepts upon which business activities are carried out. Master data includes the details (attributes and identifiers) defining core business concepts that are critical to the operation of an organization, such as its customers, products, employees, materials, suppliers, services, shareholders, facilities, equipment, and rules and regulations. Each information domain (e.g. an organization) will use a master data collection suited to its processes.

- **Reference data** are standard, agreed-upon codes that facilitate the use of transactional data within an organization and, as necessary, across collaborating organizations. Reference data management is intended to standardize the codes used across the enterprise to promote data interoperability. Reference data define the set of permissible (domain) values that can be used in master data fields (e.g. 'M', 'F' could be the valid values for a data field describing gender). Domain values that are defined and enforced to ensure data consistency and clarity with respect to the meaning of data values to minimize misinterpretation by data consumers.

- **Metadata** is data that describes or labels transactional and master data. Metadata are used to store, retrieve and interpret data values and are the primary means of organizing information systems. Thoughtful definition and use of metadata promotes data quality and is integral to the creation and management of databases as well as applications that read/write/modify data to/from/within such databases.

This data categorization applies to data associated with scientific research. Translational research generates medically related observations associated with study participants such as a disease status, a treatment regimen or a hereditary trait. These transactions are represented with data values (e.g. “60”) associated with corresponding metadata (e.g. “Heart Rate”). This observation will likely be associated with a master data concept (e.g. “vital signs”) an be expected to conform to established standards (e.g. must be a whole number greater than zero) or entered as -9999, matching a reference data value, should the observation be expected but missing from the data set. All negative numbers other than -9999 would be in violation of the rules pertaining to storing the heart rate observation as data and should, for example, generate an error if such a nonconforming value attempt to be entered into the clinical information system.

Figure 4.2 shows an example of data from a clinical study illustrating the breakdown of the data into the categories discussed above. A study recorded an observation about a patient who had a severe headache as an adverse event on day 6 of the study. The severe headache episode is an example of an *observational* data point. The observation is given context by the master data associated with the study and subject such as the subject identifier, age, sex, the study cohort to which they belong, and the study in which they participated. The study master record contains more information such as the details of the study visit during which this observation was recorded. The string values used to describe the name of an adverse event, e.g. 'Headache',

and the severity of the event, e.g. 'SEVERE', are verified using reference data values that are pre-configured for this study to maintain consistency across observations. These terms are coded and come from standard vocabularies such as the MedDRA dictionary for adverse events and the CDISC SDTM terminology. Finally, the description of the fields that describe the elements of the observation are defined by metadata. Each field has a description. The set of constraints and the rules that specify the reference data to be used for each field (not shown in the figure) are also defined as part of each field's metadata.

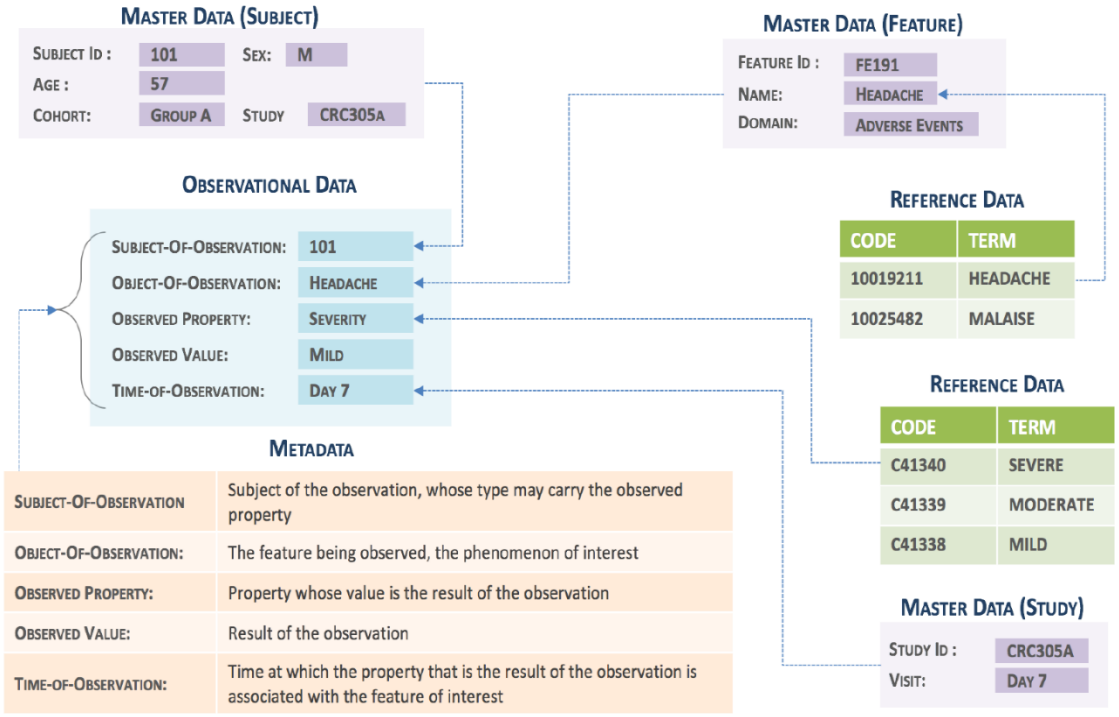


Figure 4.2: Data categories in biomedical data

4.2.2 Data dynamics

Data life cycle

That data has a life cycle has long been recognized in the field of data management. The data life cycle has three general phases:

1. the origination phase during which data are first collected.
2. the active phase during which data are accumulating and changing.
3. the inactive phase during which data are no longer expected to accumulate or change, but data are being applied or used.

English [17] refined these stages and formulated a five-phase data life cycle.

- **Planning phase**: Planning includes identifying data to be collected, defining data elements and developing data models.
- **Acquire phase**: Processes that acquire information are comprised of the business processes and computer applications that "create" data.
- **Maintain phase**: Processes that maintain information are comprised of the business processes and computer applications that "update" data.
- **Apply phase**: the actual use of the data in ways that add value.
- **Dispose phase**: When data is no longer needed it is disposed of or deleted

McGilvray[18] later extended the model by adding a "store and share" phase and naming it the ***POSMAD*** life cycle model, an acronym for Plan, Obtain, Store & Share, Maintain, Apply and Dispose. Data management decisions such as the use of online, near line or offline data storage services can be informed by the lifecycle process.

4.2.3 Data flow

Throughout its life cycle, data may be cleansed, transformed, merged, enhanced or integrated. As data are used or enhanced, new data are often created, so the life cycle has internal iterations. The life cycle model describes phases with logical dependencies, not actual data flows. There can be multiple ways that any piece of data or set of information is obtained, maintained, applied, and disposed. In actuality the data can also be stored in more than one place as data moves through different processes whether during data collection or during data processing. Data flows may go round and round through these phases, e.g. from data maintenance back to data creation and then returning to data maintenance and so on in more cycles. A *data flow* is the movement of data. Data flows are documented by data flow diagrams which provide a way to document the movement of data between data sources and data sinks (the places where data are stored) and the operations conducted on them.

Figure 4.1 shows an example of a data flow in translational research context. In a typical translational research project, datasets are generated from different sources and stored in project-specific operational databases, which are specially tailored for data collection and management of day-to-day operations. Data from these databases are usually exported, cleansed, validated, and merged using various Extract-Transform-Load (ETL) processes to produce a reliable non-redundant collection of curated datasets, parts of which become subject for queries through data marts or data warehouses designed for analytical purposes and specific groups of users.

4.2.4 Data lineage

Data not only has a life cycle, it also has lineage (i.e. a pathway along which it moves from its point of origin to its point of usage, sometimes called the *data chain*). Understanding the data lineage requires documenting the origin of data, as well as their movement between sources

and sinks, and transformation through data processing phases. To effectively manage data through these phases, we will need to identify these states that data sits at between and during these phases. This will help us identify the different states of data assets that will be subject to data management (Figure 4.3).

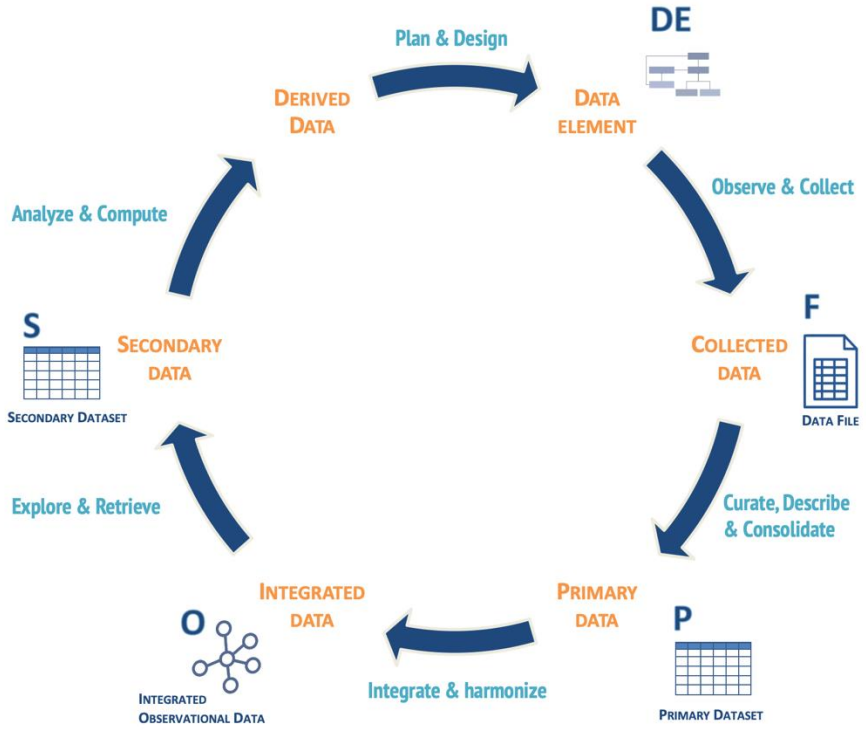


Figure 4.3: Data asset lineage showing the different *states* of data throughout the different phases of the data life cycle

4.3 Identifying life cycle *states* of research data

A ***data model***, a schematic of data elements and their relationships, is created to represent data within an information system. The model is comprised of a set of metadata that describes the transactional data which will be captured by the data flow. Related metadata are organized into master data concepts to better represent the business process. Related master data concepts are in turn referenced to further represent the business process. For example, a clinical study is a master data concept comprised of a study identifier, title and other descriptive data. A study participant is also a master data concept comprised of a site/subject identifier, gender, ethnicity and other descriptive transactional data. However, study and study participant are related in that a study will have zero, one or more study participants. Therefore, the master concept of a study includes the master concept of a study participant and this relationship is expressed in the data model.

The details associated with modeling data within an information system will next be described.

4.3.1 Data elements

The international standard ISO 11179 defines data as "a re-interpretable representation of information in a formalized manner suitable for communication, interpretation or processing". The transactional data elements in a hypothetical research workflow should be identified during the planning and design stage. Research projects in scientific studies start with the formation of research questions about various phenomena under study. These phenomena have observable information that can be represented by data values. For each phenomenon of interest, a researcher must determine what information is needed and how that information might be obtained. The requisite information obtained (usually measured) from the phenomena will be represented as data values. The international standard ISO/IEC 11179 formalizes the association between real-world phenomena of interest, the observation method, and the resulting data value into the concept of a 'data element'. Functionally, a data element provides the meaning behind a measurement by associating its data value (e.g. a specific number (27)) with a metadata descriptor such as Body Mass Index (BMI). As an example, the number 27 is a reasonable value for either BMI or age such that the meaning of the data value cannot be inferred from the value itself. However, when the value is measured (27) it is assigned explicitly to the appropriate metadata (BMI) ensuring that the value will have the correct meaning when entered into the data flow.

As a data building block, the Data Element will play a role in all states of data whether at the planning and design phase, the collection phase or later during the integration and harmonization phase. In the context of research data life cycle management, the *role* of a Data Element is to manage the `meaning' of a data item. This is not the same role provided by concept definitions that are often assembled in data dictionaries or ontologies. The informational value that is embedded by the elements constituting the ISO/IEC 11179 defined Data Element model provide much power and control over the semantics of a single data item rather than providing a simple definition. This is due to the separation of operational meaning from the conceptual meaning which we discuss in more details in section 4.4.1

4.3.2 Created data

During the `collection and acquisition' stage of a data life cycle, decisions about what data to collect culminate in data collection activities. It is during these activities that observable data is created by obtaining *values* for the set of planned and designed *data elements* about the phenomena *of interest* subject to research.

In a research study, there are many different ways that data values are obtained depending on the source, modality and method of acquiring data. For example, in a clinical trial a screening event collects data via a Case Report Form (CRF), in primary care data during a patient visit is recorded in electronic Health Records (eHR), medical reports or in medical images, while in molecular profiling assays, data from patient samples are obtained using instruments and software tools that outputs data in the form of standard or proprietary formatted files.

Data at this stage is sometimes referred to as `raw data' in reference to its processed and curated form later in the analytical phase. However, in reference to its function and purpose during its life cycle, data at this stage exists in a state that can be referred to as *Created Data*. Created data can exist in many different forms and structures. However, the common property that exists for all forms of created data is that they are *optimised for managing the data collection process* within the environment or system generating them. For example, data collection forms are designed in a way that facilitates data recording and database models supporting this activity will also be optimised for data entry. Data in this state is not (and usually should not be) optimised for other data related processes such as sharing, archiving, exploring and analysis.

Therefore, in a research data management context, using data in their `as created' state for any other research activity other than obtaining, creating, and growing the data should be discouraged. In a typical study, raw data will be collected by different people, systems, and organizations that will eventually be transferred to the research information system. To manage these data files as collected data assets, it is important to ensure that for each data asset, the associations between its content and the set of data elements it represents, as well as the context in which it was acquired is maintained as the data transfers to the research information system.

4.3.3 Primary data

One of the main objectives and motivations of the FAIR data management model is to enable and support data re-usability as well as data re-purposing. It is arguable that well annotated and described *processed and derived* analysis-ready data is re-usable data but it is only re-usable for the analysis workflow that it was purposefully processed for. In this case, this data can be used for reproducibility of research results. However, re-purposing data for different analyses and for different research questions than the original data owners proposed would not be possible due to the data's derived and transformed state.

“Primary Data” is one of the managed states of data during its life cycle. Data in this state should exhibit well defined structure, syntax, and semantics that permits the interpretation of data by humans as well as machines. Projects' research workflows often skip over this stage of the data life cycle and fast forwards to using the data in analysis for the purpose of investigating the project's own research questions and producing its own deliverables.

From a FAIR data management perspective, *primary data* are important research data assets that fulfil the purpose of long-term value preservation of data supporting its re-use and re-purposing.

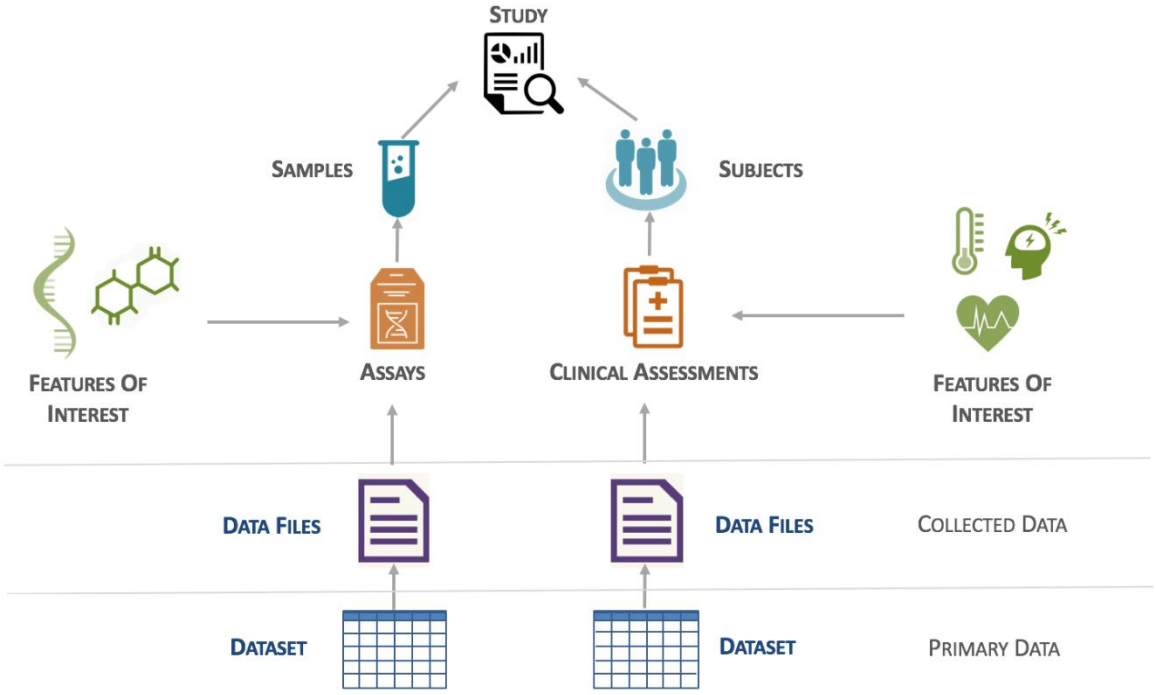


Figure 4.4 Study-centric view of Translational Research Study. Translational research studies generally produce two main categories of data: (1) low dimensional clinical and subject data, and (2) high-dimensional molecular characterization data that result from the different 'omics assay technologies. The first category profiles subjects (patients, animal models, cell culture...etc), while the second profiles the molecular activity in samples extracted from these subjects. Together, data from both categories are organized and defined with respect to a study, which provides the context for generating and analysing such data.

Primary data is study-centric, meaning it reflects the design and planning decisions of the study or project that generated them in order to preserve the *context* and *purpose* for which the data was originally designed and collected. Figure 4.4 illustrates the concept of study-centric data organization. This is important to guarantee the integrity and reliability of these resource in serving reproducible research. Creating and describing primary data becomes the target output of the data curation activities that otherwise are channeled to produce processed and aggregated secondary data that is fit for single-use project-focused analyses.

The structure and the organization of data in the primary state should hence be optimised for:

1. data manipulation, data curation and data quality tasks.
2. preparedness for long term storage with metadata being associated with the dataset itself
3. data sharing and reuse preserving the context within which data was generated at an appropriate level of granularity.

These requirements influence the design of the data model for primary datasets, which is presented in section 4.4.3

4.3.4 Integrated data

In data intensive research projects, data integration is often seen as the primary data processing task and the first go-to step after data collection. The main challenge that drove the development of translational research platforms was to create an environment for researchers to navigate the complex data space spanning clinical and 'omics observational data. This environment would typically involve the development of a common data model to persist data in an integrated form and a set of data manipulation processes that *extract* data from sources, *transform and integrate* extracted data according to the data model, and finally *load* the integrated data into a data storage. Following this ETL process data becomes available for analysis.

Defining the data integration model involves understanding the data integration requirements. These requirements are usually dictated by the individual project objectives or an organization's business needs. For example, the review paper by Canuel et al. discussed in the previous chapter highlights the different integration models offered by the different translational platforms depending on the initial aims of each platform and the community, organisation or project it supports.

There is no doubt data utilization should be driven by data consumers; however, integration requirements defined this way are subjective to the format and queries of the direct data consumers. This may be efficient for a single use of data, but it greatly reduces the potential for data re-use as a result of the transformations and summarization that data is subjected to during this stage. Summarizing data often results in information loss unless a more prohibitive and expensive process is put in place and that is data lineage or provenance management.

From a data life cycle perspective, data in an integrated state should objectively reveal the functional relationships that exist between the various data elements that were chosen and designed during the planning phase. When a researcher plans a study, they will choose to collect data and acquire measurements from different sources and domains, yet they would all collectively contribute to understanding the phenomena under investigation. An observation about the occurrence of headache might be recorded but the pattern of occurrence is also recorded and an exposure to a treatment might be related to this observation. Also, blood test measurements might be recorded for this person and these measurements can reveal a bit more about the level of toxicity caused by the treatment. These implicit relationships should be established in the data model to enable a meaningful and contextual navigation of the observed data.

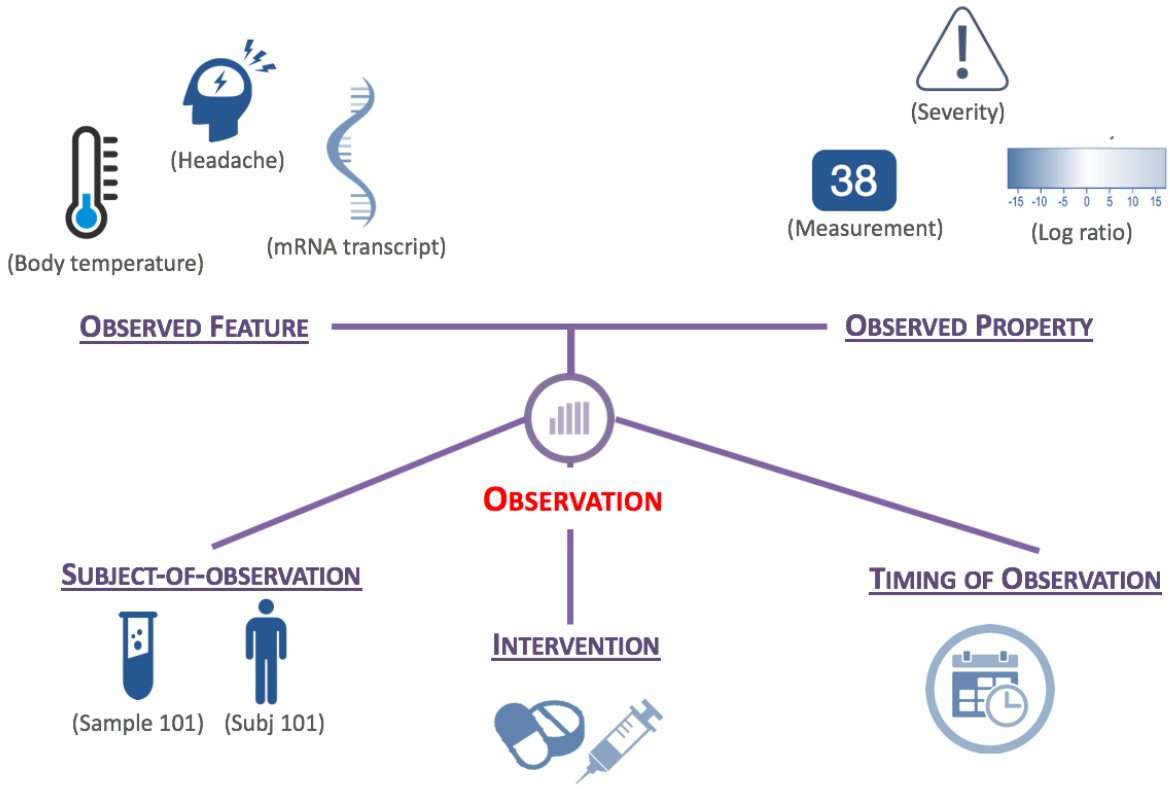


Figure 4.5 Observation-centric view of Integrated Data

Therefore, integrated data should be *observation-centric* as illustrated by Figure 4.5. The data model developed for it should be optimised to reveal and establish all functional relationships that exist amongst all collected and recorded observations. A life cycle management approach should therefore ensure that the metadata that is necessary to build this model is available and well described before the data integration process is executed.

To sum up the states we have discussed so far, data during data collection is optimised for recording and acquiring the observed values for the planned and designed data elements. Following the consolidation process, observations that are collected together are grouped in logically defined and structured primary datasets optimised for data transfer and archival. Similar to the genetic information in a chromosome state, primary data is not organized in way that is optimised for examining and exploring functionally related observations and findings. The purpose of integrated data is to establish and reveal the functional relationships that implicitly exist between the hundreds or sometimes thousands of data elements that were designed to reflect the research requirements of the data owners/producers. Data consumer requirements are reflected onto another distinct state of data, which is discussed in the following section.

4.3.5 Secondary data

Data is valuable only when it is consumed or applied. The ultimate purpose of managing data through the design, acquisition and maintenance phases is to derive the most value from data once it reaches its usage phase. In research, this is realized when data is efficiently and purposefully used by scientists to perform their research analyses and improve their understanding and scientific knowledge accordingly.

The usage stage is therefore characterized by a change in function and purpose of data to reflect the requirements of data consumers. From a data life cycle perspective, this is a shift from the management requirements of a project's data producers and data custodian, which were reflected on the previous stages to those of the data consumers such as analysts and external users. In the Open Archival Information System (OAIS) model, data at this stage is referred to as `The Dissemination Information Packages (DIP)' that are produced in response to queries from data consumers.

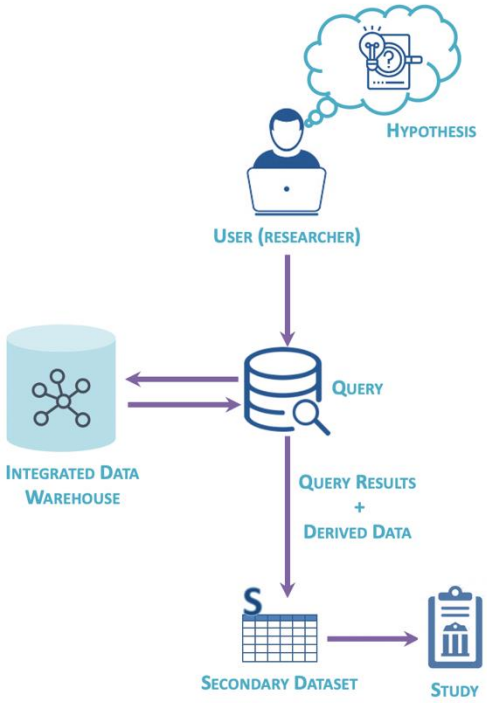


Figure 4.6 Secondary data as a user-centric view of research data

Secondary data essentially represent a different state of data leaving the primary and integrated state to be in a hypothesis-focused user-initiated state that is being prepared for use. This distinction is key to the sustainability of data through re-use and re-purposing because it allows the influences of the two main stakeholders, the producers and the consumers, not to interfere with one another.

Unlike primary data, secondary data is therefore user-centric as depicted by Figure 4.6. This is because users who are interested in analyzing data will almost have to have their own footprint on the retrieved data before they use them in their analyses. This can include running their own queries to create different subsets of data from different projects to compare or summarize results, deriving new data or modifying content of the queried data to suit their needs, or adding their own metadata to describe the content they have retrieved with respect to their hypothesis.

Secondary datasets are similar to *Primary datasets* in the requirement to be structurally and semantically annotated with rich metadata, but they differ in their use and the purpose they serve. Secondary datasets' primary role is to support efficient generation, replication, and review of analysis results. The overall principle in designing secondary datasets and related metadata is that there must be clear and unambiguous communication of their content, source and quality to be able to communicate it with other researchers for sharing and reproducibility.

Data will transition from the *integrated state* to the *secondary data* state through the process of data exploration and retrieval. This is when users use services like querying, retrieval and visualization that operate on the integrated data to select a subset of data elements to create a new dataset which becomes an input to their analysis. A Secondary dataset will contain data from different domains, different studies and most importantly derived data that reflect the needs of a researcher to run a particular analysis. Data at this stage also crosses the boundary between the custodianship environment and the analytical environment.

Secondary datasets should exhibit the following characteristics to fulfill their purpose.

1. Secondary datasets must provide traceability to show the source or derivation of a value or a variable (i.e., the data's lineage or relationship between a transformed value and its pre-transformed source value). The metadata must identify when and how analysis data have been derived or imputed.
2. Secondary datasets must be associated with metadata to facilitate clear and unambiguous reporting.
3. Secondary datasets should have a structure and content that allow statistical analyses to be performed with minimal programming. Such datasets are described as "analysis-ready."

4.4 A metadata management framework for translational research

The primary purpose of the translational research metadata framework (TREMF) is to create a common metadata management framework for translational research studies using existing community driven standards. The TREMF design is influenced by the Meta-Object-Facility (MOF) specification [19], the CDISC standards (SDTM, SDM-XML, PRM) defining data standards and models for clinical data representation in different forms, the ISA-TAB specification [20] for describing molecular-assay metadata, and the Observation pattern developed by Fowler and Odell[21]. The central theme of the TREMF approach to metadata management is extensibility and consistency. The aim is to provide a framework that supports the addition of new data types, while maintaining a standard and domain specific representation of data. To achieve this, the TREMF was designed based on a three-layered architecture as illustrated in Figure 4.4. Each layer is concerned with a different aspect of metadata management. Collectively, they form a comprehensive metadata management framework for the standardization, integration, and harmonization of translational research data. The first layer is the domain model describing the study, its main elements and the relationships between them. It establishes the context for data integration. The second layer is the 'Dataset meta-model': an extensible meta-model based on community standards describing data in the form of a 'dataset' to support standardization of data ingestion. The third layer is the 'Observation meta-model' describing data in the form of an 'observation' to support data harmonization for analytical queries. In the following subsections, we discuss each layer in detail.

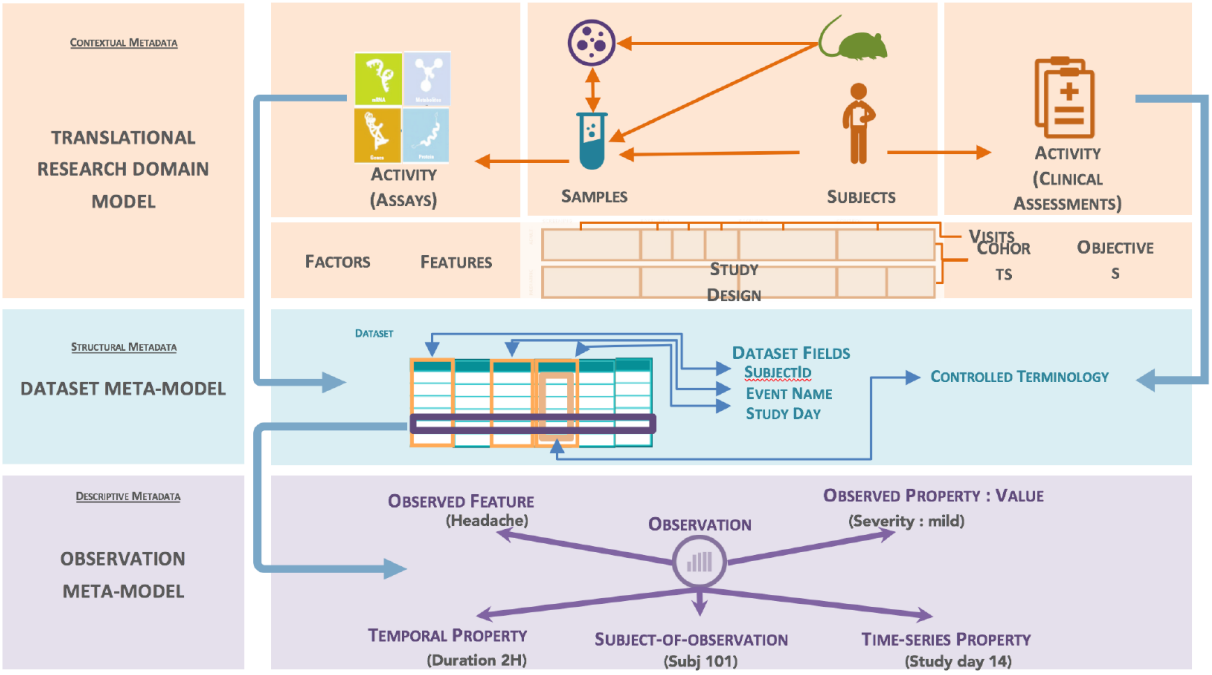


Figure 4.4: Translational Research Metadata Management Framework

4.4.1 Layer one: Data Element

Decisions about what data to collect start with the formation of research questions and culminate in data collection instruments, either user-enterable forms or devices to collect values for specific data elements. Documentation of data element definition is one of the most important artifacts for a study and these definitions are usually presented as data dictionaries. Data dictionaries contain a list of data elements that are labeled (e.g. BMI), a data type (e.g. INTEGER), and valid values for discrete data elements (e.g. =>0 or -9999 (representing a missing value)). Although this information may be sufficient within a single project, each data element should have a conceptual and operational definition to support reuse.

A conceptual definition is what you might find in a dictionary or a thesaurus; it explains what the concept means, describing the general characteristics and key aspects that distinguish it from other related things. The source of these definitions usually comes in the form of reference data that is defined by a source external to the study such as a community-established ontology or terminology.

Operational definitions explain how a concept is measured during the conduct of a research project. These definitions help assure that measurements are taken consistently such that data will be comparable. Additionally, operational definitions document how the data were obtained. As such, operational definitions promote reproducibility.

Together, conceptual and operational definitions relate the features of interest with the study data results.

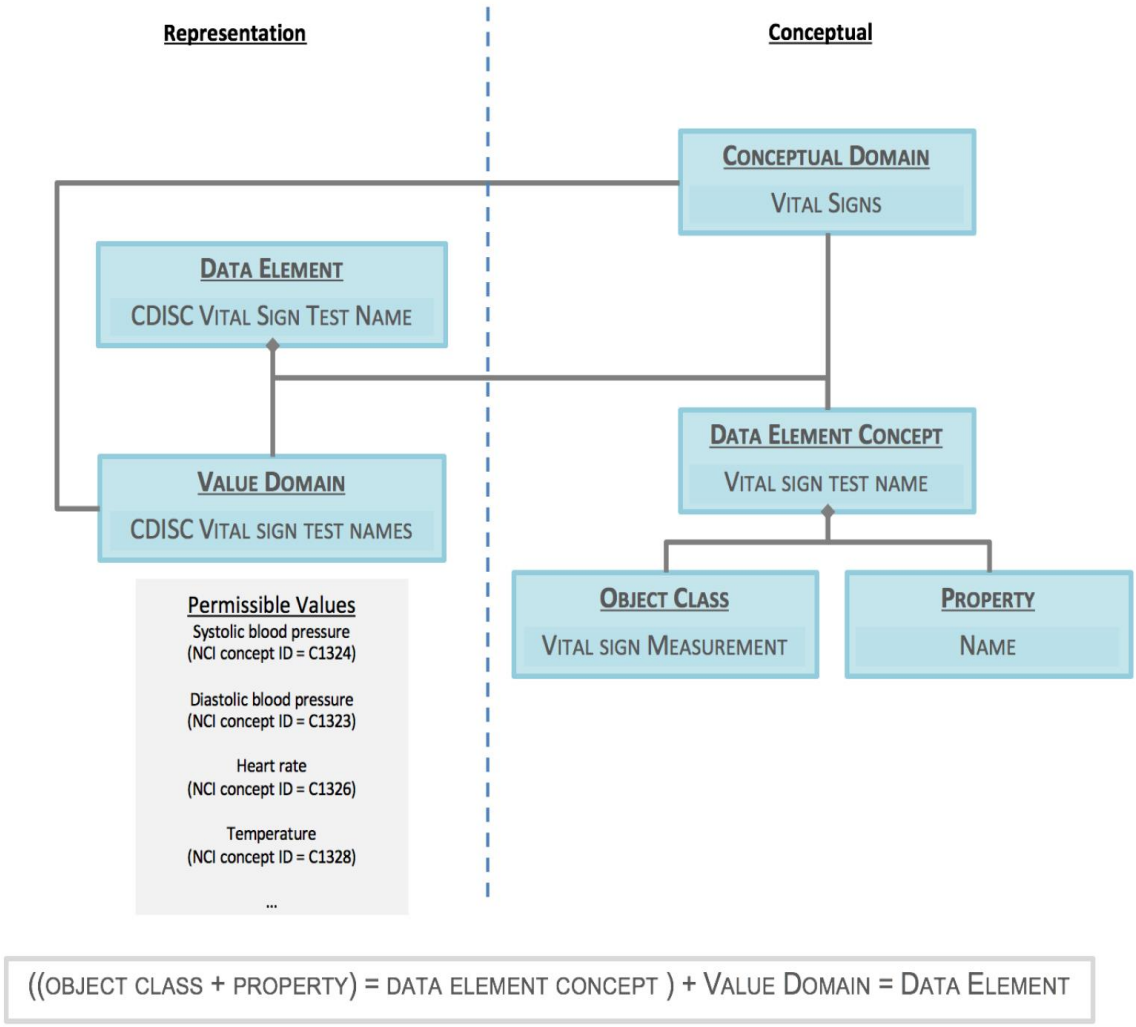


Figure 4.5: Data element conceptual model

In section 4.4.1 we defined a data element as the atomic unit of information exchange and use. It is a construct used to assign meaning to acquired or observed data. The data element carries the definitional information about the data value and is a combination of metadata and reference data to unambiguously describe observed or acquired data. As described above, the ISO 11179 model is a semantic model that formalizes and standardizes metadata necessary to describe data elements such that they can be documented managed and shared (Figure 4.6). This section covers the concepts of metadata pertaining to data elements necessary to completely specify a data element. The ISO 11179 is a framework for the specification and standardization of data elements describes data elements in a four-part framework. The four parts are: a conceptual domain (CD), a data element concept (DEC), a data element (DE), and a value domain (VD) (Figure 4.5).

For example, if the data element was the 'patient sex', the value domain would include values for all sex values (Male, Female, Unknown). Thus, the value domain contains the valid values for the data element; however, there can be more than one way of representing these values.

Some studies might use (m,f,u) or (0,1,2). ISO 11179 standard handles this situation of multiple possible answer representations through having a data element (DE) and a data element concept (DEC). A DEC in this example can be the 'sex', which there are multiple representations of permissible values (value domains) that can specify the sex type. The pairing up of the DEC and one of these value domains creates a data element.

For all data elements

Conceptual definition
Operational definition
CRF question (if applicable)
Unique identifier
Data element name
Data type
Mapping from the data element to the structure in which the data are stored
Documentation of change to data definition

For discrete data elements

List of valid values
Conceptual definition for each valid value
Operational definition for each valid value

For continuous data elements

Unit of measure
Statement of the range or limits on valid values
Precision

For derived data elements

Algorithm used for calculation or a reference to one

Figure 4.6: Data element- level metadata for research

4.4.2 Layer two: Observation Data Model

The challenges in approaching a diverse observational dataset, a user needs both to simply assess the breadth of content (i.e. set of metadata values) available and systematically navigate to the detailed observational data values. This challenge is intensified when attempting to integrate clinical and molecular observations across 'omics technologies. There is no common standard for describing molecular observational data from omics assays such as microarray assay, sequencing, spectrometry etc. The ISA-TAB standard describes experimental metadata as well as the samples and the assays that were used to generate the data. However, omics instruments often use proprietary data formats and structures. The most common structure is a simple data matrix, where the columns represent the features being observed, the rows represent the samples assayed and the cells representing the observed values. Data cubes are usually defined when multiple observation values and multiple dimensions are measured per observation.

Managing data at this level requires the identification of a set of semantically labelled data elements and the associations between them to act as anchor points to which data at this level can be integrated and harmonized across heterogenous biomedical datasets. This is a crucial

step in enabling researchers to identify potential (hidden) relationships, patterns, associations, and correlations that exist in these large heterogeneous datasets. For example, a researcher might want to test the correlation between the severity of an adverse event with the increase of a lab test urine measurement and with the treatment of a vaccine. Another common use case is to compare data from one clinical program with another program in the same therapeutic class. This use case requires data harmonization across programs to support data comparisons. How can we define an observation-level data model that can enable researchers to navigate this huge data spectrum with a consistent and systematic approach?

The main purpose is to provide a standard for structuring observed data and related contextual data into a semantically defined common data model. The key idea in this model is the underlying presumption that an 'Observation' cannot be modelled as a simple fact-attribute concept, but rather it consists of discrete pieces of information as designed by the researcher to collect the desired information about the phenomenon of interest. To get an idea about how this might be achieved, the underlying concept of the data element will be extended as discussed in section 4.4.1. A data element defines data in terms of observability: the phenomena of interest, the observational information to be observed/measured, and the resulting data value of the observations/measurement. Observations and measurements in clinical or molecular data can be conceptualized as a group of multiple data elements put together in a semantically defined vector providing the desired contextual information necessary for the interpretation of a phenomenon of interest. Observational data is the equivalent of transactional data associating a number of master entities including an observation feature, a subject or a sample each with their own set of characteristics and collectively they form the semantics and context for recording a data value for such an observation. "Subject 101 had mild nausea starting on Study Day 6" is an observation that involves a subject (101), an observable event (nausea), an observed quality of this event (mild), and a point in time qualifier (day 6). Each of the above constructs of an observation is a *data element* associating a phenomenon of interest (nausea), with an observable quality (subject, severity, timepoint) and the observed value (101, mild, day 6). Using a transcriptomic assay example, "The measured log intensity of expression of gene wnt11 in sample A is 7.5" is an observation that involves an observable feature (wnt11), an observed quality (transcription expression log intensity), a sample (A) with the observed value (7.5).

Observation Conceptual Model

Despite the vast complexity and diversity of possible biomedical observations, all such information can, in principle, be described via a straightforward set of five underlying concepts. These are specified by the observation model that describes the principle data elements and their relationships that make up an observation. The observational data model captures the core information needed for describing scientific observations and provides a common language that can be used to harmonize representation and supporting software implementations. It also supports integration, interoperability, query and analysis. The model defines and interrelates the conceptual subcomponents that are intrinsic to any and all kinds of “observation”. Figure 4.7 illustrates the different constructs of an observation using a set of semantically labelled data

elements. This model deconstructs an instance of an observation into the following construct data elements:

- Object-of-observation: the feature being observed, whether in a clinical or molecular setup; e.g., weight, albumin, headache, TP53, CD40 - described by a topic descriptor
- Subject-of-observation: the entity upon which the observation is being observed
- Observed property: qualitative or quantitative property of the observed feature being observed or measured; e.g. count, result of test, severity of headache, amount of dosage
- Temporal properties: timing attributes that are not longitudinal such as time of collection, duration, start of event, interval …etc.
- Time-series properties: properties that cause the repetition of the same observation over time, resulting in a longitudinal observation; e.g. visit, planned study day, time point.

In addition to the above attributes of the observation, supplementary data is often collected to provide more context and granularity to the observation with attributes about the observed-feature, or about the subject-of-observation. These attributes are linked via the master entities described in the TR model discussed in the following section.

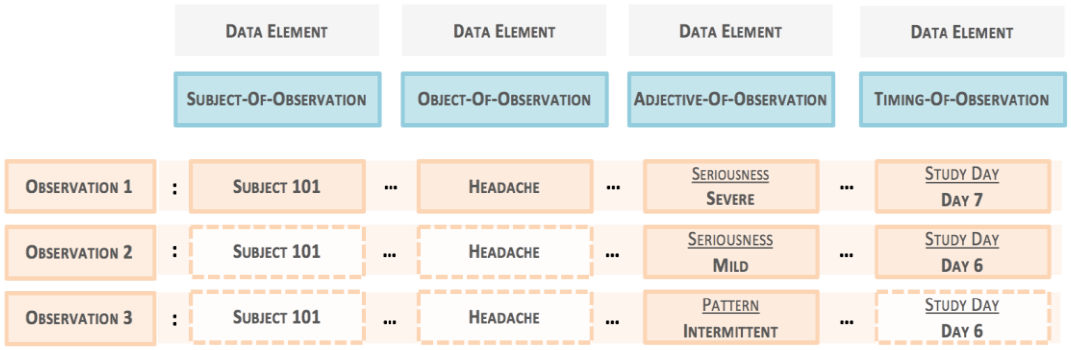


Figure 4.7: Observation conceptual model

These concepts are applicable to high dimensional molecular observations as well. These assays generate a substantial set of observable features such as transcripts, gene sequences, gene names, proteins and metabolites. Observations at the molecular level also follow the same semantics implied by the observation model. The object-of-observation in this case is the feature being observed. For a microarray experiment, the object of observation is the transcript; for flow cytometry, it is the cell; for a proteomics assay, it is the polypeptide or protein. In each of these assays a set of quantities are being measured. For example, the raw or log ratio of the intensity, count or concentration of a metabolite or a cell type. These assays measure these observations for a set of samples that could then be described by a set of attributes giving more granularity to the observation as well as adding a time dimension if the assay design involved sampling at different timepoints.

4.4.3 Layer three: Dataset meta-model

The dataset meta-model defines the requirements and the properties of the primary data asset. As previously explained in section 4.3, primary data assets provide a standardized way to group and organizes observations as defined by the observation model (section 4.4.2) into structurally and semantically annotated datasets. The primary dataset facilitates data curation, data integration and data reuse. A primary dataset follows the principles of a *tidy* dataset. Although not a standard, these principles provide a framework for thinking about the organization of data within a dataset. Developed by Hadley Wickham, this framework encompasses data, tools and workflows. In this section, the three characteristics of a tidy dataset will be briefly described, then based on these principles the framework's dataset model will be defined.

Tidy data

There are many ways to structure data in a dataset. Most research datasets are rectangular tables made up of rows and columns. Table 4.1 gives an example of a dataset about vital sign measurements conducted during a clinical trial over two visits. The dataset contains four rows and eight columns illustrating a common structure usually referred to as the horizontal (or wide) format. Table 4.2 shows the same data as Table 4.1 but using the vertical (or long) format. The inconsistency of those two dataset representations shows that the structural metadata of a dataset is not enough to describe the underlying semantics or meaning of the values displayed in the table.

Table 4.1: Tabulated data in wide format

SubjId	Cohort	sysbp_sc	diabp_sc	heart_rate_sc	sysbp_fu	diabp_fu	heart_rate_fu
s-01	cohort_A	100	65	89	95	63	70
s-02	cohort_A	134	77	60	120	71	66
s-03	cohort_A	182	122	110	190	100	100
s-04	cohort_A	112	81	90	120	98	78
s-05	cohort_A	127	74	70	135	75	72

Table 4.2: Tabulated data in long format

subjId	Cohort	sysbp	diabp	heart_rate	Visit
s_01	cohort A	100	65	76	Screening
s_02	cohort A	134	77	60	Screening
s_03	cohort B	182	112	96	Screening
s_04	cohort B	112	81	38	Screening
s_01	cohort A	128	87	62	Follow up
s_02	cohort A	125	80	68	Follow up
s_03	cohort B	132	80	71	Follow up
s_04	cohort B	132	87	84	Follow up

Semantically, a dataset is a collection of *values*, either quantitative or qualitative. Every value belongs to a *variable* and an *observation*. A variable contains all values that measure the same underlying attribute (like height, temperature, duration) across subjects. An observation contains all values measured on the same subject-matter (like a subject, or a study or a sample) across attributes.

Contextually, a dataset is a collection of related observations. In any given research study there are usually multiple *domains* of observations. For example, in a clinical trial of a new vaccine we might have three observational domains: demographic data collected from each person (age, sex, race), medical data collected from each subject on each day (adverse events, laboratory tests), and molecular profiling data about each subject's genetic activity (gene expression, protein profiling, etc.).

For a given dataset, it is usually easy to figure out what are observations and what are variables, but it is quite difficult to precisely define variables and observations in general. Different experimental designs will most certainly lead to different usages of variables vs observations. However, a common framework can be employed to guide the process of deciding which attributes are observations and which are variables when it comes to designing a dataset. The concept of a *tidy dataset* is to provide a standardized way to link the *structure* of a dataset (the layout) with its *semantics* (the meaning of its content) to make the job of storing and accessing the data easier. Three fundamental principles make a tidy dataset:

- Each variable forms a column
- Each observation forms a row
- Each type of observational unit (domain) forms a table

Table 4.3 is a tidy version of Table 4.2. Each row represents a single observation, the result of one lab test of one subject, and each column is a variable. Tidy data principles try to guard against the most common problems that lead to messy and difficult to manipulate datasets. These problems are:

1. Column headers are values, not variable names

2. Multiple variables are stored in one column
3. Variables are stored in both rows and columns
4. Multiple types of observational units are stored in the same table
5. A single observational unit is stored in multiple tables.

Dealing with these problems makes most of the data curation tasks that will be discussed subsequently.

Table 4.3: Tabulated data in tidy format

subjId	Cohort	Test	Measurement	Visit
s_01	cohort A	Systolic Blood Pressure	100	Screening
s_02	cohort A	Systolic Blood Pressure	134	Screening
s_03	cohort B	Systolic Blood Pressure	182	Screening
s_01	cohort A	Systolic Blood Pressure	128	Follow up
s_02	cohort A	Systolic Blood Pressure	125	Follow up
s_03	cohort B	Systolic Blood Pressure	132	Follow up
s_01	cohort A	Diastolic Blood Pressure	65	Screening
s_02	cohort A	Diastolic Blood Pressure	77	Screening
s_03	cohort B	Diastolic Blood Pressure	112	Screening
s_01	cohort A	Diastolic Blood Pressure	87	Follow up
s_02	cohort A	Diastolic Blood Pressure	80	Follow up
s_03	cohort B	Diastolic Blood Pressure	80	Follow up
s_01	cohort A	hear rate	76	Screening
s_02	cohort A	hear rate	60	Screening
s_03	cohort B	hear rate	96	Screening
s_01	cohort A	hear rate	62	Follow up
s_02	cohort A	hear rate	68	Follow up
s_03	cohort B	hear rate	71	Follow up

Why tidy?

In TREMF, primary data assets serve three data management functions: data curation, data integration, and data reuse; hence, their design should be optimized to serve these functions. Tidy dataset principles were chosen as the basis for defining this class of data assets as their design makes it easier for a curator or a computer program to target or extract variables that describe the same underlying attribute for all observations in a dataset. For example, integrating data from two datasets structured like Table 4.3 would require less work for a database loader—using simpler annotations—to identify the columns containing 'visit name' or 'test measurement' for each observation than if both datasets followed Table 4.1 or Table 4.2 designs. The latter would require reading values with the same semantic meaning from multiple columns. Furthermore, in case of a data curation task, a common data transformation or validation task would make sure that all values of an attribute (e.g. vital sign test names) use a common terminology. Defining rules on tidy dataset columns would greatly facilitate validating the contents of a dataset or transforming values of an attribute from one dictionary

to another. Metadata constraints must be employed when transforming data values into and out of the standardized format to prevent inappropriate type conversions.

An 'observation' is defined as a vector of related variables or qualifiers. Each variable/qualifier describes a specific aspect of a measured or collected observation. Principles of the tidy dataset are well suited for this vector model of an observation because its layout ensures that values of different variables from the same observation are always paired and exist on the same row. Each observation vector corresponds to a row in a dataset while each column corresponds to one type of the six qualifiers making up the observation. Table 4.4 shows an example of dataset for observations. One row describes the full observation (Subject 101 has a Severe Headache on Day 6 of the study). Each column carries one value for each of the observation attributes: subject-of-observation (subject 101), feature-of-observation (headache), the feature observed property (severity) and the time-of-observation (day 6).

Table 4.4: An example of a tidy dataset for observation data

Subject	Feature	Severity	Study day
Subject 101	Headache	Severe	6
Subject 101	Nausea	Mild	6
Subject 101	Headache	Severe	7
Subject 101	Headache	Mild	8
Subject 102	Vomiting	Mild	5
Subject 103	Headache	Mild	6
Subject 104	Headache	Severe	7

Standards compliance

In the context of data sharing and secondary use, standards-based data transfer becomes key for wider adoption and outreach to the community. The third main function that a primary data asset serves is to support data transfer. This applies to both the data ingestion phase (importing data into the data management environment) and the data sharing phase (transfer of data from the initial data management environment to other data management environments). To achieve seamless data sharing, the data management framework must use established community standards and natively support datasets that are structured and formatted to confirm with these standards. Moreover, the data management framework may need to conform to multiple standards to support a wide variety of data types and broadly interoperate with a diversity of collaborating systems. The main object of the framework presented in this chapter is to incorporate a generalized model that is compliant with at least two of the most relevant standard

data exchange formats serving the translational research domain: ***CDISC-SDTM*** for tabulation of clinical data and the ***ISA-TAB*** standard for sample and assay metadata tabulation. Although both standards do not comprehensively cover all translational data elements, they represent the most commonly generated data elements in translational research studies. It is important to reiterate that layer two of the framework is concerned with the structural metadata that is used to organize observational data into consistently annotated datasets. Standards pertinent to this layer only apply to data organization.

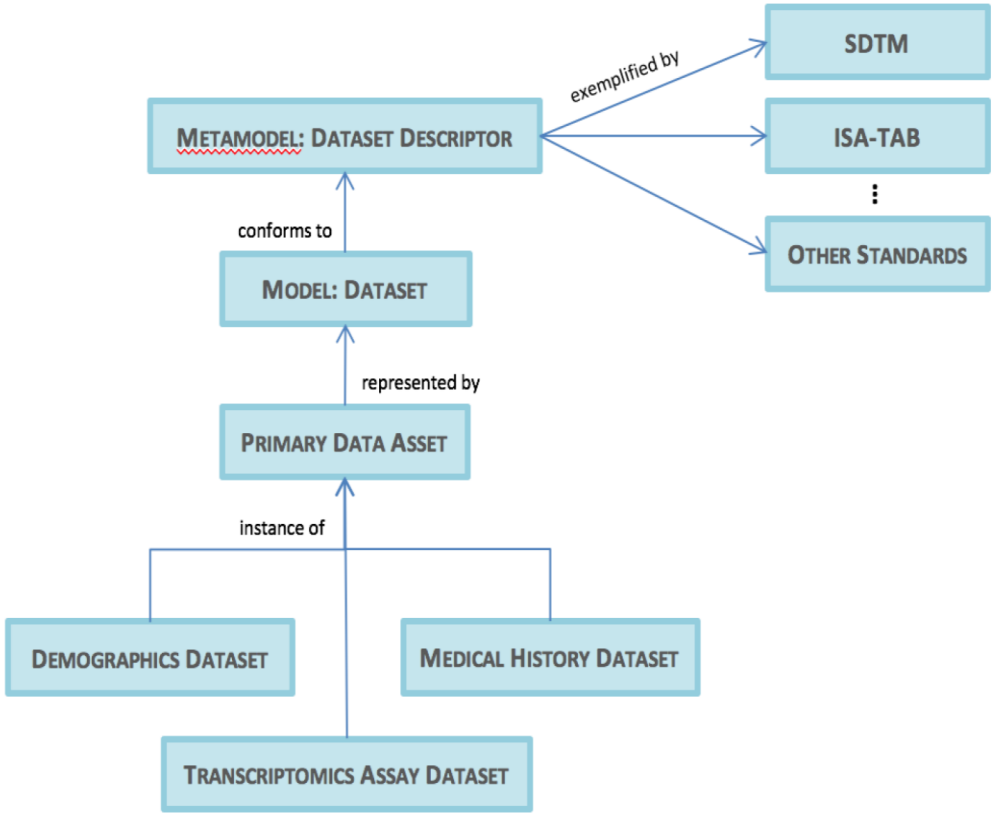


Figure 4.8: Dataset conceptual meta-model

Based on the requirements described above, the framework's second layer is a meta-model based on a common set of basic concepts adopted from the principles of tidy data and implemented in community data standards such as the CDISC SDTM and the ISA-TAB. Figure 4.8 describes the conceptual model of TREMF layer 2. A demographics dataset, a medical history dataset, and a transcriptomic assay dataset are examples of a primary data asset. These data assets are represented by a ***dataset model***. Each dataset model conforms to a meta-model described by a ***dataset descriptor***. A metamodel defines the meta-data necessary to describe the structure, semantics and constraints of a dataset model. A dataset descriptor is exemplified by community standards which provide standard dataset templates. These

templates can be readily used to define standard dataset models. In the following section, we will define the dataset meta-model.

In the simple semantics of a Tidy dataset columns should represent measured variables in the dataset while rows should represent instances of observations being measured against the set of attributes denoted by these variables. To define a tidy dataset for observations, we need to extend these simple rules to be able to describe the *semantics* of the variables (columns) and their relationship to each other in describing the observation vector as described by the observation data model. These extensions also need to be compliant with the supported data standards. Each dataset is described by a 'dataset descriptor', which defines its domain, the structure of the dataset, the syntax and semantics of its fields, any enforced controlled vocabularies, and validation rules.

The meta-model defines two meta-classes: the dataset, which is composed of a set of fields. Each row in a dataset is an instance of the subject matter which is being described by a series of named fields. Each field, which normally corresponds to a column in a dataset, has a role to describe a certain aspect of the row entity in this dataset. A role describes the syntax and semantics of information conveyed by the field. A dataset is assumed to hold data about one subject matter (e.g. subject, study, sample or assay feature) and it is set to belong to a 'domain' that describes the context for the data reported in that dataset (e.g. laboratory test results, demographics, microarray sample metadata, or flow cytometry processed data).

4.4.4 Layer four: Domain model

As defined earlier, master data describe domain entities that provide context for the observed data in the form of common concepts that relate to all translational research studies. The fourth layer of the metadata framework comprises the TR domain model, which describes the core elements of a translational research study and their interrelationships. The domain model provides the contextual study information within which the observational data was collected.

The master data of any translational research arguably comprise a minimal set of common entities that will exist in this domain regardless of the focus of research. This domain model is the contextual metadata that relates data (layers one, two and three) to the master entities common to the translational research domain.

The Entity-Attribute-Value approach to data modelling is applied to most data integration solutions, mainly to minimize the cost of maintenance should data attributes change or be added over time. This reasoning is valid to some extent—building a flexible schema for easily integrating new data elements is justified reasoning; however, active data integration is only part of the data lifecycle. Data processing, storage, archiving, sharing, and repurposing are all phases that require managing contextual information consistently and maintaining explicitly defining relationships between entities of interest. Even cross-study data integration requires a minimal set of contextual metadata about the study design, the study activities conducted and

the relationships between the subjects and their corresponding biological samples from which molecular data is generated. Relational models, of the type championed in this chapter, enable utilization of various abstraction levels that are important in integrating clinical and molecular data. This model is optimized for persisting master data objects. The TR domain model maintains a consistent representation of TR studies. This consistent representation enables cross-study data integration by establishing the generalized study context and relating the observed data to this context (Figure 4.9).

Translational research studies generally produce two main categories of data types:

1. Low dimensional clinical and subject data
2. High dimensional molecular characterizations generated by 'omics technologies

The first category profiles subjects (patients, animal models, cell culture, etc.), while the second profiles the molecular activity within samples extracted from patients. Together, data from both categories are organized and defined with respect to a study, which provides the context for generating and analyzing such data. The foremost requirement is to establish the relationship between the data associated with the clinical subject and the corresponding molecular data associated with the subject's samples. Compliance with established TR data standards is, as stated prior, of critical importance.

A hybrid model based on both the CDISC Study Design Model (SDM) and the ISA data model is presented to provide standards for clinical assessments, biomarker/omics assays and reporting. Data generated from an activity is represented as a 'dataset' in the hybrid model, which is defined by the second layer of the TREMF. Adhering to the TR domain model maintains a consistent representation of TR studies and enables cross-study data integration by establishing the study context and relating the observed data to it.

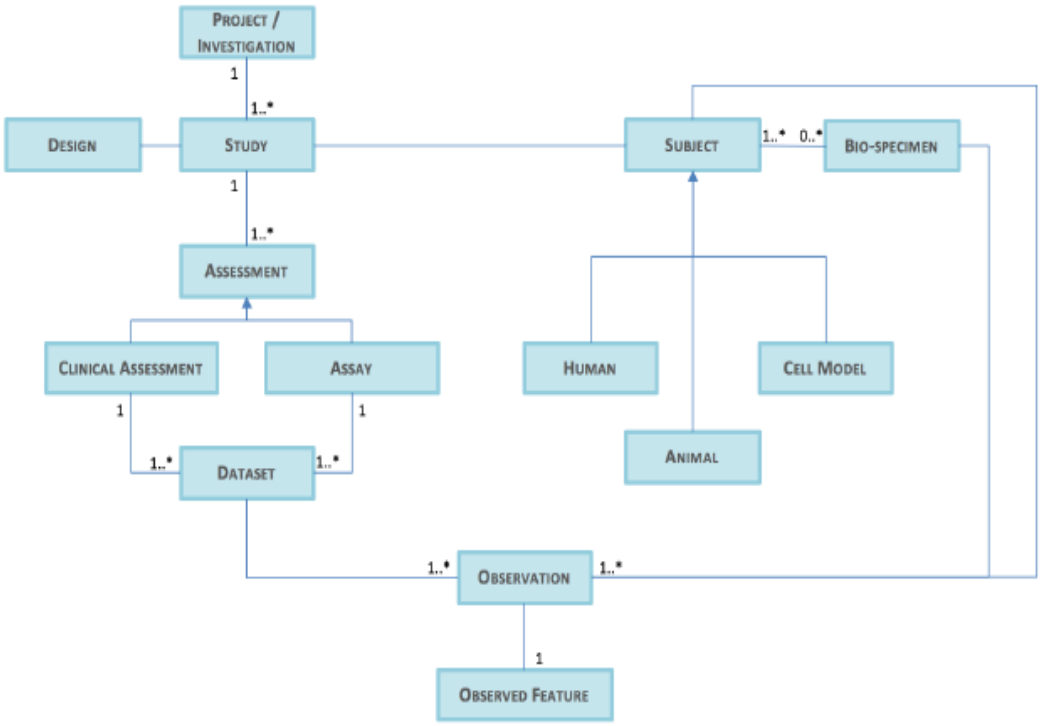


Figure 4.9: TR Domain model

4.5 Building a translational research data management platform

Figure 4.10 illustrates the application of the TREMF approach to manage the different stages of the data pipeline from data ingestion, transformation, exploration, integration, analysis and publication/reporting. The platform maintains the lineage across the various datasets as these are created and transformed throughout the data flow. In the following sections, we present in details the platform's modules and features as illustrated in Figure 4.11. The system architecture and implementation are provided as supplemental artifacts.

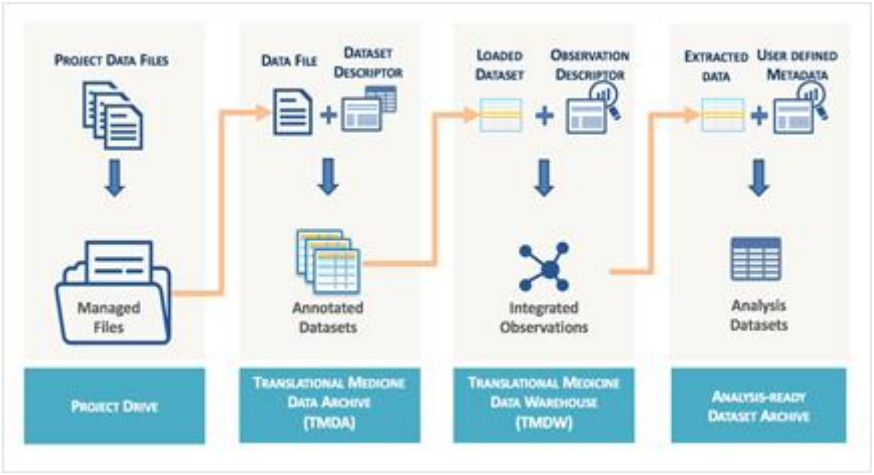


Figure 4.10: EHS application of the translational research metadata framework to manage the data pipeline from the state of files to annotated and integrated analysis-ready datasets.

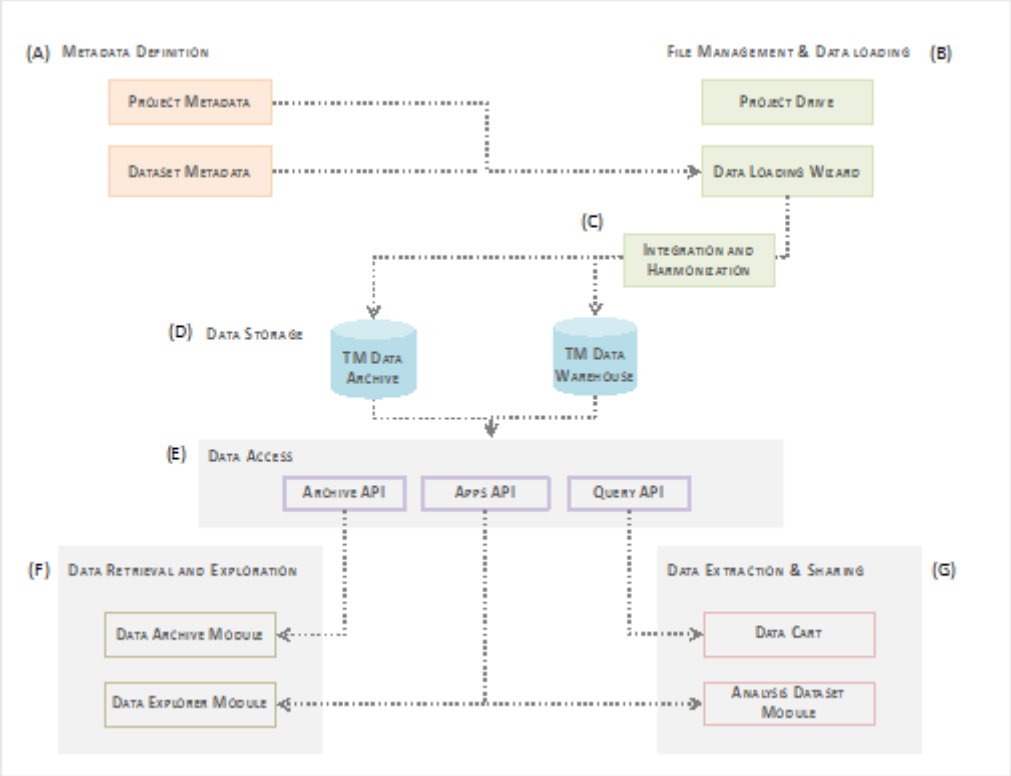


Figure 4.11: EHS modules

4.5.1 Metadata Definition

As illustrated by Figure 4.1, a project's data life cycle starts by identifying and describing the project and the data to be collected (i.e. defining the metadata). Metadata is usually assigned by Electronic Data Capture (EDC) tools, such as OpenClinica[22] or RedCap[23], as new data is collected. However, metadata is crucial for managing data throughout the data life-cycle, therefore, it is essential to incorporate metadata into the data custodianship environment. Data may transform in structure throughout the data flow to become readily useful by data consumers; however, the metadata, or certainly the meaning implied by the metadata, will remain consistent during transformation. The metadata definition module (Figure 4.11a) offers data managers a set of features via a simple and intuitive dashboard to define and manage metadata. This includes defining the research project metadata for TREMF layer 1 (Figure 4.2a) and experimental metadata data for TREMF layer 2.

4.5.1.1 Defining project metadata

Setting up a project from the data manager dashboard is the entry point into the system. A project can be a single study, a multi-study (planned related studies), or a meta-study (unrelated studies). The 'studies panel' enables managers to enter information about each study within a project such as study design, eligibility criteria, objectives and other study metadata elements

via a web form compliant with the CDISC Protocol Representation Model (PRM) and CDISC Study Design Model (SDM). The 'activity panel' allows the data manager to create and manage the project's planned clinical activities, molecular assays and their associated dataset descriptors, while the 'Members and Users' panel is used to manage user roles and their data access rights.

4.5.1.2 Defining dataset metadata

A dataset is a pointer to one or more data files. Each dataset is associated with a *dataset descriptor* comprising its metadata describing the data structure and meanings of each column. Based on the dataset meta-model, standard compliant dataset descriptor templates were created to cover all CDISC SDTM domains (findings, events and interventions) and preloaded them into the database. Each template contains the domain level metadata, field level metadata and any associated value-level controlled terminologies. Similarly, for assays, we preloaded standard templates for sample and feature metadata based on the ISA model. The assay data are tailored to the specific assay technologies (e.g. microarray gene expression, flow cytometry, proteomic and immuno assays). The 'manage activity' page allows a data manager to create and edit dataset descriptors for each planned activity based on the preloaded standard templates. Features include excluding/including fields, setting mandatory fields, specifying controlled vocabularies for a field's permissible values, as well as adding new fields.

4.5.2 Data Storage

Data storage often takes different forms for different purposes throughout the data life cycle. In the early stages, data are stored in an operational database, such as those embedded in EDC tools and Biobanks. These databases are specifically designed for operational data collection and management to support clinical conduct. Once operational data collection is complete, data from these various sources are processed and ingested into a data repository, which is a type of database that compiles data into a well-defined consistent data model. Data repositories provide an efficient structure for data storage offering researchers a consistent 'single source of truth' for their project's primary datasets. Unfortunately, the process of creating or using a data repository is omitted in most translational research projects. More often, only simplified data structures, often called data marts, that are optimized for a limited set of project-specific queries and analyses are used for the project data. These data marts typically perform well for the limited purposes for which they were designed. However, data marts are often difficult to interrogate for purposes not considered when the project was scoped. Given the richness inherent in TR data, data marts inhibit opportunities for data reuse.

Moving data between these different storage systems requires Extract, Transform and Load (ETL) processes. These processes often can create a bottleneck in the data pipeline depending on the source and target systems, the design of the ETL processes and the extent of required data transformation/integration and delay data use and analysis. TREMF was designed to support two models for data: a dataset-based model suitable for storing primary data, and an observation-based model suitable for storing integrated data. Based on these two models we

implemented a two-schema storage solution that takes away the inefficiencies of moving data between different database implementations, thus removing the need for a separate ETL process. These two storage solutions are: the Translational Medicine Data Archive (TMDA) storing primary annotated datasets, and the Translational Medicine Data Warehouse (TMDW) storing integrated subject and sample observations for querying and exploring (Figure 4.4d).

4.5.3 File management and data loading

The TDMA module provides users with credentials to upload and manage the project's data files. Creation of new data files, or changes to existing files, are represented in an audit trail that records the file loading status, the user account associated with the file operation and the timestamp of the operation. The project's staging directory organizes the project files and serves as the entry point for loading data into the platform's databases. To load a file, the user launches the loading wizard which takes them through a series of steps to associate the file with one of the previously defined dataset descriptors. The descriptor is used by the loading process as a reference for parsing and validating the file contents accordingly. Once validated, the loading process persists the file as an annotated dataset in the archive database. The integration and harmonization process next extracts the dataset's content and loads the content into the observation data warehouse as illustrated by Figure 4.11b. The loading wizard natively supports CDISC SDTM formatted files.

4.5.4 Integrating and harmonizing data?

Integration and harmonization are the processes during which data is transformed from the structured dataset form to the granular observation-based form. These processes take place as part of the data loading wizard (Figure 4.11d). Data for each subject are integrated across different domains (demographics, diagnosis, laboratory tests, medications, etc.); across studies (clinical trials); and when molecular data is measured, across multiple omics platforms linking omics data to the phenotype data. The resulting integrated data is loaded into the data warehouse. The backbone of the integration and harmonization process is the dataset meta-model of the TREMF implemented in each dataset's metadata descriptor. Different datasets will have different column names and different compositions, but the 'meaning' of these columns, defined by the meta-model, makes it possible to identify focal columns that are used as 'integration keys' to link data from different datasets. Semantic harmonization of clinical data is achieved by processing the controlled values for the integration keys to create a unique set of harmonized 'observed features'. The basis of the harmonized data is the creation of an 'observation descriptor' describing the data values available for each observed feature according to the observation model defined in layer 3 of TREMF. These descriptors become the searchable elements for the data warehouse observation query.

4.5.5 Data Access

The software architecture of EHS is based on a loosely coupled backend service application and a frontend client application communicating programmatically (this is done technically using a RESTful (web-based) Application Programming Interface (API). The API-based

design is expected to encourage the development of new applications as well as third-party integrations with other platforms. Incorporating EHS data into any external app requires a HTTP library and a JSON parser. HTTPS is supported for secure client access. EHS API is organized into three distinct API collections: the 'Data Archive API' provides read-only accessibility to the TMDA database allowing third party applications to find and access meta-data rich translational medicine research datasets, the 'Observation query API' exposes endpoints to query TMDW, selecting, filtering and combining data from all available domains of data (this API supports building integration pipelines for analytical tools), and the 'Apps API' is a client dependent API serving the frontend client applications (Figure 4.11e).

4.5.6 Retrieving and exploring data

4.5.6.1 Project archive module

Often researchers need to find and retrieve entire third-party datasets to process and include in their analyses or to publish their own data for reuse. Based on the 'Data Archive API', this module offers a straightforward graphical user interface to search and browse the catalogue of projects stored in the data repository, enabling the retrieval of structurally and semantically annotated *primary* datasets for sharing and reuse. For each stored project, a project summary page provides metadata about the project and links to the associated datasets available to download.

4.5.6.2 Data explorer & query module

The data explorer is a visual browser and query interface supporting observation-level exploration and retrieval of data stored in the TMDW. Based on the 'Observation query API', the data explorer uses on-demand synchronized charts to provide a hypothesis-free, interactive, and easy-to-use graphical interface to explore integrated subject and sample related observations. This is particularly useful during the initial phases of research when no clear hypotheses are immediately available. The user interfaces design is based on an intuitive domain-aware visual layout organizing data across three panels.

- The first panel hosts subject and study data elements, such as subject demographics, study arms, visits, etc…
- The second panel is for exploring the harmonized clinical observation features organized by the CDISC general observation class-domain hierarchy
- The third panel is for molecular observations organized by assay type

The three panels offer a faceted browsing component for the related data elements and an interactive dashboard showing a chart for each selected data element or observation feature.

Besides offering a visual environment for browsing and localizing interesting features in the data, the 'data explorer' acts as a visual query builder to support hypothesis generation through interactive data selection and filtering directly off the charts. All charts across the three panels are synchronized. Adding a chart for an observation or a data element essentially adds it to the

query. Applying filtering on any chart automatically cascades the effect of the filter to all other visualized charts effectively affecting the number of subjects and samples currently satisfied by the query. For example, filtering for an explicit range of 'diastolic blood pressure' measurements followed by applying a filter on 'study arm' and 'visit' will automatically be propagated to the related omics assays, reducing them to only the matching filtered subjects and vice versa. In addition to the plotted charts, a count panel displayed on top is dynamically updated to show the number of subjects, samples and assays satisfying the selected and filtered data.

4.5.7 Data extraction

The data custodianship environment streamlines the process by which researchers extract and retrieve data for their analyses that are up-to-date and properly annotated. EHS aims to manage the data extraction process by allowing users to export data by creating 'analysis datasets'. An analysis dataset is formatted to facilitate the application of data analysis tools the dataset (i.e. analysis-ready). An analysis dataset stores the query that the user specifies to extract the required data rather than the actual resulting data. This allows export files to be automatically generated every time there are changes to the primary data source including the contents of any derived fields. The analysis dataset also holds general metadata information about its contents, as well as field-level metadata describing the columns in each of its data files. Sharing permissions enable users to share or publish their datasets widely to promote open data philosophies and experimental reproducibility.

A 'data cart' feature in the data explorer module allows the user to save their queries to retrieve later or to 'checkout' whereby the query results are exported into analysis datasets. At checkout, the server prepares the data exports according to the data query giving the user the option to add their own descriptions and tags before they are ready to download or save to their analysis datasets library. Analysis datasets are stored in a separate data collection that is user-focused and not project based. They are accessible via the API using their unique URL to download associated export files. The analysis datasets library page provides a user workspace to manage their own created datasets, shared datasets from other users and datasets made public by other users.

4.6 Case-studies

The work presented here was developed at Imperial College London Data Science Institute (ICL-DSI) as part of collaborations with Innovative Medicine Initiative (IMI) funded projects. Requirements for EHS were gathered through supporting and addressing real world problems experienced by IMI TR consortia including U-BIOPRED, OncoTrack, PreDiCT-T and BIOVACSAFE. The first production implementation of the TREMF platform was developed for BioVacSafe (Biomarkers for Enhanced Vaccine Safety), an IMI funded project that investigates vaccine reactogenicity to enhance immunosafety of novel vaccines.

4.6.1 The ERS proof-of-concept

To demonstrate the applicability of the developed framework and the usability of our platform in supporting cross-study research and re-use of data, we conducted a pilot study with the European Respiratory Society (ERS) to compare various subpopulations of asthma and COPD patients from two independent studies: 'U-BIOPRED' (Unbiased BIOmarkers in PREDiction of respiratory disease outcomes) and 'EvA' (Emphysema versus Airway disease) respectively. Using the metadata module, a meta-study project was created for the pilot, defining two studies with different subject cohorts, four clinical activities: laboratory tests, vital signs, spirometry and reversibility tests, and a gene expression assay. For each activity, a dataset descriptor was pre-defined based on one of the preloaded CDISC SDTM standard templates to guarantee that overlapping clinical variables are uniquely represented across the two studies. Data files selected for the pilot were then uploaded to the dedicated project drive space and each loaded into the data repository and data-warehouse simultaneously via the loading wizard. Once loaded, data from a total of 1,294 subjects and 39 unique and harmonized clinical variables were readily integrated in the observation data-warehouse. Using only the data explorer (applying no computer programming techniques), lead investigators interrogated the integrated data, used these data to generate hypotheses using visually coordinated plots of the clinical features of interest, determined instantly whether sufficient samples were available to conduct follow on analyses and, finally, saved and extracted the desired analysis ready datasets. One of these hypotheses was to test whether asthma and COPD sufferers with abnormally high eosinophil cell counts and airflow obstruction share similar gene expression profiles. This proof-of-concept demonstrated the feasibility of reusing data for secondary research gathered from two independent consortia by utilizing our platform and its underlying metadata framework.

4.6.2 BioVacSafe Data Management System

Following the ERS proof-of-concept, we continued developing EHS as part of delivering a production implementation for the BioVacSafe project. BioVacSafe is a multi-study and multi-site project that generated clinical, pre-clinical and 'omics data for assessment of vaccine responses with an emphasis on immunosafety and immunogenicity[24]. Data were collected and stored from two independent sites, running five clinical trials investigating seven cohorts with overlapping clinical and molecular observations. Clinical data included subject demographics, laboratory tests (hematology, urinalysis, chemistry), vital signs and MedDRA-coded adverse events (solicited and non-solicited). Data from molecular assays included: microarray gene expression profiling, cytokine/chemokine profiling, whole blood leukocytes, flight mass spectrometry (CyTOF) and Immunophenotyping of Monocytes using Fluorescence activated cell sorting (FACS). The platform and its underlying metadata framework provided an intuitive, systematic and standard compliant approach to streamline the process of data integration and harmonization across the consortia's work streams. Once loaded, the explorer module also offered researchers a systematic hypothesis-free method to navigate through the

entire range of data, and to export different research-focused analysis datasets. For instance, a common exploratory use case was to select subjects based on some combined clinical observations specifying a potential reactogenicity profile and export their corresponding assay data to run differential analysis to look for correlated molecular signatures.

4.6.3 Summary

This chapter introduced the art of data management. The nature of data was explored in the context of the complicated transformations that raw data must undergo to prepare data for meticulous interrogation leading to sophisticated scientific analysis and interpretation. Rigorous application of community data standards was explained as a foundational necessity to unlock the power and value of study data for subsequent integration and reuse to address research questions not considered by the study investigators for which such data was originally collected. An open system, EHS, that implements comprehensive data flow and lifecycle processes was presented as a tangible prototypical implementation having precedence in driving precision medicine data analysis conducted by large scale public private partnerships.

Readers will certainly have gained an appreciation of the substantial data processing infrastructure required to support TR programs, infrastructure that is often, in the best systems, hidden from scientific data consumers. However, readers having clinical and scientific responsibilities for TR projects should also now realize their critical responsibility in applying their scientific expertise to aid in the design of the data models and structures that will support their research efforts. This aid is not only crucial for expediting the achievement of initial research goals but may be the key to realizing high value unexpected research opportunities through reuse of high-quality data residing in readily approachable information systems.

Chapter 4 References:

[1] Altman RB. 2012. Translational bioinformatics: linking the molecular world to the clinical world. Clin. Pharmacol. Ther. PMID: 22549287

[2] Butte AJ. 2008. Translational Bioinformatics: Coming of Age. J. Am. Med. Inform. Assn. PMID: 18755990

[3] Griffin PC et al. 2017. Best Practice Data Life Cycle Approaches for the Life Sciences. bioRxiv. doi:10.1101/167619

[4] Canuel V et al. 2015. Translational research platforms integrating clinical and omics data: a review of publicly available. Brief Bioinform. PMID: 24608524

[5] Dunn W et al. 2016. Exploring and visualizing multidimensional data in translational research platforms. Brief Bioinform. PMID: 27585944

[6] Skolariki K, Avramouli A. 2016. The use of translational research platforms in clinical and biomedical data exploration. GeNeDis, Springer International Publishing.

[7] dbGaP [Internet]. Bethesda (MD): NCBI [Date Unknown; Accessed January 2019]. Available from: https://www.ncbi.nlm.nih.gov/gap

[8] Immport: Bioinormatics for the Future of Immunology [Internet]. [Date Unknown; Accessed January 2019]. Available from: http://www.immport.org/immport-open/public/home/home

[9] Wilkinson MD et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data. PMID: 26978244

[10] Martin AM, Seigneuret N et al. 2013. Data Standards are Needed to Move Translational Medicine Forward. Transl. Med. doi: 10.4172/2161-1025.1000119

[11] Tenenbaum JD. 2016. Translational Bioinformatics - Past, Present, and Future. GPB. PMID: 26876718

[12] Vamathevan J, Birney E. 2017. A Review of Recent Advances in Translational Bioinformatics: Bridges from Biology to Medicine. Yearb. Med. Inform. PMID: 29063562

[13]Louie B et al. 2007. Data integration and genomic medicine. J Biomed. Inform. PMID:16574494

[14] CDISC. Clinical Data Interchange Standards Consortium. 2005.

[15] Sansone S.-A. et al. 2012. Toward interoperable bioscience data. Nat. Genet. PMID: 22281772

[16] Rocca-Serra P. et al. April 2016, 2018. eTRIKS Standards Starter Pack Release 1.1

[17]English LP. Improving data warehouse and business information quality: methods for reducing costs and increasing profits. New York (NY). Wiley, 1999.

[18] McGilvray, Danette. Executing data quality projects: Ten steps to quality data and trusted information (TM). Amsterdam (NL). Elsevier, 2008.

[19] Dasu T, Johnson T. Exploratory data mining and data cleaning. Vol. 479. Hoboken (NJ). John Wiley & Sons, 2003.

[20] Meta Object Facility (MOF). Object management Group. Specification v1. 4, 2002.

[21] Rocca-Serra P et al. 2008. Specification documentation: release candidate 1. ISA-TAB 1.0.

[22] Fowler, M. 1997. Analysis patterns: reusable object models, Addison-Wesley Professional. ISBN: 978-0201895421

[23] Open Clinica [Internet]. Open Clinica LLC. [Date Unknown; Accessed January 2019]. Available from: https://www.openclinica.com/

[24] REDCap [Internet]. Vanderbilt University. [Date Unknown; Accessed January 2019]. Available from: https://www.project-redcap.org/

[25] Lewis, D. J. M., Lythgoe, M. P. 2015. Application of 'Systems Vaccinology' to Evaluate Inflammation and Reactogenicity of Adjuvanted Preventative Vaccines. *J Immunol Res*

Chapter 5: Getting Knowledge from Data

Xian Yang and Yike Guo

5.1 Obtaining knowledge from biomedical data

5.1.1 How to detect and remove confounders

Translational medicine research commonly adopts high-throughput technologies to generate quantitative measurements of patients, such as microarrays, bead chips, mass spectrometers and gene sequencing. This section discusses methods of detecting and removing batch effects (also some unwanted variations) in high throughput experiments. Batch effects are technical sources of variations commonly occurring during sample preparation. In precision medicine research, large cohorts are typically enrolled in the study leading to a corresponding large number of samples. Handling many samples at once is technically impractical and hence the data must be split into manageable rounds of processing. Batch effects cannot be avoided in the raw data generation steps. Samples processed under the same conditions (e.g., consistent laboratories, reagent lots and personnel) will likely be inadvertently biased leading to variation across sample sets processed under different conditions. If technical variations confound with biology, then it becomes difficult to detect real differential features from the dataset. Some examples can be found in[1] [2], in which the biological factors and technical variables are extremely correlated that results in concerns on the validity of biological findings[3]. Figure 5.1[4] provides a sample batch effect, where ten example genes are susceptible. The data used in this figure is from a bladder cancer study[5]. Hence, before carrying out any statistical or machine learning methods for biomarker detection or predictive model construction, it is necessary to check whether batch effects have been avoided by careful experimental design.

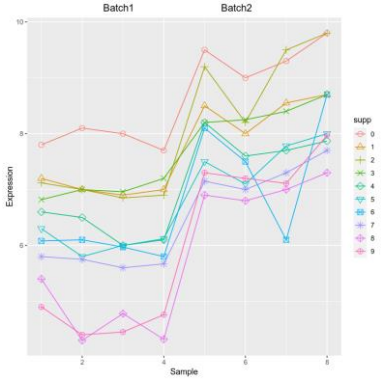


Figure 5.1: Ten example genes from different samples having different expression levels in two different batches.

A traditional way to detect batch effect is applying exploratory approaches, in which principal component analysis (PCA)[6 7] is one of the most popular methods. As an unsupervised machine learning technique, PCA[8] performs an orthogonal transformation to convert observations of correlated variables into linearly uncorrelated ones. Principal components are linear combinations of original features (e.g., genes, proteins, lipids). The transformation is presented as clusters of data points. The data points can be shown in a scatterplot where the coordinates are principal components. PCA is commonly used as an exploratory tool to provide an intuitive understanding of the dataset[9 10 11]. Should groups or separations of data points be clearly discriminated in the scatterplot, then the data within these groups can be divided into corresponding datasets. The data points within a group identified by PCA can then be investigated to assess whether these share similar characteristics. The effect of confounders can further be quantified by examining principal components of the data. As principal components capture variations of the dataset, the method incorporates both biological and technical variability. By checking the correlation between principal components and known confounders, such as data handling time and site, we can quantify the degree of batch effects.

The volume of data generated in a precision medicine study grows immensely during the conduct of the study. Therefore, PCA assessment of large datasets has become a time-consuming task. Abraham and Inouye 2014 [12] developed a highly efficient PCA implementation based on randomized algorithms. With significant reduced time and cost, identical accuracy in extracting the top ranked principal components is achieved. This approach is one example of adapting existing traditional methods to support big data in precision medicine research. Another extension of PCA is the guided PCA method[13] which calculates a test statistic to determine whether batch effects were introduced. The proportion of total variance owing to batch effects is calculated as well as its probability value. Instead of subjective visualization, the guided PCA method provides a way to quantify batch effects[14].

Hierarchical clustering is another popular exploratory method for batch effect detection. Similar with PCA, it is also an unsupervised machine learning method to discovery patterns of samples. By using hierarchical clustering, samples are organized into a hierarchy and can be visualized in a dendrogram[15]. Neighboring samples in the hierarchy are close by some measure of distance. For instance, patients with similar Omics profiles will be closely clustered together. Some common similarity functions for distance calculation are Euclidean, city block (or

Manhattan), correlation and Hamming distance. Particularly, Euclidean distance emphasizes on large differences as the distance is squared, while the city block distance removes the square. The correlation coefficient is commonly used to measure the similarity of two time-series variables. For ordinary variables, such as {low, medium, high}, before calculating the distance we need to convert them into real valued numbers (such as 1/3, 2/3 and 1). For categorical variables, hamming distance can be applied. In short, the choice of distance function fully depends on the data types and the requirements of similarity detection.

Once batch effects have been detected, it is crucial to remove them. Along with simple removal methods, such as mean centering and ratio-based methods, there exist some comprehensive methods for high-throughput data in precision medicine research. PCA is not only a powerful tool for batch effect detection, it can be also used to remove batch effects. Proposed in Alter et al 2000[16], PCA together with ***singular value decomposition*** (SVD) has successfully remove batch effects in yeast cell cycle experiments and cancer study[17]. As it identifies the directions of greatest variation, this method only works well when the systematic bias effect generates the largest variation. When the variation of experimental design is of similar magnitudes with the batch effects, this approach would not work. Therefore, other methods, such as the distance weighted discrimination (DWD) method [18] , are developed. DWD is intrinsically a variant of support vector machine (SVM), aims at finding a separating hyperplane between two batches and projecting them onto the plane. With the estimated mean of each batch, the whole dataset is corrected by removing the DWD plane multiplied by the mean.

Linear models can be used for Batch effect removal as well. If some specific surrogates of batch factors are known, such as processing time and site, then we can directly incorporate these surrogates in linear models for group comparison (e.g, ComBat[19] [20]). If the sources of batch effects are not known, then surrogate variable analysis (SVA)[21] can be applied. Through investigating data patterns, SVA determines sources of batch effects and incorporates them into linear model for batch effect removal.

5.1.2 Handling missing values

In precision medicine research, missing values are frequently encountered, such as defective cells in a gene expression microarray, unmatched genome positions in a functional genomics assay, unreliable measurements due the detection limit of the instruments or removed participants in a clinical trial. As downstream data analysis methods usually require complete data matrices, handling missing values significantly influences the results of statistical analysis. For example, SVD, PCA and SVM cannot work well in the presence of missing values[22]. Therefore, it is necessary to explore methods of coping with missing values. Missing data

values can generally fall into two categories: values are missing at random and values, when absent, provide information about the task at hand (such as saturated detectors)[23]. The detection of a missing value of the second type is usually incorporated directly into the downstream analysis model. The most common way to deal with missing data values in the first category is doing data imputation[24]. Zero insertion is the simplest way that all missing data values are replaced by zero. Considering the bias introduced by zero insertion, more sophisticated strategies are established to estimate possible missing values from the data. The remainder of this section will introduce methods to handle missing values for typical data types, where methods for microarray data have been extensively studied. Other large-scale datasets such as mass spectrometry and phenomic data will also be investigated.

5.1.2.1 Imputation methods for microarray datasets

There are massive missing value imputation methods for microarray gene expression data, which can fall into different categories according to different criteria. For example, four main types of imputation methods are summarized in Acuña and Rodriguez 2004[25], which are case deletion, mean imputation, median imputation and K nearest neighbor imputation (kNNI). Case deletion removes all samples with missing values, while mean imputation replaced missing values by calculating the mean based on its known features. Median imputation chooses the median rather than mean, which is supposed to be more robust to outliers. KNNI uses K nearest genes to impute the missing values.

In Aittokallio 2010[26], imputation methods are divided into two different categories: generic statistical methods and application-specific methods. Generic statistical methods include the traditional ones summarized in Little 1987[27].

1. Hot deck imputation methods do the imputation using non-missing cases in the neighborhood, among which KNNI is the most popular one.
2. Model-based imputation predicts missing values by using statistical models via the expectation-maximisation (EM) algorithm[28].
3. Multiple imputation methods provide more than one value for each missing point, making the downstream methods work on each complete dataset individually and, combined, generate the final results which also reflect sampling variability.
4. Cold deck imputation imputes missing values by making use of external information, such as data from related studies.

Table 5.1 from Aittokallio 2010[29] shows some basic concepts of typical generic methods. As a departure from generic statistical methods, application specific methods take quality issues and

experimental designs into consideration. The first attempt uses the gene functionality information from the Gene Ontology (GO)[30]. GO-based semantic similarity combined with expression similarity is used for relevant gene selection during imputation. Moreover, there are some other sources used for expression level prediction, such as promoter sequence binding information on transcription factors[31], and histone acetylation state information on chromatin structure[32].

Table 5.1: Representative missing value imputation methods (from Aittokallio 2010[33])

Imputation method	**Prediction variables**	**Estimation method***
K nearest neighbours	Matrix rows (genes)	WA
Least squares regression	Matrix rows or columns (arrays)	LS
Local least squares	Matrix rows or columns	LS
Singular value decomposition	Singular vector (‘eigengens’)	EM
Bayesian principal component analysis	Principle components	EM
Gaussian mixture clustering	Gaussian components	EM
Support vector regression	Support vectors	QP

* WA, weighted average; LS, least-squares optimization; EM, expectation-maximisation algorithm; QP, quadratic programming.

The review in Moorthy, Mohamad, and Deris 2014[34] further categories the imputation methods according to the type of information, which are grouped into four main streams:

1. Global approach algorithms impute missing data based on global correlation information obtained from the entire data matrix. SVD imputation[35] [36], Bayesian principal component analysis (BPCA)[37] and support vector regression are typical global methods. Many of these methods are parameter free. For example, BPCA imputation determines the number of prediction variables automatically.

2. Local approach algorithms perform missing data estimation by only using local similar structures in the dataset. The first methods dedicated to microarray data is KNNI, where a distance-weighted average over K genes is used as an estimate for the missing values

[38]. There are some modifications of KNNI, which mainly focus on replacing the weighted average in the estimation step with regression models and improving the nearest neighboring genes selection process by using Bayesian variable selection or Spearman's correlation measure[39] [40] [41]. Least square (LS) imputation takes advantage of both gene and array correlations for fast missing data imputation, where an adaptive procedure for combining estimates is proposed[42]. Local least square (LLS) imputation is the local version of LS, which is more popular than LS as it can automatically decide the neighborhood size from the data[43]. LLS has an iterative version, called ILLS. ILLS imputes missing values sequentially starting from the genes having missing rates, the newly imputed genes are then reused in subsequent rounds of imputation[44]. Gaussian mixture clustering (GMC) imputation is also a local imputation method but it is capable of using more global correlation information[45]. GMC imputation uses the EM algorithm to cluster data into different groups whose values are averaged to obtain the estimates of missing values. Collateral missing value is another imputation method, making use of multiple parallel imputations, which gives better performance than BPCA, KNN and LS on ovarian cancer and yeast sporulation time series data[46]. Ameliorative missing value imputation further improves estimation by applying Monte Carlo simulation techniques[47]. When the rate of missing values is high or binary matrices, adaptive bicluster-based approach developed in Colantonio et al. 2010[48] is a good choice of imputation.

3. Hybrid approaches exploit both local and global correlation information for data imputation. LinCmb[49] is a typical hybrid approach that combines missing values estimated by KNN, SVD, BPCA and GMC imputation methods. LinCmb allows the imputation to be adaptive to the datasets such that global methods have a stronger weight in determining missing values when more missing entries are present.

4. Knowledge based approaches incorporate background knowledge into data imputation. Approaches in this category correspond to application specific methods defined in Aittokallio 2010[50], where GO-based imputation and histone acetylation information aided imputation are two popular ones.

5.1.2.2 Imputation methods for other big datasets

Imputation methods for microarray datasets can be, in principle, adapted to other big datasets. In particular, the label-free liquid chromatography mass spectrometry (LC-MS) based proteomics approach generates big datasets, providing quantitative profiling of complex peptide mixtures[51]. However, a substantial fraction of data at the peptide level is missing from proteomic datasets, making downstream analyses difficult[52 53 54]. Similarities are shared between proteomics and microarray-based gene expression analysis. They both return large matrices, where proteomics gives a matrix of quantitative values of peptides and microarray generates probe-level transcripts[55]. However, the missing rate of proteomic datasets is much higher than the microarray-based gene expression datasets. About 20-50% of peptides values can be missing while less than 5% of transcript abundances are not observed. Moreover, the missing values usually result from the combination of random and non-random processes, making the imputation more challenging[56]. The work in Webb-Robertson et al. 2015[57] has evaluated 10 typical imputation methods (e.g., KNN, LLS, BPCA imputation) examining them for their individual merits and caveats with respect to LC-MS proteomics data. It has been found that no such method can always perform better than the others. Thus, it is preferable to consider application-specific methods that account for the mechanisms responsible for the missing data[58]. Other relevant data sources for missing value imputation, such as the clinical annotation of samples [59], message RNA level, cellular role and information about experimentally undetected proteins can also help the imputation of LC-MS proteomics data[60].

In precision medicine research, another important large-scale data type, phenomic data, also inevitably contains missing values caused during data collection process. Phenomic data mainly contains the information of demographic measures (e.g., gender, race), environmental exposures, living habit (e.g., smoking, exercise), general health status (e.g., body mass index, blood pressure and forced vital capacity), medical images (e.g., fMRI scan), drug history and family disease history[61]. Integrative analysis of phenomic data and other Omics data has been found to improve the understanding of diseases[62 63 64 65 66]. Approaches for reducing missing data include increasing structured data documentation, reducing data input errors, and utilizing natural language processing[67]. Here, we only focus on analytical approaches to cope with missing data, primarily imputation methods. Different from datasets generated by high

throughput experiments (e.g., microarray and LC-MS) where continuous abundance values are measured, phenomic datasets contain various data types including continuous, nominal, binary and ordinal data types. As the data type is more complex, a lot of imputation methods for Omics data types cannot be well adapted to phenomic data. Moreover, many imputation methods for Omics data are established by exploring the correlation of variables. However, variables in phenomic datasets are not necessarily correlated that some missing data points cannot be imputed from other observed variables.

Multivariate imputation by chained equations (MICE) is one example method for addressing missing values in phenomic datasets [68]. MICE deals with multivariate missing data by factorizing the joint conditional probability as a sequence of conditional probabilities. Next, MICE performs multiple regressions sequentially based on different types of missing covariates. It is a nonparametric approach using Gibbs sampling to estimate parameters. Besides MICE, there is a random forest-based imputation method to impute phenomic data, which is called MissForest[69]. MissForest sets variables with missing values as response variables. Other variables are used to predict the response variables through the resampling-based classification and regression trees. It is an iterative method and the final results are obtained when the imputed values converge. In Liao et al. 2014[70], modifications on KNN dedicated to phenomic data with mixed types of variables have been proposed. As KNN is a correlation-based method, we should carefully choose methods of calculating correlation according to different types of variables. Table 5.2 has listed candidates of correlation measures for different data types used in the correlation construction of KNN. The regression methods used for imputing missing data points of different types are shown in Table 5.3 from Liao et al. 2014[71].

Table 5.2: Correlation measures for different data types

Variables	**Continuous**	**Ordinary**	**Binary**	**Categorical**
Continuous	Spearman	--	--	--
Ordinary	Polyserial [72]	Polycoric[73],[74]	--	--

Binary	Point Biserial[75]	Rank Biserial[76]	Phi[77],[78]	--
Categorical	Point Biserial extension	Rank Biserial extension	Cramer's V[79]	Cramer's V

Table 5.3: Methods for gathering imputation information of different data types from K nearest neighbours

Variables	**Continuous**	**Ordinary**	**Binary**	**Categorical**
Regression methods	Linear regression	Ordinal logistic regression	Logistic regression	Multinomial logistic regression

5.1.3 Basic statistical inference methods

The process of performing statistical hypothesis testing in the translational medicine research is illustrated in Figure 5.2. The null hypothesis and its alternative are defined first. Then a test statistic and its p-value are calculated if the null hypothesis is assumed to be true. Next, the null hypothesis is rejected by check whether the p-value is smaller than the pre-defined significance level [80].

When multiple comparisons to test null hypotheses are performed, there is a potential increase in statistical error. For example, if 10,000 independent tests are performed, and the null hypotheses are true, we expect about 500 tests to have a p-value of less than 0.05. This would lead to falsely invalidating the null hypothesis in those 500 tests, referred to as type I error or false positives. In high throughput analysis, such as microarray, we may do more than hundreds of thousands of tests. Thus, it is necessary to correct p-values for multiple testing. Bonferroni correction[81] is a popular way to compensate the increase of Type I error[82]. Although the family-wise error rate (FWER) can be controlled by the Bonferroni method or its extensions, such as the Šidák procedure[83], the Tukey's test[84], the Hochberg's step-up procedure[85] and the Dunnett's test[86], the power of detecting real differences is largely reduced. Therefore, it is necessary to develop better techniques for multiple testing, such that the Type I error can be maintained without inflating the rate of Type II error (i.e. false negatives). For large scale multiple testing

in precision medicine research, we can instead control the false discovery rate (FDR) following the Benjamini–Hochberg procedure[87].

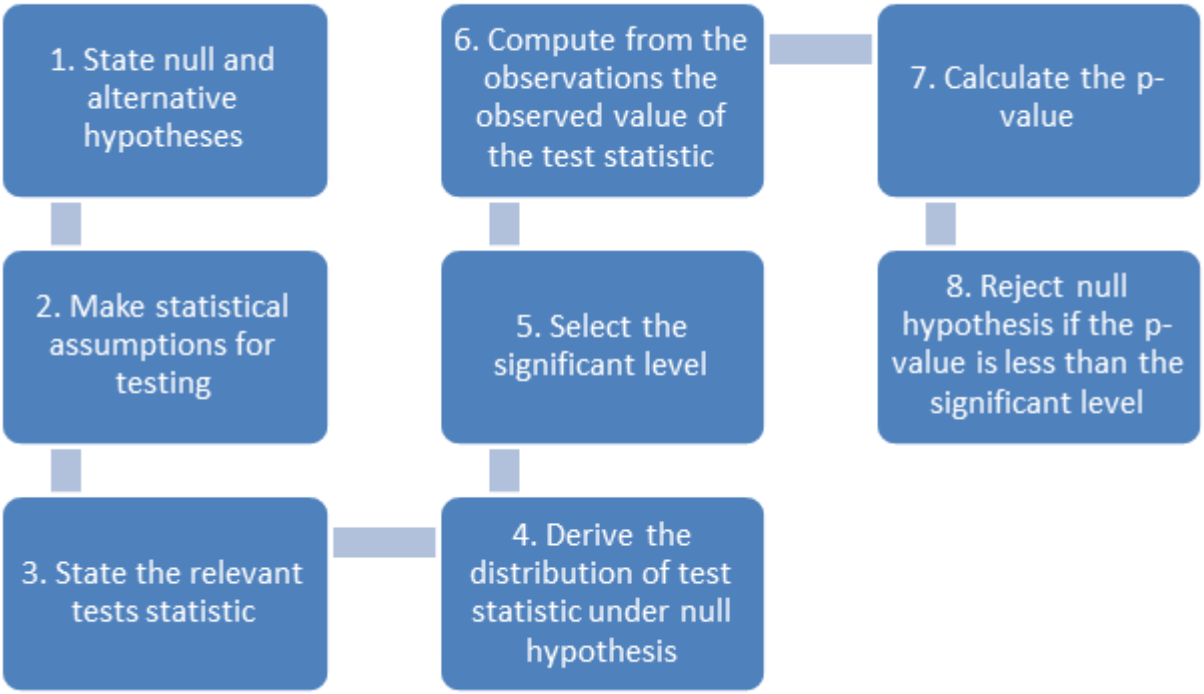


Figure 5.2: The process of carrying out statistical hypothesis testing.

Various statistical testing methods are available for comparing the distributions, such as student's t-test[88], Welch test[89], Mann-Whitney U test or Wilcoxon rank-sum test[90], Kolmogorov-Smirnov test[91], Chi-squared test[92], F-test (analysis of variance, ANOVA)[93] and permutation test[94]. Usage of common statistical tests under different conditions is shown in Figure 5.3 (from https://onlinecourses.science.psu.edu/stat500/node/68).

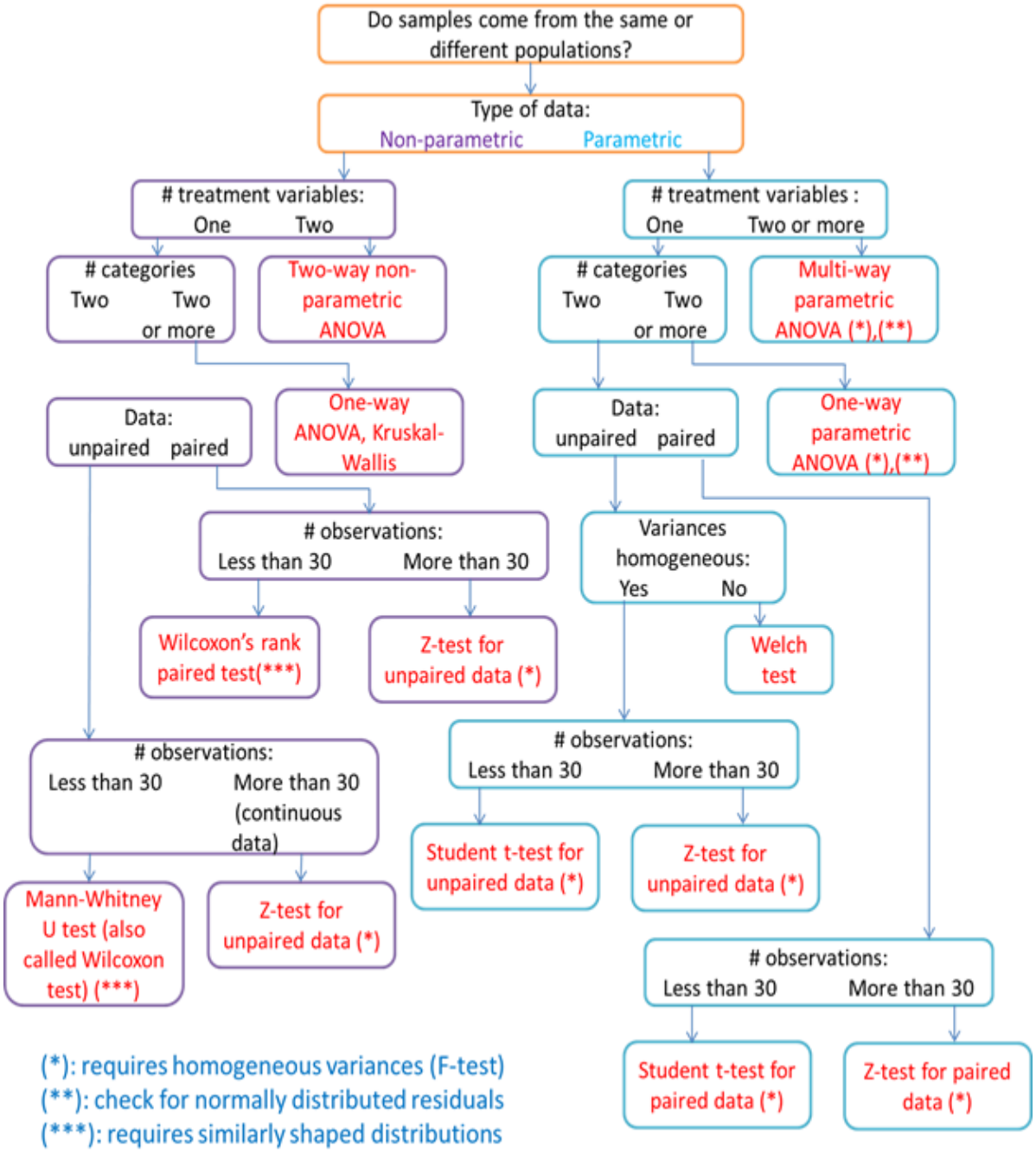


Figure 5.3: Pipelines of carrying out statistical tests in precision medicine research

5.1.4 Feature selection and construction of classifications models

To extract information from quantitative datasets, we can use machine learning methods for predictive model construction[95 96 97 98]. The datasets generated in translation medicine research usually have much larger number of features than the sample. Thus, machine learning methods

would suffer from the curse of dimensionality problem. To solve this issue, feature selection methods for reducing the feature dimensionality is of super importance.

5.1.4.1 How to choose classification methods?

There are various classification algorithms, such as linear classifiers (e.g., Fisher's linear discriminant[99], Logistic regression[100], Naïve Bayes classifier[101], Support Vector Machine[102], decision trees[103], Neural networks[104], Relevance vector machine[105] and deep learning[106]). Selecting classification methods for specific input datasets will likely lead to lively debates. Exhaustively trying different methods to see which one fit the data best can be supported by the model selection methods (such as Akaike information criterion[107], Bayesian information criterion[108] [109], Bayes factor[110] [111]), and the cross-validation (CV)[112] performance evaluation process. Common CV methods include *leave-one-out* cross-validation (LOOCV) and k-fold cross-validation that splits the whole dataset into training and validation sub-datasets (as shown in Figure 5.4).

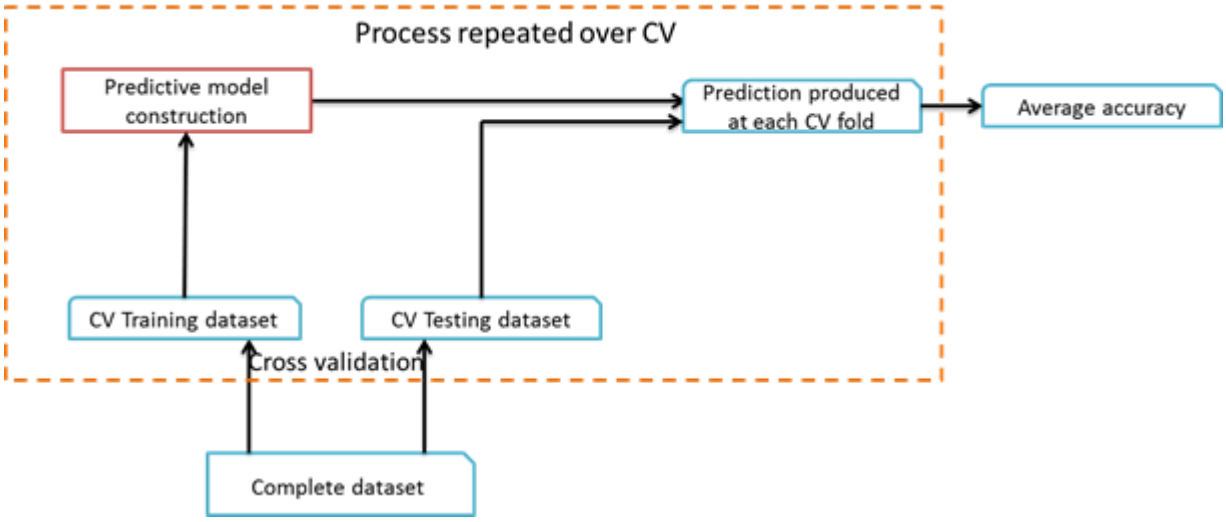


Figure 5.4: The process of cross validation for classification

Figure 5.5 shows common performance evaluation metrics (from "Sensitivity and Specificity - Wikipedia" n.d.), which are accuracy, sensitivity, specificity, false positive rate and false

negative rate. Sensitivity is calculated as the probability of detecting true positives, while specificity shows the ratio of finding negative ones correctly. To examine the sensitivity and specificity at the same time, the receiver operating characteristic (ROC) curve can be introduced, in which sensitivity is plotted against (1-specificity) by varying threshold settings[113].

Total population	Condition positive	Condition negative	
Test outcome positive	True positive	False positive (Type I error)	
Test outcome negative	False negative (Type II error)	True negative	
	True positive rate (Sensitivity) $= \frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate $= \frac{\sum \text{Flase positive}}{\sum \text{Condition negative}}$	Accuracy $= \frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$
	False negative rate $= \frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	True negative rate (Specificity) $= \frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	

Figure 5.5: Summary of statistical measurements of performance

5.1.4.2 Selecting features in the classification model

As we have mentioned, the number of features is usually much larger than the number of samples in most translational medicine research. Like in the U-BIOPRED project, each sample/patient is characterized by high-dimensional features (which can be up to millions). This curse of dimensionality problem would result in the model over-fitting. To alleviate this issue, feature selection methods can be used. Popular feature selection methods can fall into three categories: filtering, embedding and wrapping. Filtering is the most straightforward way to select features before predictive model construction. Examples methods are statistical tests using p-values for discriminant feature selection. Limitations of filtering methods lie in the difficulties in jointly detecting predictive power of multiple features. An alternative approach is embedding, like Lasso[114] and Bayesian learning[115] construct linear predictive model while selecting features by introducing sparse constraints. The process of this embedded feature selection approach is shown in Figure 5.6. s the performance is evaluated via CV that in each iteration different training dataset is used for feature selection, the process in Figure 5.6 cannot generate a single prediction model for future sample prediction. Therefore, it is necessary to explore feature selection methods as discussed in [116] [117] [118] [119]. The third category is the

wrapping method, which combined use feature selection and machine learning methods to find the best combined approach returning the best performance.

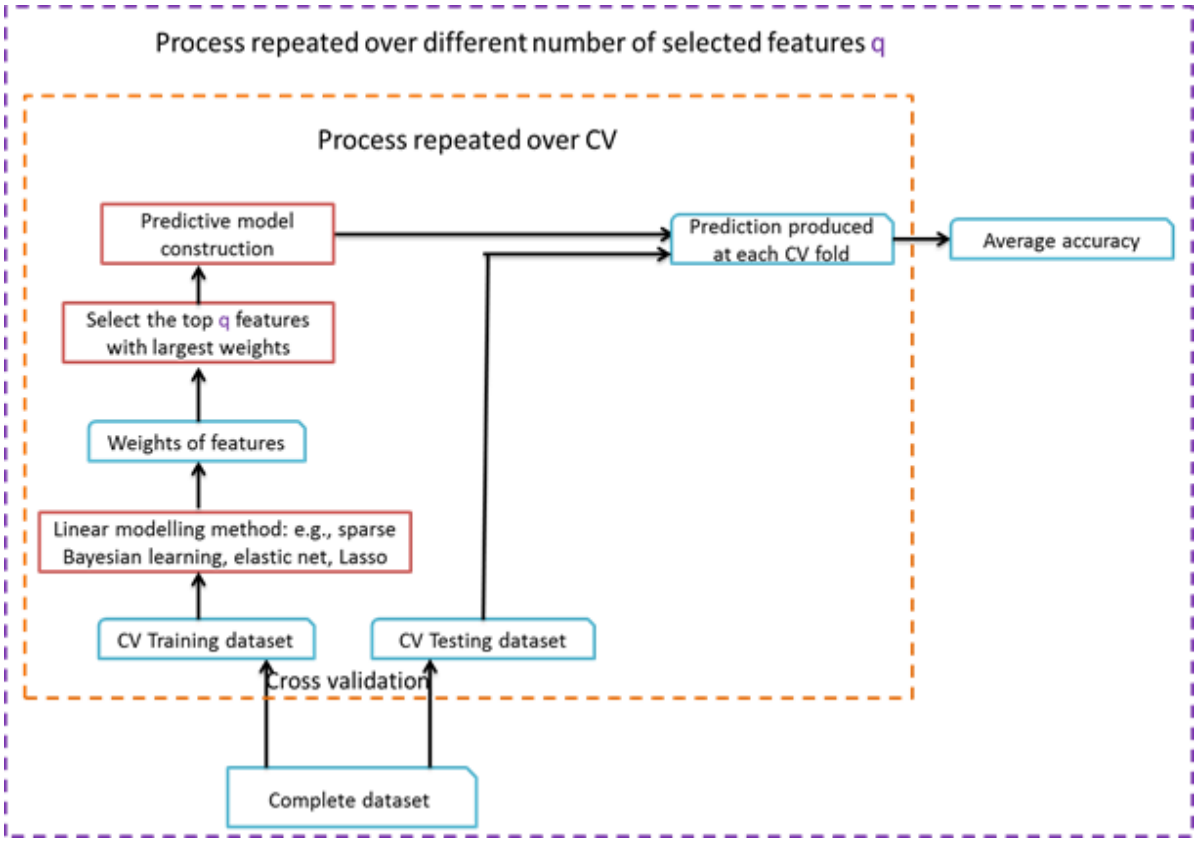


Figure 5.6: The process of cross validation with the embedded feature selection approach

5.1.5 Detecting hidden patterns behind the data

One ultimate goal of precision medicine is finding patient-tailored treatments/ drugs. We believe that many diseases (e.g, neuropsychiatric, cardiovascular, asthma, cancer and autoimmune disorders) are not single disease that patients should be treated differently based on their own characteristics[120]. Therefore, it is necessary to identify subtypes of patients for best phenotyping complex diseases. Here, we would like to discuss clustering and topological methods for hidden pattern discovery.

5.1.5.1 Clustering methods

Clustering methods, a typical unsupervised machine learning, are commonly used to find subgroups of samples, which are generally fall into two categories: feature-based clustering and similarity-based clustering[121] [122]. In translational medicine research, for example, a metabolomics dataset for patients can be represented by a matrix, where rows can be measurements of metabolite and columns represent patients. Subgroups of patients can be

detected by clustering by similar columns. If we want to find detect similar groups of metabolites and samples at the same time, we can apply biclustering methods [123], such as Cheng and Church's algorithm[124], coupled Two-way clustering[125], the iterative signature algorithm[126] and the SAMBA algorithm (Statistical-Algorithmic Method for Bicluster Analysis)[127]. In Tanay, Sharan, and Shamir 2005[128], the abovementioned methods are well discussed and their example applications in medical research can be found in [129] [130] [131].

5.1.5.2 Topological data analysis

Apart from unsupervised machine learning methods, we can also consider using topological data analysis (TDA)[132] to detect the hidden patterns of large biomedical datasets[133]. Figure 5.7 shows example results of using TDA to identify subgroups of severe asthma patients[134]. TDA constructs networks to show the hidden patterns of datasets explicitly, where patients of similar features are grouped into one node. The link between two nodes indicating they share common patients. This approach is different from clustering, allowing overlapped sub-groups of patients. It is a powerful visualization tool to help people quickly understand data. TDA is a geometric approach to shape recognition within data[135] [136]. Various machine learning and statistical analysis methods can be applied afterwards to further deep phenotype patient subgroups.

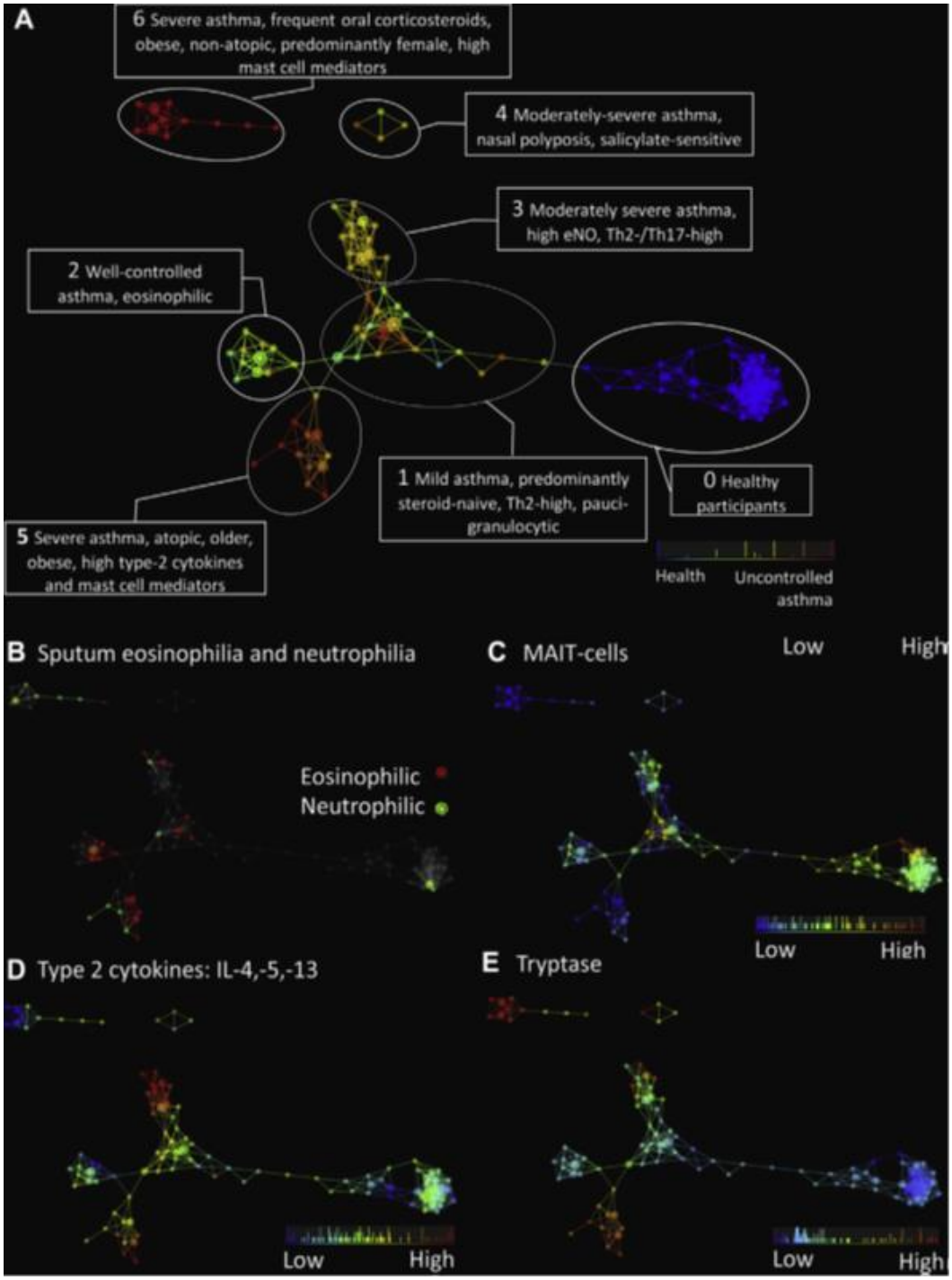


Figure 5.7: Example results generated by TDA[137]. Multidimensional clusters in asthmatic patients and healthy controls.

5.2 Solving practical data analysis issues

5.2.1 Dealing with imbalanced training datasets

In biomedical research, sample number inconsistencies across test classes are quite common. For example, it is difficult to enroll large numbers of patients into a rare disease clinical study. Even in the study of common diseases, such as asthma, the number of biopsy samples collected from healthy controls are usually much smaller than the number of samples collected from patients as healthy people are not likely to often be willing to undergo invasive procedures. Most machine learning methods suffer from imbalanced training datasets, making the prediction biased and inaccurate[138]. If the evaluation metric is not carefully chosen, the machine learning methods may only be optimized to achieve an overall prediction accuracy but not be optimized for class-specific predictions.

5.2.1.1 Evaluation metrics

Accuracy is an inappropriate metric for evaluating the performance of machine learning methods on imbalanced datasets due to the prediction accuracy being dominated by the majority class. Thus, we need an alternative evaluation of performance. Table 5.4 discusses the most common evaluation metrics. Please note that TP, FP, TN and FN represent true positive, false positive, true negative and false negative, respectively. Compared with accuracy, which is (TP+TN)/(TP+FP+TN+FN), the metrics in Table 5.4 try to give equal emphasis to imbalanced datasets.

Table 5.4: Evaluation metrics for imbalanced datasets

Evaluation metric	**Calculations**
Balanced accuracy	It computes the average of the percentages of correctly classified positives and correctly classified negatives: TP/2(TP+FN) + TN/2(TN+FP).
ROC Curve	It summarises the performance over a range of TPRs and FPRs. TPR = TP/(TP+FN); FPR = FP/(FP+TN).
Precision and Recall Curve	It summarises the performance over a range of precision and recall. Precision = TP/(TP+FP); recall = TPR.
F1 score	It measures the trade-off between precision and recall by computing 1/(1/Precision+1/Recall).

5.2.1.2 Resampling methods

The imbalanced data problem can be handled by resampling methods. The training dataset can be preprocessed to get balanced sample distributions. We can perform oversampling on the

minority class or downsampling on the majority class[139]. Table 5.5 listed out some typical resampling methods and their features.

Table 5.5: Different resampling methods and their characteristics

Resampling method	**Concept**
Random undersampling	The majority class samples are discarded at random to reach a more balanced sample distribution.
Random oversampling	The minority class samples are copied and repeated in the dataset until a more balanced sample distribution is reached.
Cluster-based oversampling	K-means clustering is performed on the minority class. Then oversampling is performed on each of the clusters to have the same number of samples, and the overall dataset to be balanced [140].
Synthetic minority oversampling	The training dataset is augmented by generating synthetic minority samples based on kNN [141].

5.2.1.3 Ensembling methods

Another way to handle the imbalanced dataset predictive model construction problem is using ensembling methods to construct several prediction models and aggregate their prediction results. There are many ensembling methods, such as bagging[142], AdaBoos[143], Random Forest[144] and gradient boosting. Table 5.6 briefly explains their concepts and characteristics.

Table 5.6: Ensembling methods and their characteristics

Ensembling method	**Concept**
Bagging	It starts with generating N bootstrapped training sample sets with replacement. Then N predictors are constructed using each bootstrapped dataset separately. Their prediction results are aggregated at the end.
Random Forest	It is similar with Bagging methods. The only difference is that each base learner is constructed on random selection of features.
Adaboost	It fits a sequence of week learners on repeated modified versions of the data. The predictions are aggregated through a weighted majority vote.
Gradient boosting	It is similar with Adaboost in constructing a strong learning from a set of weak learners. The way of creating weak learners is different. Instead of training on a new sample distribution, weak learners are trained on residual errors.

5.2.2 Dealing with small training datasets

The lack of training data often results in overfitting when we train a model. The performance of data analysis methods is consequently reduced. We can use ensemble methods as discussed above to build a strong predictor from weak learners. Along with this kind of methods, we can use transfer learning methods as discussed below.

Reusing knowledge from other auxiliary domains where the data is annotated is one idea for overcoming the problem of a small training dataset. This framework is called transfer learning[145]. Here, we would like to give an example of transfer learning in precision medicine research, where the advance machine learning technique, deep learning, is used for disease classification. The convolutional neural network (CNN), a typical architecture of deep learning, can provide great advances in extracting information from medical imaging data. However, it suffers from small training datasets. The work in Huynh, Li, and Giger 2016[146] uses transfer learning to classify mammographic tumors from medical images via CNNs originally pretrained for non-medical tasks. It is based on the assumption that structures within a CNN trained on everyday objects could be used to create a classifier for breast cancer computer aided diagnosis. AlexNet[147], a CNN model with three fully connected layers and five convolutional layers, is used to extract features from images. As it is unclear which layer of AlexNet would best fit the classification of breast tumor images, the output of each layer is fed into the classifier to find the optimal layer. The overview of the classification methodology in Huynh, Li, and Giger 2016[148] is shown in Figure 5.8. The feature extraction step is implemented through two different approaches: method A uses features from a pretrained CNN while method B extracts features via segmented-tumor-based analytical methods[149] [150] [151]. SVM is used to construct classifiers from features sets generated by different methods. An ensemble classifier is also used to average individual classifiers (method C)[152]. The classification results show that classifiers based on method A perform comparably to the one using method B and method C outperforms the others, showing that transfer learning can improve computer-aided diagnosis methods without the requirement of large training datasets.

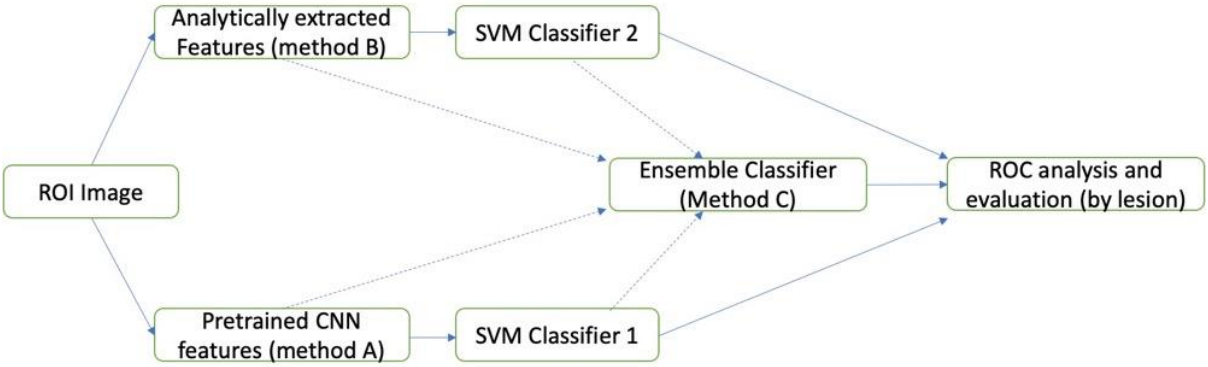


Figure 5.8 The overview of feature extraction and classification methods for mammographic tumour classification[153].

5.2.3 Dealing with partially labelled datasets

Having sufficient labelled data can be an issue in supervised learning and possibly unlabeled data can help with the prediction. There are many of hybrid techniques capable of learning both from labelled and unlabeled data. In Bogdan Gabrys and Petrakieva 2004[154], these hybrid methods are categorized into three groups: pre-labelling, post-labelling and semi-supervised approaches. Each category is summarized in Table 5.7.

Table 5.7: The main categories of training methods for partially labelled datasets

Approach category	Concept	Example methods
Pre-labelling	An initial classifier is constructed using labelled data first. Then, it is used to label the unlabeled data. After this has been done, a new classifier is constructing using both the original and newly labelled data.	Dara, Kremer, and Stacey, n.d. [155], Mitchell 2004 [156], Nigam and Ghani 2000 [157], Nigam et al. 1998[158]
Post-labelling	A data model is constructed using all available data with the application of a data density estimation procedure or clustering algorithm. Then, labelled data is then	Ghahramani and Jordan 1994 [159], Kothari and Jain, n.d.[160]

	used to label whole clusters of data by counting the number of labelled samples from specific classes within each of the clusters.	
Semi-supervised	Both labelled and unlabeled data are process at the same time. This method sit somewhere between pre-labelling and post-labelled approaches: the clustering process is constrained by the labelled data and the classification process takes into account of unlabeled data.	B. Gabrys and Bargiela 2000 [161] , Pedrycz et al. 2008 [162] , Bogdan Gabrys 2002[163]

5.2.4 Dealing with the out of memory problem caused by big data

The datasets collected in biomedical research are very large with respect to sample size and feature dimension. Traditional ways of using machine learning methods on large datasets require correspondingly large amounts of memory and may result in the "out of memory" problem. There are many ways to overcome this problem; for example, we can use parallel computing platform such as SPARK to process big data. Figure 5.9 from L'Heureux et al. 2017[164] shows the main manipulations for big data using parallel computing platforms. We can also consider reducing the feature space (e.g., PCA) or use online machine learning methods. The online machine learning methods can learn incrementally from mini batches of instances in which only a small amount of samples are loaded in the memory. There are many online learning methods implemented in popular machine learning toolkits. For example, the Python scikit-learn package supports the following four main categories of online learning algorithms:

1. classification -- perceptron, SGD classifier and Naive Bayes classifier;
2. regression -- SGD regressor and Passive aggressive regressor;
3. clustering -- mini-batch k-means;
4. feature extraction -- mini-batch dictionary learning and incremental PCA.

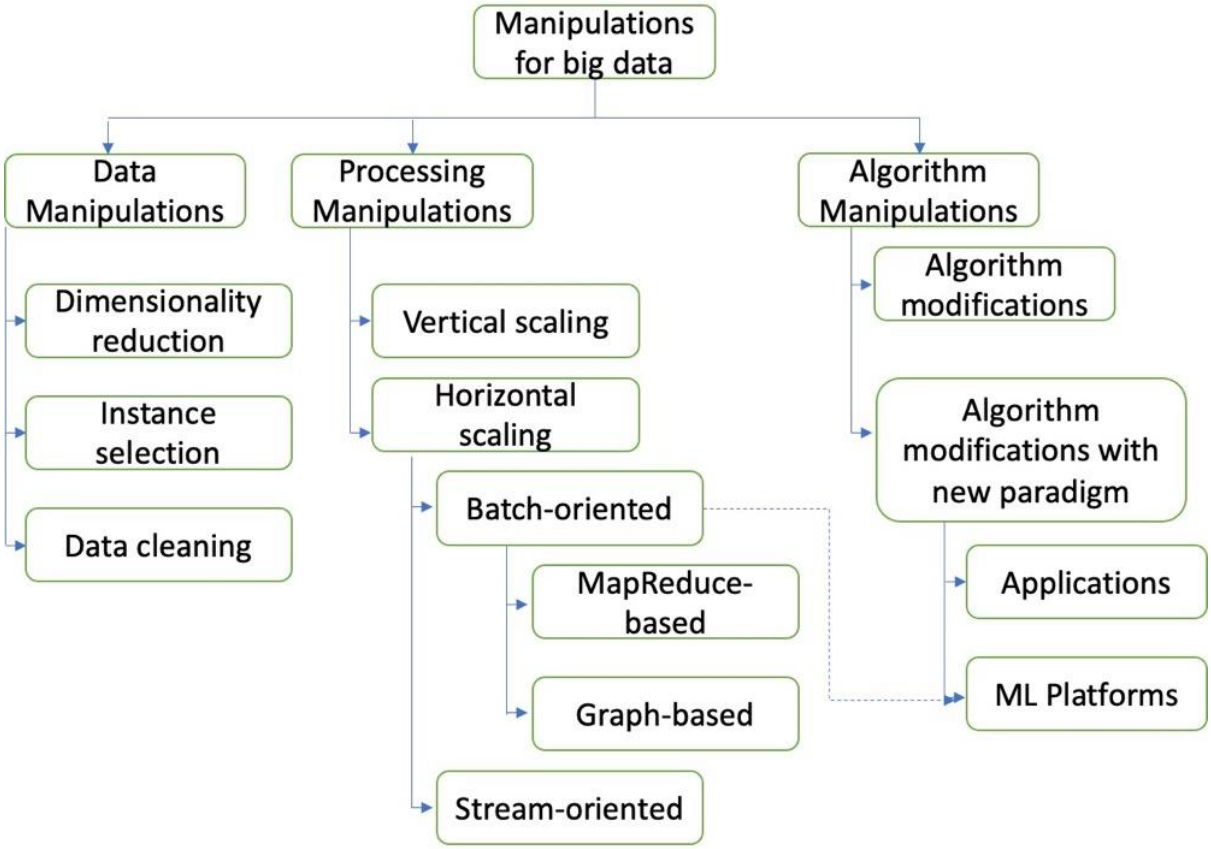


Figure 5.9: Manipulations for big data using parallel computing platforms.

5.2.5 Constructing models from datasets having both continuous and categorical variables

To choose the machine learning algorithm for predictive model construction, the first step is understanding the data type. The datasets might only contain numerical features, categorical features or both numerical and categorical features. For numerical data, there are many machine learning methods from which to choose, such as decision trees, naive bayes, SVM, logistic regression, ensemble methods (bagging, boosting), Random forest and multi-layer perceptron. For categorical data, the most common machine learning methods are naive bayes, decision trees and their ensembles such as Random forest, minimum distance classifiers or KNN with hamming distance. For the dataset with both numerical and categorical features, we could choose decision trees and its ensembles, and KNN based approach with the cost function carefully designed to handle data for both types together.

5.2.6 Dealing with correlated features

In biomedical research, the datasets could have highly correlated features, such as gene expression data. These correlated features could affect the performance of some data analysis algorithms. For example, linear regression models, such as linear regression and logistic regression, are based on the assumptions that features are independent. Multicollinearity could yield solutions numerically unstable and widely varying. For this reason, the severity of

multicollinearity is quantified by calculating the variance inflation factor (VIF). For decision tree-based methods, which are good at detecting interactions between different features, correlations among features would mask these interactions. Feature selection or reduction methods could be applied to reduce the correlation among features. One typical approach is PCA. Alternatively, correlation values between features could be calculated followed by removal of certain highly correlated features.

5.3 Biomarker discovery

5.3.1 What is biomarker?

A biomarker (also sometimes referred to as a molecular marker or signature) is a molecule, gene or other physiologic characteristic that, when measured, can be used as an indicator of a given pathophysiological process, disease, or disease subclass[165]. As such, biomarkers are fundamental to biomedical research for identifying or classifying disease sufferers, monitoring disease states, assessing responses to treatment or as potential intervention targets. A biomarker may simply be a clinical signal such as blood pressure or triglyceride levels. However, increasingly biomarkers rely on sophisticated technological measures of molecular physiology such as patterns of gene expression or changes in neural electrical activity. Ideally, biomarker measurements should be objective, safe and easy to collect, respond rapidly and sensitively to biological changes and remain consistent across cohorts of subjects that share medically relevant physical traits.

Biomarkers have been especially important with respect to cancer treatment. The genetic abnormalities that underlie the development of cancer can be detected objectively via certain DNA and RNA markers to aid in precise diagnosis. Further, biomarkers can lead to more sophisticated therapies that specifically target cancer cells while sparing healthy cells.

The increasing recognition of physiologic heterogeneity of many diseases and the importance of personal factors and life history has heightened the importance of identifying and applying informative biomarkers in contemporary medical research and practice.

5.3.2 Discovering biomarkers

The pathway to biomarker discovery and validation "is a work in progress and is evolving"[166], although guidelines have been suggested for some domains[167] and form a useful framework.

The discovery process for new biomarkers can be broadly divided into two contrasting approaches: data-driven (also "statistical", "discovery-based", "untargeted", or "un-biased") versus hypothesis-driven ("knowledge-based" or "targeted"). The former assesses biomarker candidates without any *a priori* selection or prioritization and tests these candidates in parallel,

perhaps using statistical approaches to extract the best candidates. In comparison, the latter uses contextual and mechanistic knowledge to winnow the universe of possible targets to a subset of probable candidates. While the distinction is not perfect—discovery pipelines may use a mixture of both—the division is useful for categorizing methodologies and methodological issues.

5.3.3 What are challenges of biomarker discovery?

Despite the clear need for biomarkers, intense efforts/investment to identify new biomarkers and copious data generated by high throughput technologies the number of clinically validated biomarkers is rather modest[168].

The advent of high-throughput omics technologies, in which thousands of potential targets can be easily interrogated without a priori assumptions, accelerates hypotheses generation leading to biologic insights. However, extracting meaningful molecular signatures from such dense datasets poses computational challenges[169]

The lack of gene set overlap between two FDA-approved transcriptome signatures for node negative breast cancer prognosis, and other similar examples, [170] raises concerns regarding the ability of purely statistical approaches to produce consistent findings. Many investigators are evaluating combinations of biomarkers in hopes of attaining suitable sensitivity and specificity for clinical application.

Another potential source of problems lies in the study population used for biomarker discovery. Many populations are assembled through convenience without the intent of pursuing specific biomarker identification and, correspondingly, are selected without pertinent inclusion and exclusion criteria. Research using such populations may be susceptible to confounding factors resulting in false positives.

5.4 System Biology approaches

5.4.1 Typical data-level integrative analysis methods

5.4.1.1 WGCNA

Weighted correlation network analysis (WGCNA)[171] is widely used in high dimensional data analysis to study relationships between co-expressed modules (e.g., correlated gene clusters) and with external sample traits. The basic idea of WGCNA is summarized in Figure 5.10. The first step in WGCNA is to construct a gene co-expression network based on the correlations. The next step is identifying modules from the correlation network. Modules are defined as

interconnected genes in the network, where the interconnectivity is measured by the topological overlap measure. WGCNA uses unsupervised clustering to identify modules. The next step is finding biological or clinically significant modules. Functional enrichment analysis can be used to detect pathway memberships. Statistical significance tests can be used to detect trait associated modules. To summarize the gene expression profiles of a given model, WGCNA uses the first principle component of the expression matrix of a module as the eigengene. An Eigengene network is then generated to study module relationships. If we find any interesting modules, we could carry out experiments to understand the drivers of these modules.

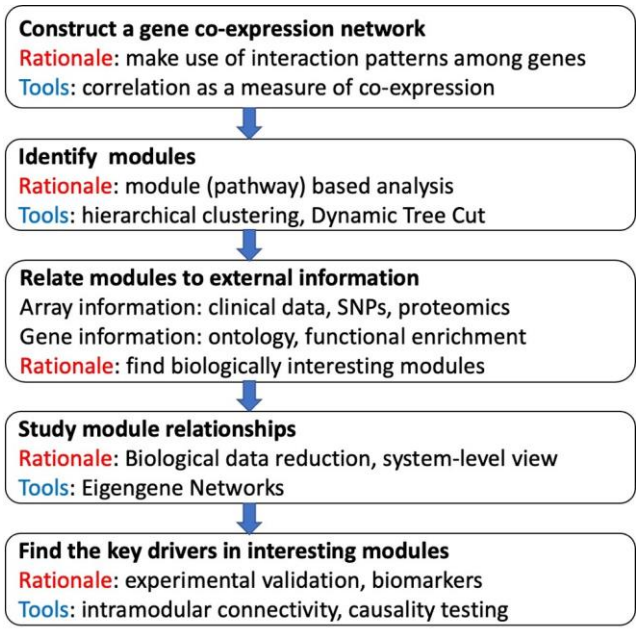


Figure 5.10: The main steps of the WGCNA method (from Langfelder and Horvath 2008[172]).

5.4.1.2 GSVA

Gene Set Variation Analysis (GSVA) is a non-parametric, unsupervised method which estimates the relative enrichment of a gene set of interest across a sample population[173174]. Hence, it allows us to observe the variation in the activity of a set of genes, such as a pathway or a gene signature, that corresponds to a specific biological condition. It produces a value, an *enrichment score* (ES), per sample and gene set, which can be examined for associations with clinical features of interest. The input of GSVA can be a gene expression matrix in the form of microarray expression values or RNA-seq counts. The kernel estimation of the cumulative density function is then performed to estimate the expression level statistic, which is then ranked for each sample. For each gene set, the KS-like random statistic is calculated. The gene set enrichment score can be either calculated as the maximum deviation from zero or difference between two sums. The output of GSVA contains a pathway enriched score for each gene set and sample.

5.4.1.3 SNF

There are many computational methods to integrate multiple datasets together. In Huang, Chaudhary, and Garmire 2017[175], both unsupervised and supervised data integration methods are discussed. There are mainly five categories of unsupervised data integration methods [176], which are matrix factorization, Bayesian, network-based, multiple kernel learning and multi-step analysis. Here, we would like to discuss one typical network-based approach, Similarity Network Fusion (SNF), that we used in the U-BIOPRED project for multiple Omics data integration. SNF fuses diverse types of genomics datasets in a cost-efficient manner, analyzing different layers of biology on the same patients, clustering patients based on this fused matrix[177]. The method uses networks of samples generated from different Omics data types as the basis for data integration. SNF returns a single similarity network that captures both shared and complementary information from different data sources. Figure 5.11 (from [178]) shows illustrative steps of SNF. In this example, two types of datasets, mRNA expression and DNA methylation, for the same patient cohort are loaded into SNF (as in Figure 5.11a). Then, similarity matrices, as it is shown in Figure 5.11b, are constructed for each data types. Patient similarity networks are then built using similarity matrices with weighted edges representing pairwise sample similarities (see 5.11c). The network fusion step in 5.11d applies a nonlinear method based on message-passing theory to iteratively update the network and make these similar[179]. As a result, weak similarities are removed while strong similarities shared among networks are retained.

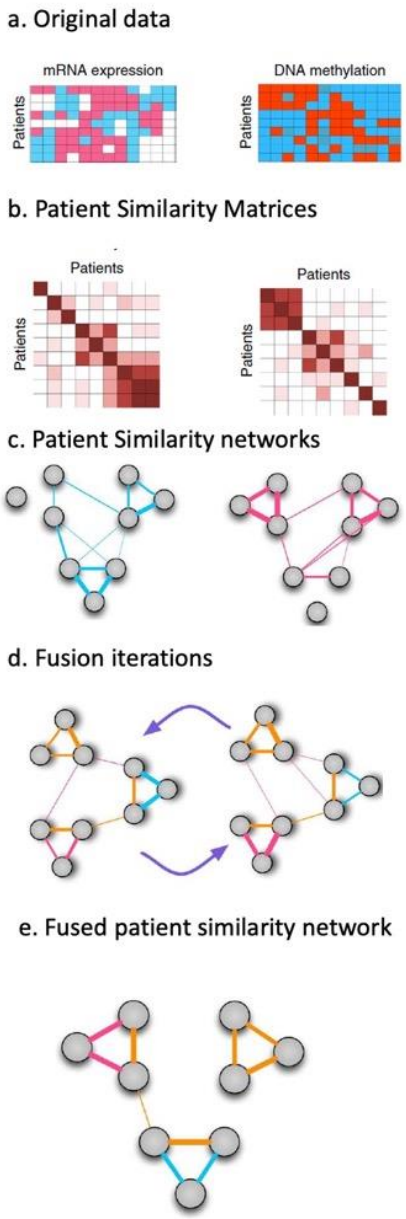


Figure 5.11: Illustrative steps of SNF

5.4.1.4 Deep learning

In recent decades, deep learning has attained great success in the areas of computer vision, remote sensing, natural language processing and bioinformatics. With the massive accumulation of Omics and healthcare data, deep learning has been increasingly used in precision medicine research, such as biomarker identification and drug discovery[180]. Deep learning algorithms are based on the use of compositional layers of neurons which can be successfully applied in disease subtyping. For example, Tan et al. 2015[181] utilized deep learning methods to categorize breast cancer patients using the information extracted from genome-wide assays. Lasko, Denny, and Levy 2013[182] combined sparse autoencoders and

Gaussian processes to distinguish gout from leukemia using uric acid sequences. Liang et al. 2015[183] developed a multimodal deep belief network for ovarian cancer patients clustering using genomic data. Miotto et al. 2016[184] presented a three-layer stack denoising autoencoder to derive a general-purpose patient representation from electronic health records. Feature construction via deep learning approaches has been shown to efficiently reduce the training data size for subsequently supervised analyses[185].

Deep learning has also been applied to integrate multiple datasets in biomedical research as well. In Liang M n.d[186], a multimodal deep belief network (DBN) is used for data integration. The basic idea is shown in Figure 5.12 (from Liang M n.d.[187]). It first uses a restricted Bolzmann machine (RBM) to encode latent features defined by each input dataset. Then, hidden variables from different modularities are fused together using the contrastive divergence (CD) algorithm. Finally, the joint representation of features is used for predictive model construction.

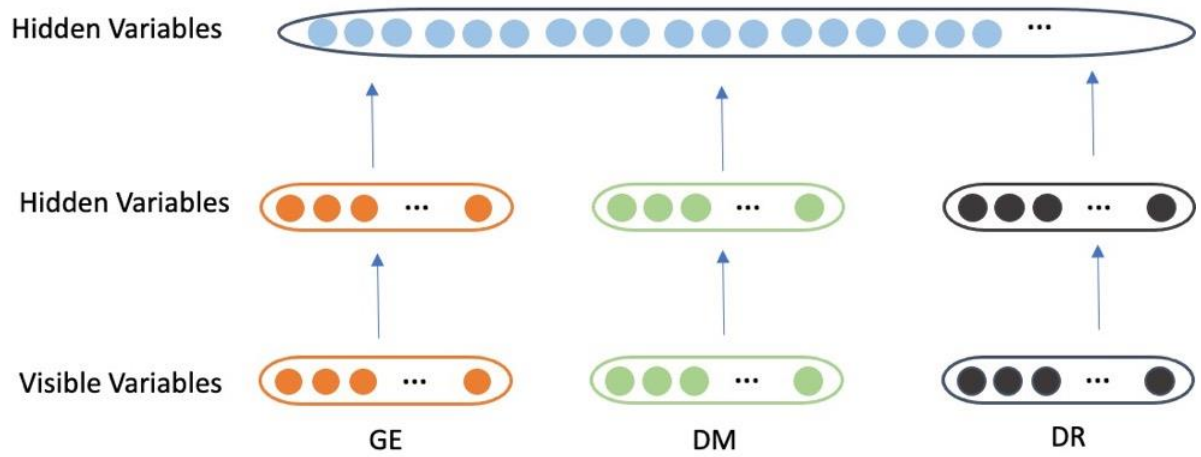


Figure 5.12: An example of a multimodal deep learning model for data integration (GE: gene expression, DM: DNA methylation, DR: drug response).

5.5 Disease maps

5.5.1 Brief overview

A disease map is defined as a pathway-based computational representation of disease mechanisms. It is a conceptual model built as a reflection of the published papers and inputs from domain experts. A disease map contains "disease-related signaling, metabolic and gene regulatory processes with evidence of their relationships to pathophysiological causes"[188].

The concept of disease maps was first introduced with comprehensive reconstructions of disease mechanisms for Parkinson's disease, Alzheimer's disease, influenza and cancer[189 190 191 192], further developed within the eTRIKS Disease Maps Lab[193] and evolved into the Disease Maps Community[194 195 196].

The Disease Maps Community brings together multiple groups from many countries and is quickly growing into a larger international effort. The involved projects are focused on building maps for specific diseases and developing the necessary supporting infrastructure and tools. One theme is consolidating resource into a centralized repository. This topic was extensively discussed during the 3rd Disease Maps Community Meeting (DMCM2018 [197]), and a combination of the Biological Pathway Exchange format[198] and the Neo4j Graph Database environment[199 200] is suggested as a possible solution.

Most of the current maps in the responsibility of the authors of this section are developed in CellDesigner (http://www.celldesigner.org) with the possibility of providing all of these maps in the Systems Biology Graphical Notation [201] (SBGN), Biological Pathway Exchange (BioPAX), the Systems Biology Markup Language (SBML)[202] and image formats.

The involvement of domain experts from several independent groups is an extremely important aspect of constructing high quality disease maps. Such a team is organized as a joint effort across several groups coordinated by the leader of the map development.

5.5.2 Computational approaches for disease maps

Computational modelling approaches for diseases are presented in the following chapter. In this section computational approaches suited specifically for disease maps are introduced addressing their advantages and pitfalls. Two major directions are described:

1. network analysis for disease maps
2. computational modelling, from static representation to dynamic exploration

5.5.2.1 Network analysis for disease maps

Given the nature of mechanistic representations within disease maps, network-based approaches are directly suitable for disease map modelling. Networks can be used, for example, to identify key interactions within disease maps and to explore the impact of their individual or joint alterations on disease progression. Sensitivity of drugs can be also predicted by investigating the network topology and analyzing perturbations induced by a chosen combination of molecular drug targets within a given disease map. Networks can be also used for comparative analyses between various disease types or stages and identification of the common sets of molecular mechanisms and modules across different pathological contexts. For example, a recent network-based comparison performed between Parkinson's and age-related diseases is presented in the work by Glaab and coauthors[203].

5.5.2.2 Modelling: from static representation to dynamic exploration

While a disease map offers a static representation of current domain-expert knowledge on specific stages of a disease or a subtype of a disease, computational modelling approaches assist in understanding dynamic features leading to disease initiation and progression. Computational models of disease maps can be used, for example, to make predictions on disease progression and on medication efficacy, to identify candidates for drug repurposing, to refine and validate existing hypotheses and to postulate new hypotheses towards identifying improved therapeutic solutions. While a computational model for a given disease map can be seen as a powerful means to gain further insight into the disease's dynamic aspects, its development depends highly on the level of detail and on the quality of information integrated into an initial map. This process may be a complex requiring additional steps such as the creation of a repository with quantitative experimental data. Moreover, the development of the model should be driven by clinical questions and needs to facilitate its validation through clinical studies.

Examples of computational approaches suitable for disease map modelling include:

1. logic models, (such as deterministic and stochastic Boolean networks) and rule-based approaches for signaling and regulatory networks
2. steady-state approaches (e.g. flux balance analysis) for metabolic networks
3. quantitative kinetic models, if kinetic information is available. Given the recent technical developments, we foresee that the integration of disease maps into multi-scale modelling approaches, which span specifics from the molecular level through cellular scale to organ and organism levels and permit inclusion of experimental data at all system levels, exploration of inter-scale phenomena relationships and analysis of perturbations at system level, becomes achievable.

Summary

This chapter provided a comprehensive overview of the wide breadth of analytical concepts and methods pertinent to translational research. General strategies for imputing missing values,

addressing confounding variables and performing statistical inference led to explanations of specialized methods for addressing pattern recognition in large scale datasets, including those commonly used for clustering and classification. Topics pertinent to supervised learning methods, including training set balancing and feature selection, were introduced with an emphasis on reviewing contemporary biomedical deep learning applications. Finally, this chapter reviewed system biology approaches for gaining biomedical insights including molecular network representations and analysis as well as the creation of consolidated disease maps. This review included several methods that were developed by the chapter authors. It is hoped that readers will repeatedly refer to this comprehensive review of computational approaches to translational research for guidance regarding analytical approaches to exploratory clinical studies.

Chapter 5 References:

[1] Petricoin EF et al. 2002. Use of Proteomic Patterns in Serum to Identify Ovarian Cancer. The Lancet. PMID: 11867112

[2] Spielman RS et al. 2007. Common Genetic Variants Account for Differences in Gene Expression among Ethnic Groups. Nat. Genet. PMID: 17206142

[3]Liotta LA. 2004. High-Resolution Serum Proteomic Patterns for Ovarian Cancer Detection. Endocr-Relat Cancer. PMID: 15163296

[4] Leek JT et al. 2010. Tackling the Widespread and Critical Impact of Batch Effects in High-Throughput Data. Nat. Genet. PMID: 20838408

[5] Dyrskjøt L et al. 2004. Gene Expression in the Urinary Bladder: A Common Carcinoma in Situ Gene Expression Signature Exists Disregarding Histopathological Classification. Cancer Res. PMID: 15173019

[6] Yang H et al. 2008. Randomization in Laboratory Procedure Is Key to Obtaining Reproducible Microarray Results. PloS One. PMID: 9009020

[7] Holmes S et al. 2011. Visualization and Statistical Comparisons of Microbial Communities Using R Packages on Phylochip Data. Pacific Symposium on Biocomputing. PMID: 21121042

[8] Jolliffe IT et al. 2009. Principal Component Analysis: A Review and Recent Developments. PHILOS. T. R. SOC. A. PMID: 26953178

[9] Desdouits N et al. 2015. Principal Component Analysis Reveals Correlation of Cavities Evolution and Functional Motions in Proteins. J. Mol. Graph.

[10]Alonso-Gutierrez J et al. 2015. Principal Component Analysis of Proteomics (PCAP) as a Tool to Direct Metabolic Engineering. Metab. Eng. PMID: 25554074

[11] Zhang JD et al. 2015. Pathway Reporter Genes Define Molecular Phenotypes of Human Cells. BMC Genomics. PMID: 25903797

[12] Abraham G, and Inouye M. 2014. Fast Principal Component Analysis of Large-Scale Genome-Wide Data. PloS One. PMID: 24718290

[13] Reese SE et al. 2013. A New Statistic for Identifying Batch Effects in High-Throughput Genomic Data That Uses Guided Principal Component Analysis. Bioinformatics. PMID: 23958724

[14] Oytam Y et al. 2016. Risk-Conscious Correction of Batch Effects: Maximising Information Extraction from High-Throughput Genomic Datasets. BMC Bioinformatics. PMID: 27585881

[15] Fahad A et al. 2014. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. IEEE.

[16] Alter O et al. 2000. Singular Value Decomposition for Genome-Wide Expression Data Processing and Modeling. Proceedings of the National Academy of Sciences of the United States of America. PMID: 10963673

[17] Nielsen TO et al. 2002. Molecular Characterisation of Soft Tissue Tumours: A Gene Expression Study. The Lancet. PMID: 1965276

[18] Benito M et al. 2004. Adjustment of Systematic Microarray Data Biases. Bioinformatics. PMID: 14693816

[19] Johnson WE et al. 2007. Adjusting Batch Effects in Microarray Expression Data Using Empirical Bayes Methods. Biostatistics. PMID: 16632515

[20] Scherer A. Batch Effects and Noise in Microarray Experiments: Sources and Solutions. Hoboken (NJ). John Wiley & Sons, 2009. ISBN: 9780470741382

[21] Leek JT and Storey JD. 2007. Capturing Heterogeneity in Gene Expression Studies by Surrogate Variable Analysis. PLoS Genetics. PMID: 17907809

[22] Liew AWC et al. 2011. Missing Value Imputation for Gene Expression Data: Computational Techniques to Recover Missing Data from Available Information. Briefings in Bioinformatics. PMID: 21156727

[23] Libbrecht MW, and Stafford Noble W. 2015. Machine Learning Applications in Genetics and Genomics. Nat. Genet. PMID: 25948244

[24] Luengo J et al. 2011. On the Choice of the Best Imputation Methods for Missing Values Considering Three Groups of Classification Methods. Knowledge and Information Systems.

[25] Acuña E and Rodriguez C. 2004. The Treatment of Missing Values and Its Effect on Classifier Accuracy. In Classification, Clustering, and Data Mining Applications, pp 639–47

[26] Aittokallio T. 2010. Dealing with Missing Values in Large-Scale Studies: Microarray Data Imputation and beyond. Briefings in Bioinformatics. PMID: 19965979

[27] Little RJA. Statistical Analysis With Missing Data. Hoboken (NJ). John Wiley & Sons, 1987.

[28] Dempster, Laird and Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological). Vol. 39, No. 1 (1977). pp. 1-38

[29] Aittokallio T. 2010. Dealing with Missing Values in Large-Scale Studies: Microarray Data Imputation and beyond. Briefings in Bioinformatics. PMID: 19965979

[30] Tuikkala JL et al. 2005. Improving Missing Value Estimation in Microarray Data with Gene Ontology. Bioinformatics. PMID: 16377613

[31] Elo LL et al. "Predicting Gene Expression from Combined Expression and Promoter Profile Similarity with Application to Missing Value Imputation." In Modeling and Simulation in Science, Engineering and Technology. Boston (MA). Birkhäuser, 2007. doi: 10.1007/978-0-8176-4558-8_9

[32] Xiang Q et al. 2008. Missing Value Imputation for Microarray Gene Expression Data Using Histone Acetylation Information. BMC Bioinformatics. PMID: 8510747

[33]Aittokallio T. 2010. Dealing with missing values in large-scale studies: microarray data imputation and beyond. Brief Bioinform. PMID: 19965979

[34] Moorthy K et al. 2014. A Review on Missing Value Imputation Algorithms for Microarray Gene Expression Data. Current Bioinformatics.

[35] Friedland, Shmuel, Amir Niknejad, and Laura Chihara. 2006. "A Simultaneous Reconstruction of Missing Data in DNA Microarrays." Linear Algebra and Its Applications.

[36] Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. 2001. "Missing Value Estimation Methods for DNA Microarrays." Bioinformatics.

[37] Oba, Shigeyuki, Masa-Aki Sato, Ichiro Takemasa, Morito Monden, Ken-Ichi Matsubara, and Shin Ishii. 2003. "A Bayesian Missing Value Estimation Method for Gene Expression Profile Data." Bioinformatics.

[38] Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. 2001. "Missing Value Estimation Methods for DNA Microarrays." Bioinformatics.

[39] Kim, H., G. H. Golub, and H. Park. 2006. "Missing Value Estimation for DNA Microarray Gene Expression Data: Local Least Squares Imputation." Bioinformatics.

[40] Brás, L. P., and J. C. Menezes. 2006. "Dealing with Gene Expression Missing Data." IEE Proceedings - Systems Biology.

[41] Zhou, X., X. Wang, and E. R. Dougherty. 2003. "Missing-Value Estimation Using Linear and Non-Linear Regression with Bayesian Gene Selection." Bioinformatics.

[42] Bo, T. H. 2004. "LSimpute: Accurate Estimation of Missing Values in Microarray Data with Least Squares Methods." Nucleic Acids Research.

[43] Kim, H., G. H. Golub, and H. Park. 2006. "Missing Value Estimation for DNA Microarray Gene Expression Data: Local Least Squares Imputation." Bioinformatics.

[44] Cai, Zhipeng, Maysam Heydari, and Guohui Lin. 2006. "Iterated Local Least Squares Microarray Missing Value Imputation." Journal of Bioinformatics and Computational Biology.

[45] Ouyang, Ming, William J. Welsh, and Panos Georgopoulos. 2004. "Gaussian Mixture Clustering and Imputation of Microarray Data." Bioinformatics

[46] Sehgal, Muhammad Shoaib B., Iqbal Gondal, and Laurence S. Dooley. 2005. "Collateral Missing Value Imputation: A New Robust Missing Value Estimation Algorithm for Microarray Data." Bioinformatics

[47] Sehgal, Muhammad Shoaib B., Iqbal Gondal, Laurence S. Dooley, and Ross Coppel. 2008. "Ameliorative Missing Value Imputation for Robust Biological Knowledge Inference." Journal of Biomedical Informatics

[48] Colantonio, Alessandro, Roberto Di Pietro, Alberto Ocello, and Nino Vincenzo Verde. 2010. "ABBA." In Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10. https://doi.org/10.1145/1774088.1774304

[49] Jörnsten, Rebecka, Hui-Yu Wang, William J. Welsh, and Ming Ouyang. 2005. "DNA Microarray Data Imputation and Significance Analysis of Differential Expression." Bioinformatics

[50] Aittokallio, Tero. 2010. “Dealing with Missing Values in Large-Scale Studies: Microarray Data Imputation and beyond.” Briefings in Bioinformatics

[51] Van Oudenhove, Laurence, and Bart Devreese. 2013. “A Review on Recent Developments in Mass Spectrometry Instrumentation and Quantitative Tools Advancing Bacterial Proteomics.” Applied Microbiology and Biotechnology

[52] Waters, Katrina M., Joel G. Pounds, and Brian D. Thrall. 2006. “Data Merging for Integrated Microarray and Proteomic Analysis.” Briefings in Functional Genomics & Proteomics

[53] Goh, Wilson W. B., Yie H. Lee, Maxey Chung, and Limsoon Wong. 2012. “How Advancement in Biological Network Analysis Methods Empowers Proteomics.” Proteomics

[54] Goh, Wilson Wen Bin, Marek J. Sergot, Judy C. G. Sng, Judy Cg Sng, and Limsoon Wong. 2013. “Comparative Network-Based Recovery Analysis and Proteomic Profiling of Neurological Changes in Valproic Acid-Treated Mice.” Journal of Proteome Research

[55] Pavelka, Norman, Marjorie L. Fournier, Selene K. Swanson, Mattia Pelizzola, Paola Ricciardi-Castagnoli, Laurence Florens, and Michael P. Washburn. 2008. “Statistical Similarities between Transcriptomics and Quantitative Shotgun Proteomics Data.” MCP

[56] Li, Feng, Lei Nie, Gang Wu, Jianjun Qiao, and Weiwen Zhang. 2011. “Prediction and Characterization of Missing Proteomic Data in Desulfovibrio Vulgaris” Comparative and Functional Genomics

[57] Webb-Robertson, Bobbie-Jo M., Holli K. Wiberg, Melissa M. Matzke, Joseph N. Brown, Jing Wang, Jason E. McDermott, Richard D. Smith, et al. 2015. “Review, Evaluation, and Discussion of the Challenges of Missing Value Imputation for Mass Spectrometry-Based Label-Free Global Proteomics.” Journal of Proteome Research

[58] Karpievitch, Yuliya, Jeff Stanley, Thomas Taverner, Jianhua Huang, Joshua N. Adkins, Charles Ansong, Fred Heffron, et al. 2009. “A Statistical Framework for Protein Quantitation in Bottom-up MS-Based Proteomics.” Bioinformatics

[59] Emerson, John W., Marisa Dolled-Filhart, Lyndsay Harris, David L. Rimm, and David P. Tuck. 2009. “Quantitative Assessment of Tissue Biomarkers and Construction of a Model to Predict Outcome in Breast Cancer Using Multiple Imputation.” Cancer Informatics

[60] Torres-García, Wandaliz, Weiwen Zhang, George C. Runger, Roger H. Johnson, and Deirdre R. Meldrum. 2009. “Integrative Analysis of Transcriptomic and Proteomic Data of

Desulfovibrio Vulgaris: A Non-Linear Model to Predict Abundance of Undetected Proteins." Bioinformatics

[61] Liao, Serena G., Yan Lin, Dongwan D. Kang, Divay Chandra, Jessica Bon, Naftali Kaminski, Frank C. Sciurba, and George C. Tseng. 2014. "Missing Value Imputation in High-Dimensional Phenomic Data: Imputable or Not, and How?" BMC Bioinformatics

[62] Warner, Jeremy L., Gil Alterovitz, Kelly Bodio, and Robin M. Joyce. 2013. "External Phenome Analysis Enables a Rational Federated Query Strategy to Detect Changing Rates of Treatment-Related Complications Associated with Multiple Myeloma." JAMIA

[63] Denny, Joshua C., Marylyn D. Ritchie, Melissa A. Basford, Jill M. Pulley, Lisa Bastarache, Kristin Brown-Gentry, Deede Wang, Dan R. Masys, Dan M. Roden, and Dana C. Crawford. 2010. "PheWAS: Demonstrating the Feasibility of a Phenome-Wide Scan to Discover Gene-Disease Associations." Bioinformatics

[64] Lyalina, Svetlana, Bethany Percha, Paea LePendu, Srinivasan V. Iyer, Russ B. Altman, and Nigam H. Shah. 2013. "Identifying Phenotypic Signatures of Neuropsychiatric Disorders from Electronic Medical Records." JAMIA

[65] Hanauer, David A., and Naren Ramakrishnan. 2013. "Modeling Temporal Relationships in Large Scale Clinical Associations." JAMIA

[66] Lasko, Thomas A., Joshua C. Denny, and Mia A. Levy. 2013. Correction: Computational Phenotype Discovery Using Unsupervised Feature Learning over Noisy, Sparse, and Irregular Clinical Data. PloS One 8. https://doi.org/10.1371/annotation/0c88e0d5-dade-4376-8ee1-49ed4ff238e2.

[67] Wells, Brian J., Kevin M. Chagin, Amy S. Nowacki, and Michael W. Kattan. 2013. "Strategies for Handling Missing Data in Electronic Health Record Derived Data." EGEMS

[68] Burgette, Lane F., and Jerome P. Reiter. 2010. "Multiple Imputation for Missing Data via Sequential Regression Trees." American Journal of Epidemiology

[69] Stekhoven, Daniel J, Peter Bühlmann. 2012. "MissForest--Non-Parametric Missing Value Imputation for Mixed-Type Data." Bioinformatics

[70] Liao, Serena G., Yan Lin, Dongwan D. Kang, Divay Chandra, Jessica Bon, Naftali Kaminski, Frank C. Sciurba, George C. Tseng. 2014. "Missing Value Imputation in High-Dimensional Phenomic Data: Imputable or Not, and How?" BMC Bioinformatics

[71] Ibidem

[72] Olsson U et al. 1982. The polyserial correlation coefficient. Psychometrika. Volume 47

[73] Ibidem

[74] Olsson U. 1979. Maximum likelihood estimation of the polychoric correlation coefficient. Psychometrika. Volume 44

[75] Olkin I, Tate RF. 1961. Multivariate Correlation Models with Mixed Discrete and Continuous Variables. Ann. Math. Stat. Volume 32.

[76] Agresti A. 1981. Measures of Nominal-Ordinal Association. J. Am. Stat. Assoc.

[77] Yule GU, Udny Yule G. 1912. On the Methods of Measuring Association Between Two Attributes. J. R. Stat. Soc.

[78] Pearson K. 1894. Mathematical Contributions to the Theory of Evolution. II. Skew Variation in Homogeneous Material. Proc. R. Soc. Lond. Vol. 186.

[79] Cramér H. 1946. Mathematical Methods of Statistics. Mathematical Methods of Statistics.

[80] Wasserstein, Ronald L, Nicole A. Lazar. 2016. "The ASA's Statement Onp-Values: Context, Process, and Purpose." The American Statistician.

[81] Dunn, Olive Jean. 1959. "Estimation of the Medians for Dependent Variables." Annals of Mathematical Statistics.

[82] Anscombe, F. J. Rupert G. Miller. 1985. "Simultaneous Statistical Inference." Journal of the American Statistical Association.

[83] Sidak, Zbynek. 1967. "Rectangular Confidence Regions for the Means of Multivariate Normal Distributions." Journal of the American Statistical Association.

[84] Tukey, John W. 1949. "Comparing Individual Means in the Analysis of Variance." Biometrics.

[85] Hochberg, Yosef. 1988. "A Sharper Bonferroni Procedure for Multiple Tests of Significance." Biometrika.

[86] Upton, Graham, and Ian Cook. A Dictionary of Statistics 3e. Oxford (UK). Oxford University Press, 2014.

[87] Benjamini, Yoav. 2010. “Discovering the False Discovery Rate.” Journal of the Royal Statistical Society. Series B, Statistical Methodology.

[88] Mankiewicz, Richard. 2000. The Story of Mathematics. Weidenfeld & Nicolson

[89] Welch, B. L. 1947. “The Generalization of `Student’s' Problem When Several Different Population Variances Are Involved.” Biometrika.

[90] Mann, H. B., D. R. Whitney. 1947. “On a Test of Whether One of Two Random Variables Is Stochastically Larger than the Other.” Annals of Mathematical Statistics.

[91] Gibbons, Jean Dickinson, Subhabrata Chakraborti. Nonparametric Statistical Inference, Fifth Edition. London (UK). CRC Press, 2010.

[92] Yates, F. 1934. “Contingency Tables Involving Small Numbers and the χ 2 Test.” Supplement to the Journal of the Royal Statistical Society.

[93] Box, G. E. P. 1953. “NON-NORMALITY AND TESTS ON VARIANCES.” Biometrika.

[94] Agresti, A. 2001. “Exact Inference for Categorical Data: Recent Advances and Continuing Controversies.” Statistics in Medicine.

[95] Davis, J., M. Maes, A. Andreazza, J. J. McGrath, S. J. Tye, M. Berk. 2015. “Towards a Classification of Biomarkers of Neuropsychiatric Disease: From Encompass to Compass.” Molecular Psychiatry.

[96] Eckardt, Kai-Uwe, Seth L. Alper, Corinne Antignac, Anthony J. Bleyer, Dominique Chauveau, Karin Dahan, Constantinos Deltas, et al. 2015. Autosomal Dominant Tubulointerstitial Kidney Disease: Diagnosis, Classification, and management—A KDIGO Consensus Report. Kidney International

[97] Wisittipanit, Nuttachat, Huzefa Rangwala, Masoumeh Sikaroodi, Ali Keshavarzian, Ece A. Mutlu, Patrick Gillevet. 2015. Classification Methods for the Analysis of LH-PCR Data Associated with Inflammatory Bowel Disease Patients. International Journal of Bioinformatics Research and Applications

[98] Möller, Christiane, Yolande A. L. Pijnenburg, Wiesje M. van der Flier, Adriaan Versteeg, Betty Tijms, Jan C. de Munck, Anne Hafkemeijer, et al. 2016. Alzheimer Disease and Behavioral Variant Frontotemporal Dementia: Automatic Classification Based on Cortical Atrophy for Single-Subject Diagnosis. Radiology.

[99] Fisher, R. A. 1936. "The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics.

[100] Kilic, Selim. 2015. Binary Logistic Regression Analysis." Journal of Mood Disorders.

[101] Chhogyal, Kinzang, Abhaya Nayak. 2016. An Empirical Study of a Simple Naive Bayes Classifier Based on Ranking Functions. In Lecture Notes in Computer Science.

[102] Cortes, Corinna, Vladimir Vapnik. 1995. Support-Vector Networks. Machine Learning.

[103] Quinlan, J. R. 1987. Simplifying Decision Trees. International Journal of Man-Machine Studies.

[104] Bishop, Christopher M. Neural Networks for Pattern Recognition. New York (NY). Oxford University Press, 1995. Pattern Recognition and Machine Learning. New York (NY). Springer Verlag, 2006.

[105] Tipping, Michael E. 2001. Sparse Bayesian Learning and the Relevance Vector Machine. JMLR.

[106] LeCun, Yann, Yoshua Bengio, Geoffrey Hinton. 2015. Deep Learning. Nature.

[107] Aho, Ken, Dewayne Derryberry, and Teri Peterson. 2014. Model Selection for Ecologists: The Worldviews of AIC and BIC. Ecology.

[108] Schwarz, Gideon. 1978. Estimating the Dimension of a Model. Annals of Statistics.

[109] Dutta, Ritabrata, Malgortaza Bogdan, Jayanta K. Ghosh. 2015. Model Selection and Multiple Testing - A Bayesian and Empirical Bayes Overview and Some New Results.

[110] Toni, Tina, Michael P. H. Stumpf. 2010. Simulation-Based Model Selection for Dynamical Systems in Systems and Population Biology. Bioinformatics.

[111]Hug, Sabine, Daniel Schmidl, Wei Bo Li, Matthias B. Greiter, Fabian J. Theis. 2015. Bayesian Model Selection Methods and Their Application to Biological ODE Systems. Studies in Mechanobiology, Tissue Engineering and Biomaterials.

[112] Yang, X., Y. Guo, P. Skipp, A. Rowe. 2012. Automating Mass Spectrometry Proteomics Analysis.

[113] Fawcett, Tom. 2006. An Introduction to ROC Analysis. Pattern Recognition Letters.

[114] Tibshirani, Robert. 2011. Regression Shrinkage and Selection via the Lasso: A Retrospective. Journal of the Royal Statistical Society. Series B, Statistical Methodology.

[115] Tipping, Michael E. 2001. Sparse Bayesian Learning and the Relevance Vector Machine. JMLR.

[116] Abeel, Thomas, Thibault Helleputte, Yves Van de Peer, Pierre Dupont, Yvan Saeys. 2010. Robust Biomarker Identification for Cancer Diagnosis with Ensemble Feature Selection Methods. Bioinformatics.

[117] Zucknick, Manuela, Sylvia Richardson, Euan A. Stronach. 2008. Comparing the Characteristics of Gene Expression Profiles Derived by Univariate and Multivariate Classification Methods. Statistical Applications in Genetics and Molecular Biology.

[118] Ahmed, Ismaïl, Anna-Liisa Hartikainen, Marjo-Riitta Järvelin, Sylvia Richardson. 2011. False Discovery Rate Estimation for Stability Selection: Application to Genome-Wide Association Studies. Statistical Applications in Genetics and Molecular Biology.

[119] Alexander, David H, Kenneth Lange. 2011. Stability Selection for Genome-Wide Association. Genetic Epidemiology.

[120] Saria, Suchi, Anna Goldenberg. 2015. Subtyping: What It Is and Its Role in Precision Medicine. IEEE Intelligent Systems.

[121] Murphy, Kevin P. Machine Learning: A Probabilistic Perspective. Cambridge (MA). MIT Press, 2012.

[122] Bishop, Christopher M. Neural Networks for Pattern Recognition. New York (NY). Oxford University Press, 1995.
Pattern Recognition and Machine Learning. New York (NY). Springer Verlag, 2006.

[123] Madeira, Sara C, Arlindo L. Oliveira. 2004. Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM.

[124] Cheng Y, Church GM. 2000. Biclustering of expression data. Proc. Int. Conf. Intell. Syst. Mol. Biol, PMID: 10977070

[125] Gertz G et al. 2000. Coupled two-way clustering analysis of gene microarray data. Proc. Natl. Acad. Sci.

[126] Bergmann S et al. 2003. Iterative signature algorithm for the analysis of large-scale gene expression data. Phys Rev E Stat Nonlin Soft Matter Phys., PMID: 12689096

[127] Tanay A et al. 2004. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. Proc. Natl. Acad. Sci USA. PMID: 14973197

[128] Tanay A et al. 2005. Biclustering Algorithms. Chapman & Hall/CRC Computer & Information Science Series.

[129] Oghabian A et al. 2014. Biclustering methods: biological relevance and application in gene expression analysis. PLoS One. PMID: 24651574

[130] Cha K et al. 2015. Discovering transnosological molecular basis of human brain diseases using biclustering analysis of integrated gene expression data. BMC Med Inform Decis Mak. PMID: 26043779

[131] Williams A, Halappanavar S. 2017. Application of bi-clustering of gene expression data and gene set enrichment analysis methods to identify potentially disease causing nanomaterials. Data Brief. PMID: 29159232

[132] Russell, Jesse, Ronald Cohn. 2012. Topological Data Analysis. Book on Demand Limited.

[133] Nicolau, Monica, Arnold J. Levine, Gunnar Carlsson. 2011. Topology Based Data Analysis Identifies a Subgroup of Breast Cancers with a Unique Mutational Profile and Excellent Survival. Proceedings of the National Academy of Sciences of the United States of America.

[134] Hinks, Timothy S. C., Xiaoying Zhou, Karl J. Staples, Borislav D. Dimitrov, Alexander Manta, Tanya Petrossian, Pek Y. Lum, et al. 2015. Innate and Adaptive T Cells in Asthmatic Patients: Relationship to Severity and Disease Mechanisms. The Journal of Allergy and Clinical Immunology.

[135] Lum, P. Y., G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, G. Carlsson. 2013. Extracting Insights from the Shape of Complex Data Using Topology. Scientific Reports.

[136] Rucco, M., E. Merelli, D. Herman, D. Ramanan, T. Petrossian, L. Falsetti, C. Nitti, A. Salvi. 2015. Using Topological Data Analysis for Diagnosis Pulmonary Embolism. Journal of Theoretical and Applied Computer Science.

[137] Hinks TS et al. 2015. Innate and adaptive T cells in asthmatic patients: Relationship to severity and disease mechanisms. J Allergy Clin Immunol. PMID: 25746968

[138] Blog, Guest, Tavish Srivastava, Pranav Dar, Faizan Shaikh. 2017. How to Handle Imbalanced Classification Problems in Machine Learning?. Analytics Vidhya.

[139] Hoens, T. Ryan, T. Ryan Hoens, Nitesh V. Chawla. 2013. Imbalanced Datasets: From Sampling to Classifiers. In Imbalanced Learning.

[140] Jo T, Japkowicz N. 2004. Class imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter.

[141] Chawla NV et al. 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res.

[142] Breiman, Leo. 1996. Bagging Predictors. Machine Learning.

[143] Freund, Yoav, Robert E. Schapire. 1996. Experiments with a New Boosting Algorithm. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning.

[144] Cutler, Adele. 2014. Random Forests. In Wiley StatsRef: Statistics Reference Online.

[145] Pan, Sinno Jialin, Qiang Yang. 2010. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering.

[146] Huynh, Benjamin Q., Hui Li, Maryellen L. Giger. 2016. Digital Mammographic Tumor Classification Using Transfer Learning from Deep Convolutional Neural Networks. Journal of Medical Imaging.

[147] Krizhevsky, Alex, Ilya Sutskever, Geoffrey E. Hinton. 2017. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM.

[148] Huynh, Benjamin Q., Hui Li, Maryellen L. Giger. 2016. Digital Mammographic Tumor Classification Using Transfer Learning from Deep Convolutional Neural Networks Journal of Medical Imaging.

[149] Chen, Weijie, Maryellen L. Giger, Hui Li, Ulrich Bick, Gillian M. Newstead. 2007. Volumetric Texture Analysis of Breast Lesions on Contrast-Enhanced Magnetic Resonance Images. Magnetic Resonance in Medicine: Official Journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine.

[150] Huo, Z., M. L. Giger, C. J. Vyborny, D. E. Wolverton, R. A. Schmidt, K. Doi. 1998. Automated Computerized (Classification of Malignant and Benign Masses on Digitized Mammograms. Academic Radiology.

[151] Li, Hui, Maryellen L. Giger, Yading Yuan, Weijie Chen, Karla Horsch, Li Lan, Andrew R. Jamieson, Charlene A. Sennett, Sanaz A. Jansen. 2008. Evaluation of Computer-Aided Diagnosis on a Large Clinical Full-Field Digital Mammographic Dataset. Academic Radiology.

[152] Kittler, Josef, Mohamad Hatef, Robert P. W. Duin, Jiri Matas. 1998. On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[153] Huynh, Benjamin Q., Hui Li, Maryellen L. Giger. 2016. Digital Mammographic Tumor Classification Using Transfer Learning from Deep Convolutional Neural Networks. Journal of Medical Imaging.

[154] Gabrys, Bogdan, Lina Petrakieva. 2004. Combining Labelled and Unlabelled Data in the Design of Pattern Classification Systems. International Journal of Approximate Reasoning: Official Publication of the North American Fuzzy Information Processing Society.

[155] Dara R et al. 2002. Clustering unlabeled data with SOMs improves classification of labeled real-world data. ISBN: 0-7803-7278-6

[156] Mitchell TM. 2004. The Role of Unlabeled Data in Supervised Learning. Language, Knowledge, and Representation.

[157] Nigam K, Ghani R. 2000. Analyzing the effectiveness and applicability of co-training. Proceedings of the ninth international conference on Information and knowledge management - CIKM '00.

[158] Nigam K et al. 1998. Using EM to Classify Text from Labeled and Unlabeled Documents. Machine Learning, Vol. 39.

[159] Ghahramani Z, Jordan MI. 1994. Supervised learning from incomplete data via an EM approach. Advances in Neural Information Processing Systems.

[160] Kothari R, Jain V. 2002. Learning from labeled and unlabeled data. Proceedings of the 2002 International Joint Conference on Neural Networks.

[161] Gabrys B, Bargiela A. 2000. General fuzzy min-max neural network for clustering and classification. IEEE Trans. Neural Net, Vol 11.

[162] Pedrycz W et al. 2008. Fuzzy Clustering With Partial Supervision in Organization and Classification of Digital Images. IEEE Trans. Fuzzy Syst, Vol. 16.

[163] Gabrys B. 2002. Neuro-fuzzy approach to processing inputs with missing values in pattern recognition problems. Int. J. Approx. Reason, Vol. 30.

[164] L'Heureux, Alexandra, Katarina Grolinger, Hany F. Elyamany, Miriam A. M. Capretz. 2017. Machine Learning With Big Data: Challenges and Approaches. IEEE Access.

[165] Biomarkers Definitions Working Group. 2001. Clin. Pharmacol. Ther. doi: 10.1067/mcp.2001.113989

[166] Ransohoff DF. 2008. The Process to Discover and Develop Biomarkers for Cancer: A Work in Progress, JNCI, Volume 100.

[167] Pepe MS et al. 2001. Phases of biomarker development for early detection of cancer. J. Natl. Cancer Inst.

[168] Ludwig JA, Weinstein JN. 2005. Biomarkers in cancer staging, prognosis and treatment selection. Nat Rev Cancer.

[169]Zarringhalam K et al. 2018. Robust phenotype prediction from gene expression data using differential shrinkage of co-regulated genes. Sci Rep. doi: 10.1038/s41598-018-19635-0

[170] Fan C, Oh DS, Wessels L, Weigelt B, Nuyten DS, Nobel AB, van't Veer LJ, Perou CM. 2006. Concordance among gene-expression-based predictors for breast cancer. N. Engl. J. Med.

[171] Langfelder, Peter, Steve Horvath. 2008. WGCNA: An R Package for Weighted Correlation Network Analysis. BMC Bioinformatics.

[172] Ibidem

[173] Hänzelmann, Sonja, Robert Castelo, Justin Guinney. 2013. GSVA: Gene Set Variation Analysis for Microarray and RNA-Seq Data. BMC Bioinformatics.

[174] Ibidem

[175] Huang, Sijia, Kumardeep Chaudhary, Lana X. Garmire. 2017. More Is Better: Recent Progress in Multi-Omics Data Integration Methods. Frontiers in Genetics.

[176] Wang, Bo, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. 2014. Similarity Network Fusion for Aggregating Data Types on a Genomic Scale. Nature Methods.

[177]Radermacher FJ. 1990. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (Judea Pearl). SIAM Rev.

[178] Huang S et al. 2017. More is better: recent progress in multi-omics data integration methods. Frontiers in Genetics. PMID: 28670325

[179] Wang, Bo, et al. 2014. Similarity network fusion for aggregating data types on a genomic scale. Nature methods, PMID: 24464287

[180] Mamoshina, Polina, Armando Vieira, Evgeny Putin, Alex Zhavoronkov. 2016. Applications of Deep Learning in Biomedicine. Molecular Pharmaceutics.

[181]Tan, Jie, Matthew Ung, Chao Cheng, Casey S. Greene. 2015. Unsupervised Feature Construction and Knowledge Extraction from Genome-Wide Assays of Breast Cancer with Denoising Autoencoders. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing.

[182]Lasko, Thomas A., Joshua C. Denny, Mia A. Levy. 2013. Correction: Computational Phenotype Discovery Using Unsupervised Feature Learning over Noisy, Sparse, and Irregular Clinical Data. PloS One.

[183] Liang, Muxuan, Zhizhong Li, Ting Chen, Jianyang Zeng. 2015. Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach. IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM.

[184]Miotto, Riccardo, Li Li, Brian A. Kidd, Joel T. Dudley. 2016. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. Scientific Reports.

[185] Beaulieu-Jones, Brett K., Casey S. Greene, Pooled Resource Open-Access ALS Clinical Trials Consortium. 2016. Semi-Supervised Learning of the Electronic Health Record for Phenotype Stratification. Journal of Biomedical Informatics.

[186] Liang M, et al. Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach. PubMed - NCBI.

[187] Ibidem

[188] Mazein A et al. 2018. Systems medicine disease maps: community-driven comprehensive representation of disease mechanisms. NPJ. Syst. Biol. Appl. PMID: 29872544

[189]Mizuno S et al. 2012. AlzPathway: a comprehensive map of signaling pathways of Alzheimer's disease. BMC Syst Biol. PMID: 22647208

[190]Matsuoka Y et al. 2013. A comprehensive map of the influenza A virus replication cycle. BMC Syst Biol. PMID 24088197

[191] Fujita KA et al. 2014. Integrating pathways of Parkinson's disease in a molecular interaction map. Mol. Neurobiol. PMID 23832570

[192] Kuperstein I et al. 2015. Atlas of Cancer Signalling Network: a systems biology resource for integrative analysis of cancer data with Google Maps. Oncogenesis. PMID 26192618

[193] https://diseaseknowledgebase.etriks.org/disease-map/ Access: December 2018

[194] Systems Medicine Disease Maps [Internet]. Disease Maps Project. [Date Unknown; Accessed December 2018]. Available from: http://disease-maps.org

[195] Ostaszewski M et al. 2018. Community-driven roadmap for integrated disease maps. Brief Bioinform. PMID 29688273

[196] Mazein A et al. 2018. Systems medicine disease maps: community-driven comprehensive representation of disease mechanisms. NPJ. Syst. Biol. Appl. PMID: 29872544

[197] 3rd Disease Maps Community Meeting [Internet]. Disease Maps Project. [Date Unknown: Accessed December 2018]. Available from: http://disease-maps.org/DMCM2018

[198] Demir E et al. 2010. The BioPAX community standard for pathway data sharing. Nat Biotechnol. PMID 20829833

[199] Balaur, Irina, Alexander Mazein, Mansoor Saqi, Artem Lysenko, Christopher J. Rawlings, Charles Auffray. 2017. Recon2Neo4j: Applying Graph Database Technologies for Managing Comprehensive Genome-Scale Networks. Bioinformatics.

[200] Lysenko, Artem, Irina A. Roznovăţ, Mansoor Saqi, Alexander Mazein, Christopher J. Rawlings, Charles Auffray. 2016. Representing and Querying Disease Networks Using Graph Databases. BioData Mining.

[201] Le Novère N et al. 2009. The Systems Biology Graphical Notation. Nat Biotechnol. PMID 19668183

[202] Hucka M et al. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics. PMID 12611808

[203] Glaab E, Schneider R. 2015. Comparative pathway and network analysis of brain transcriptome changes during adult aging and in Parkinson's disease. Neurobiol Dis. PMID: 25447234

Chapter 6: tranSMART: A Data Warehouse for Biomedical Data Analysis

Florian Guitton and Yike Guo

6.1 tranSMART background

Much enthusiasm and energy are being directed toward open-source software approaches, pre-competitive data sharing, and external innovation in the biopharmaceutical industry. TranSMART was developed by the Centocore division of Johnson and Johnson (J&J) and was released open source in 2012 to promote the system's use as a standard translational information solution across academia and industry. J&J's intent was to enable shared support for the platform's continued technical development. Additionally, costly data curation and mapping efforts could be minimized across community research partners if all partners used the same information system, greatly enabling data exchange.

The architecture and features of tranSMART will be detailed in this chapter. J&J extended I2B2 (Informatics Integrating Benchtop to Bedside), an open license clinical data management platform developed by investigators at the Harvard Medical School, to support the management and analysis of microarray gene expression platforms thereby creating tranSMART. At the time of this writing, the latest tranSMART version release is v17.1 available from the I2B2/tranSMART Foundation. This version was funded by certain eTRIKS partners and is also released as eTRIKS version 5.0 with the eTRIKS version having a newly developed graphical user interface called "Borderline". The tranSMART version detailed in this chapter is v16.2 which, at the time of this writing, is the most recent stable release.

TranSMART is comprised of a web 2.0 application for data analysis and visualization which uses the open license statistical application R for mathematical computing, an extract transform load (ETL) application for ingesting data into the platform and a relational database for persisting data loaded to the platform. Although the initial version of tranSMART was dependent on the commercial Oracle relational database management system (RDBMS) users now have a choice between using Oracle or the open source PostgreSQL RDBMS.

Developers representing a variety of organizations have made tranSMART interoperable with value added analytical solutions including the Galaxy data analysis platform, Dalliance genome browser, Cytoscape network analysis platform, GeneData Profiler, Ingenuity Pathway Analyzer, XNAT image management platform and many others.

6.2 Background

tranSMART consolidates clinical and corresponding high dimensional molecular omics data (gene expression profiles, genotypes, serum protein panels, metabolomics, proteomics data, etc.) across one or more studies within a single data warehouse. The tranSMART ETL process can load a wide variety of clinical datasets, regardless of the source of these data, provided that a curator can map the individual data elements into the tranSMART database. In practice, a great many studies have been mapped and loaded into tranSMART instances hosted by a sizeable community of organizations. Moreover, a set of successful service providers have emerged to support data curation and hosting for tranSMART users. Analysis ready data sets in tabular formats (i.e. primary data sets as described in chapter 4) that are pre-processed against community or organizational data standards having values that have been prepared for use by analysis methods are very well suited for tranSMART. Specialty data modalities, such as medical images, have been supported by custom integrations with fit for purpose applications. For example, an organization has enabled co-retrieval of medical images with clinical attributes by maintaining references between subject and visit data stored in tranSMART with corresponding medical images stored in XNAT.

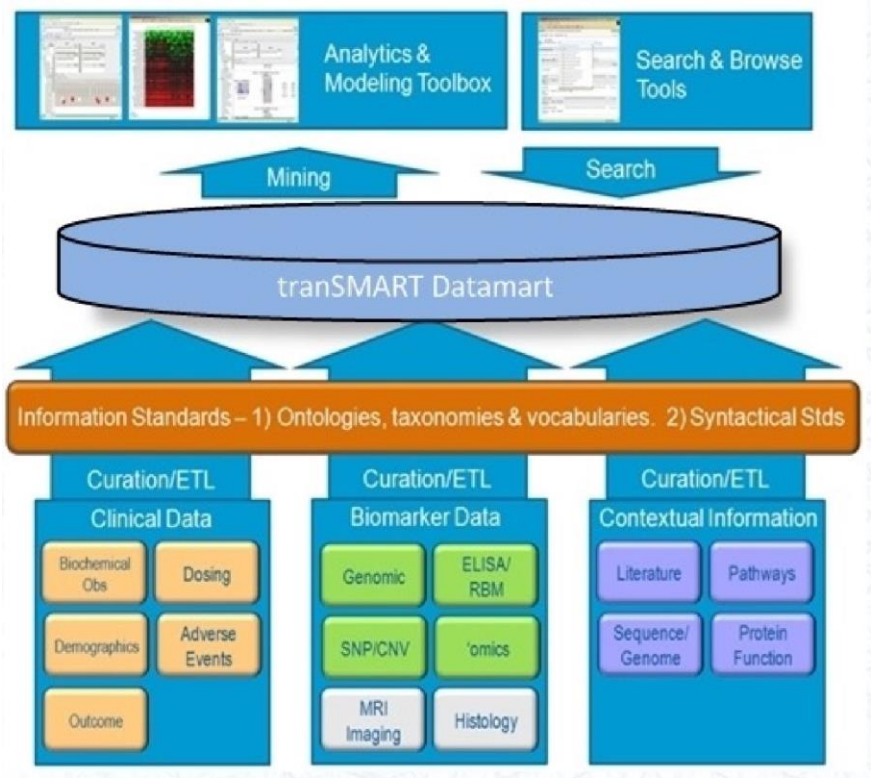


Figure 6.1: Diagram of the tranSMART

tranSMART uses a three-tiered architecture, a presentation tier, business tier, and data tier. As such, presentation, business processing and data management are physically separated as shown in Figure 6.2 to ease maintainability with respect to upgrading, replacing or substituting with respect to any of the tiers.

6.2.1 Presentation tier

The presentation tier is built from a suite of technologies including such as Java/Groovy server pages (GSP, JSP), Ajax JavaScript Framework, JSON and XML. The presentation layer is mainly written in GSP for rendering views using the Grails framework allowing dynamic and static content to be used in combination. Ajax JavaScript framework builds dynamic web pages on the client browser (preferably Firefox or Chrome). JSON and XML are used to pass data between the server and browser applications.

6.2.2 Business tier

The business tier is responsible for data processing including brokering interactions between the data and presentation tiers. The business tier implements the Spring Security framework. Most of the business tier software is written in Grails including the Model and Controller components of the model-view-controller (MVC) pattern that is central to tranSMART's design. tranSMART supports both SOAP and RESTful web services. SOAP (Simple Object Access Protocol) is a protocol for exchanging structured information across networks that relies on XML (Extensible Markup Language) for its message format. tranSMART also provides a RESTful (Representational State Transfer) web service application programming interface (API) to ease interoperability. The business tier implements a "plug in" architecture that includes the R module for mathematical computing. The business tier implements the operational services such as search, analysis, export and several others that will be detailed later.

6.2.3 Data tier

This data tier is responsible for exposing information held in tranSMART's database and file system. As noted above, there is a choice between the Oracle and PostgreSQL databases. GORM (Grails Object Relational Model) and hibernate are used for object relational mapping to encapsulate the database from the business tier query logic.

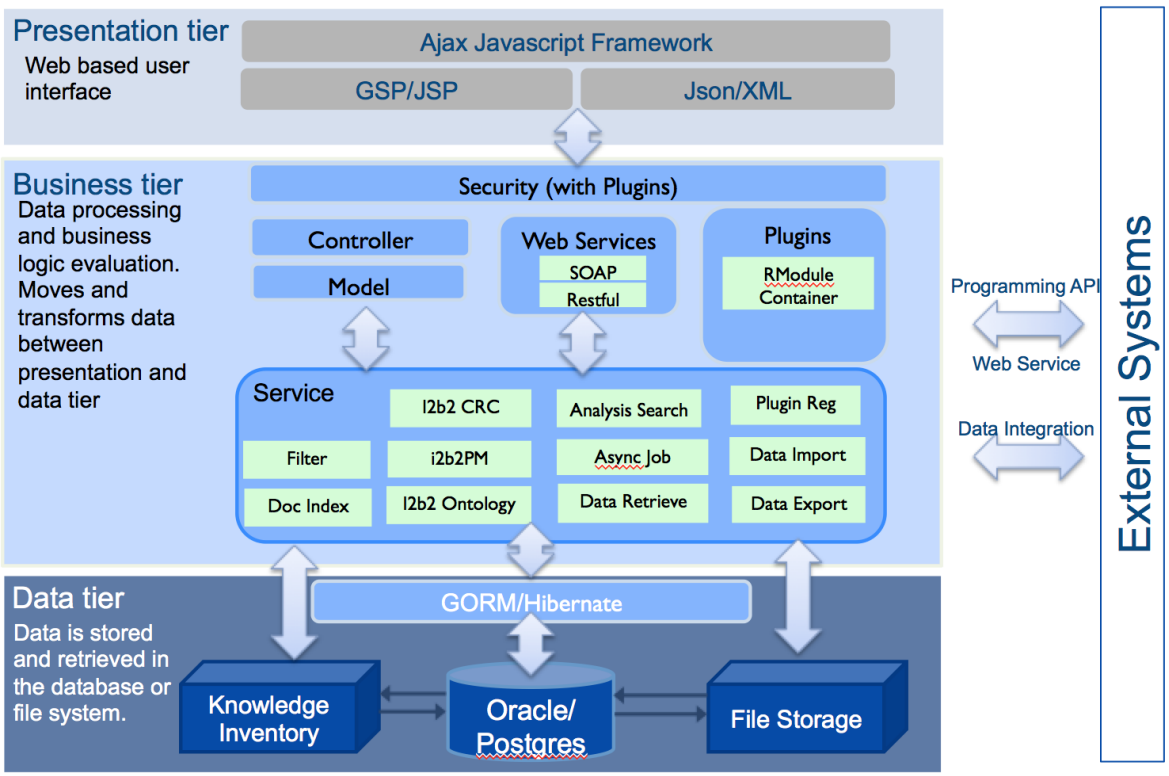


Figure 6.2: tranSMART three-tiered architecture

6.3 tranSMART functionality

Although this section does not review all tranSMART capabilities it will provide the reader with enough information to understand how tranSMART can support translation research study teams.

6.3.1 Search Panel

TranSMART allows users to search there is a search the system generally to discover research data and literature that match search terms that the user provides. Files that are returned from the search can be added to a cart for export. See Figure 6.3 Selected results from the search box can be used as a filter in the Active Filters box on the left panel.

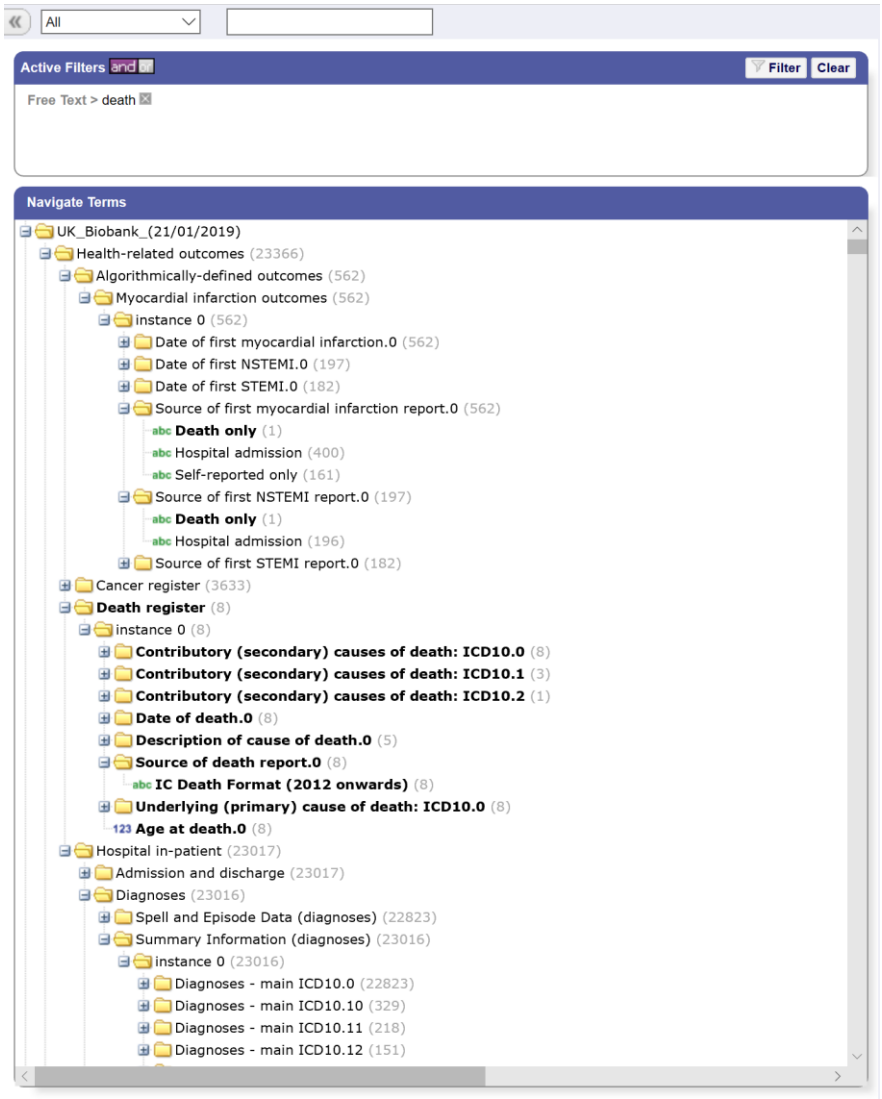


Figure 6.3: Search Panel

6.3.2 Analyze

tranSMART provides a biological concept search to build one or more cohorts of subjects using tools within the Analyze window. The analyze tool consists of a study hierarchy which is configurable by ETL curators for individual studies and is created as part of the ETL mapping process. The study hierarchy, or Program Explorer, is to the left of the Browse Window. Attributes within the study hierarchy can be identified through a free text search box. Attributes identified in this manner are opened and displayed in the hierarchy tree utility.

Alternatively, users can use the hierarchy tree utility to manually navigate study hierarchies to find attributes of interest. Attributes (associated with data values) are leaves of the study hierarchy.

Users can select clinical attributes of interest by dragging and dropping these into one or more query boxes of the search interface (right side of the Browse window). Attributes dragged and

dropped within boxes are subject to boolean "or" logic while attributes across boxes are subject to boolean "and" logic. In this manner, one or two cohorts (list(s) of subjects) can be created based on the attributes selected and how these are arranged in the query boxes. See Figure 6.4.

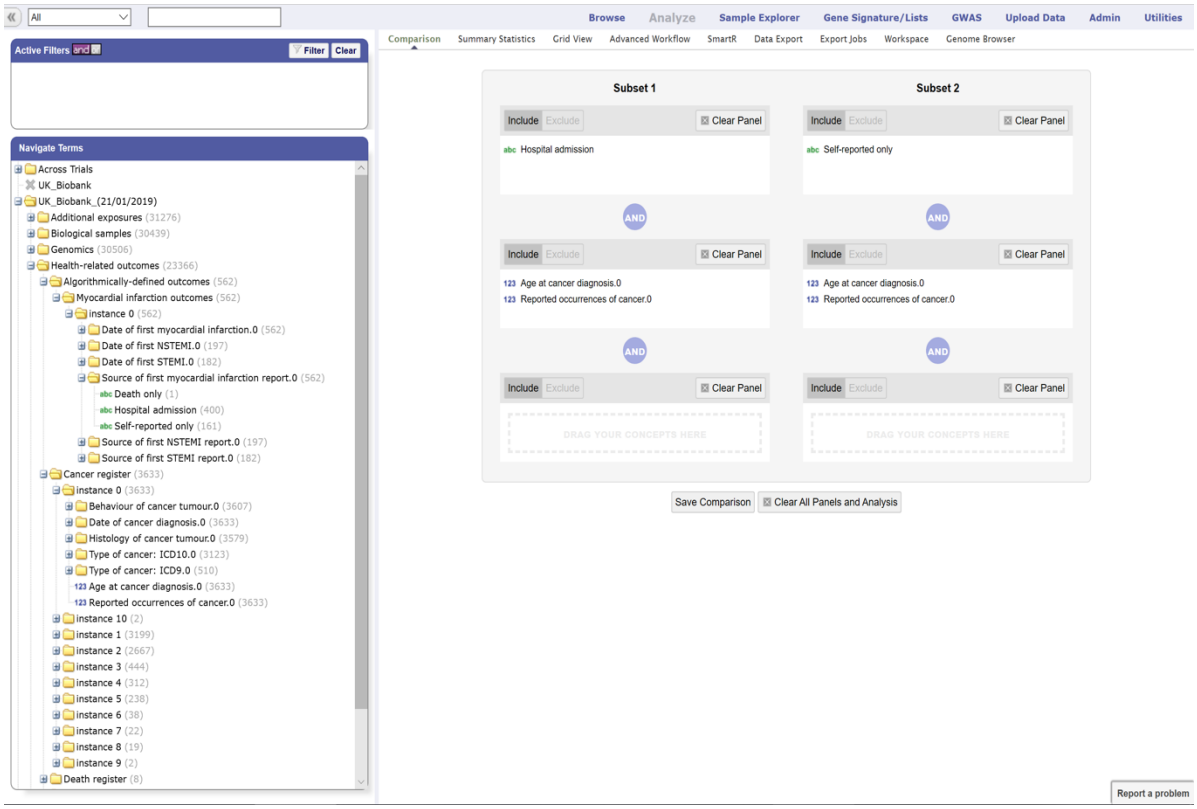


Figure 6.4: Cohort selection panel where multiple subset can be selected for comparison

If the "Summary Statistics" tab is selected tranSMART generates age, sex and race summary statistics with respect to the cohorts and displays these on a new view (see Figure 6.5). Attributes that are dragged and dropped from the navigation tree onto this interface will also be summarized. If two cohort are represented tranSMART will perform a student's t-test for inferring meaningful differences in attributes across the cohorts. If there is only a single cohort selected the data will be displayed in a summary manner.

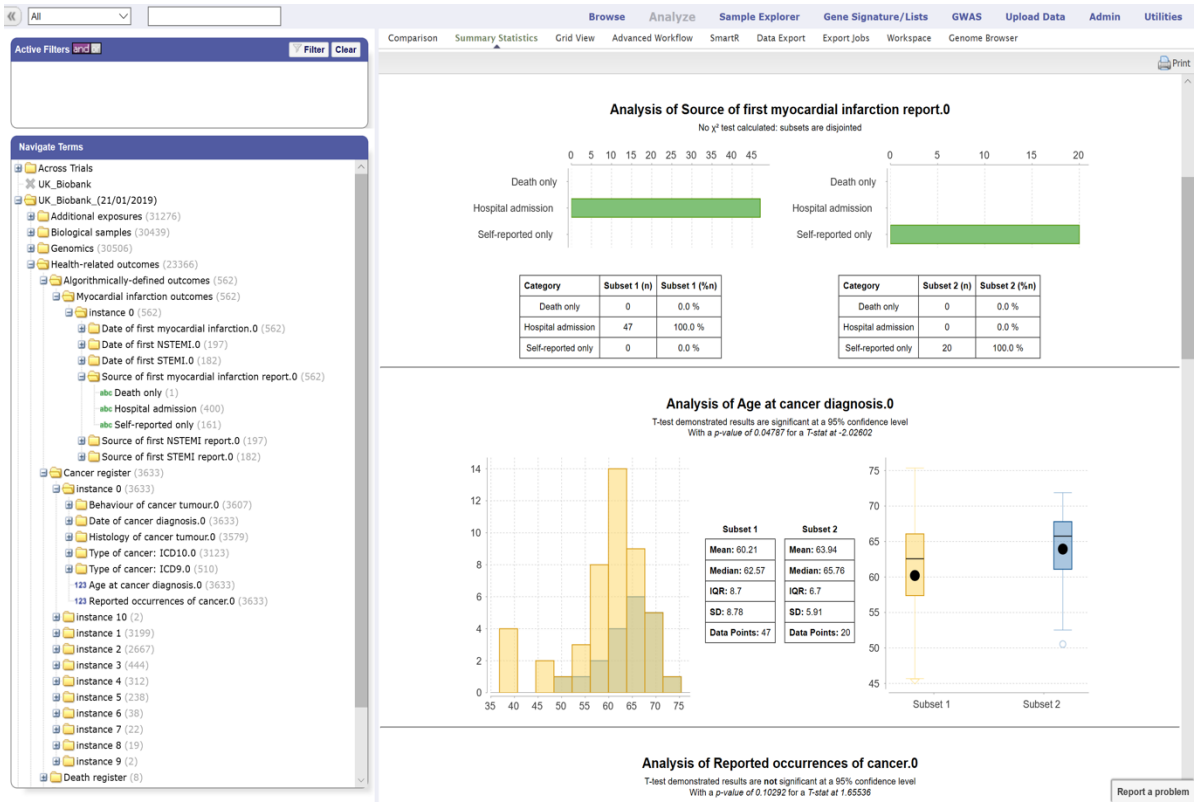


Figure 6.5: Summary statistics displaying information about the currently selected cohort

Data can be secured at the study level by administrators. Studies hierarchies will not open for user who are not authorized and, therefore, their data will not be selectable. The roles associated with studies can be set as follows.

- **View:** The view role allows users to define the criteria for the study groups to be compared, generate summary statistics for the study groups and specify points of comparison for the study groups.
- **Export:** The export role assigns users the view role and allows users to export data.
- **Own**: Users with OWN access level are assigned the export roles and are noted subject matter expert for the study.

Cohort queries can be saved (the query is saved to re-build the result set, the result set is not saved with the query) for recall or to share with other users using the workspace tab (see figure 6.6).

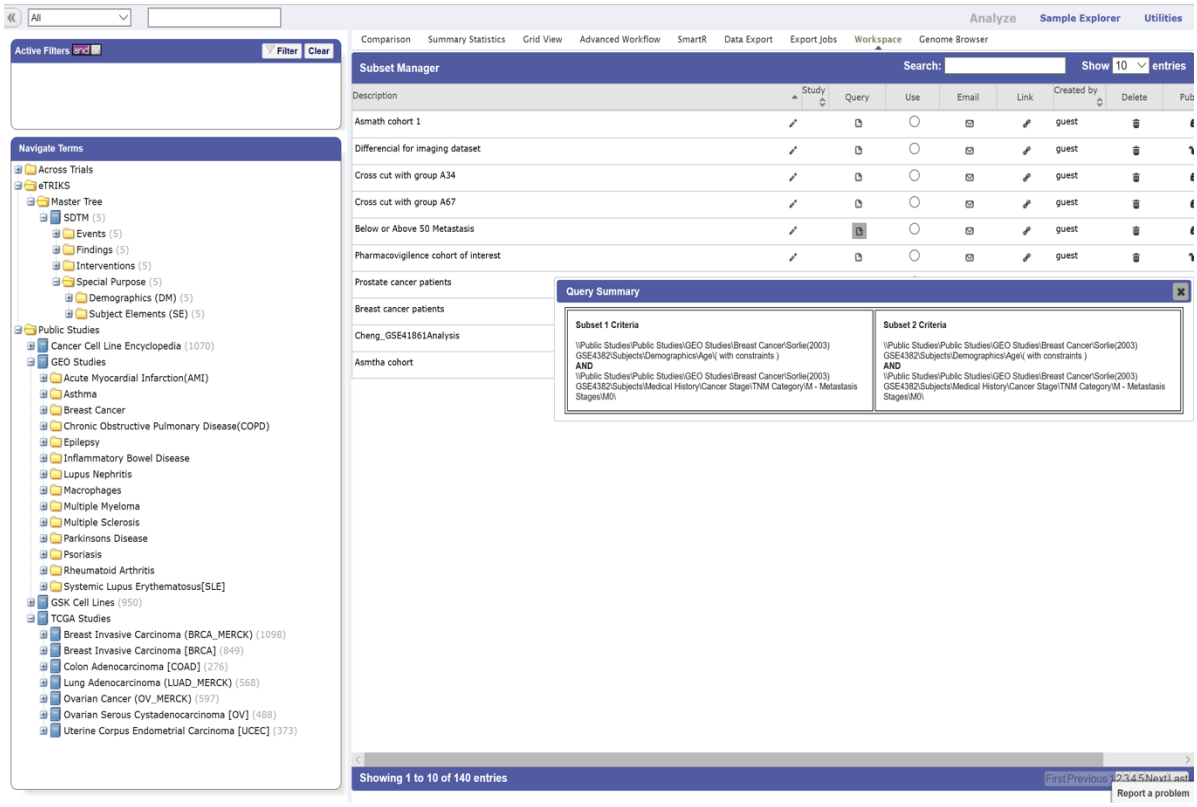


Figure 6.6: Workspace environment where cohorts can be saved

Once cohorts are selected, corresponding high dimensional datasets can be subset with the cohorts and applied in analysis methods. The Advanced Workflow tab offers several analysis tools that can be applied to selected cohorts including correlation analysis, forest plot, survival analysis, heatmap generation, principal component analysis (PCA), scatter plot with linear regression, box plot with analysis of variance, hierarchical clustering, IC50, K-Means Clustering, Line graph, Logistic regression, Fisher test, Waterfall plot (see Figure 6.7).

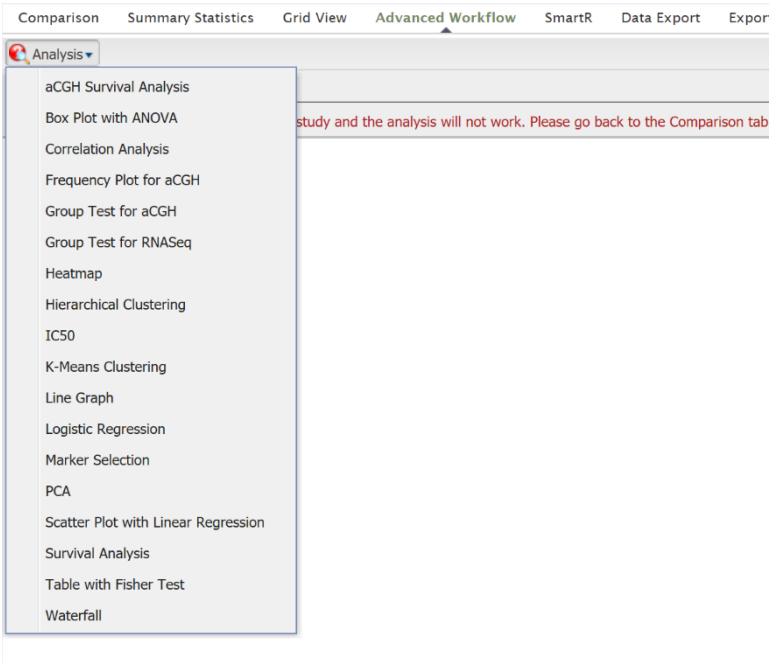


Figure 6.7: List of available workflows

Each advanced workflow method has its own pertinent input form and output display (see Figure 6.8, Figure 6.9, Figure 6.10, Figure 6.11). The workflows generally extract data and launch an R script to generate the analysis and output files. While the analysis is running the window displays a progress bar. On completion the results display appears. Any images or output files should be available to view or download.

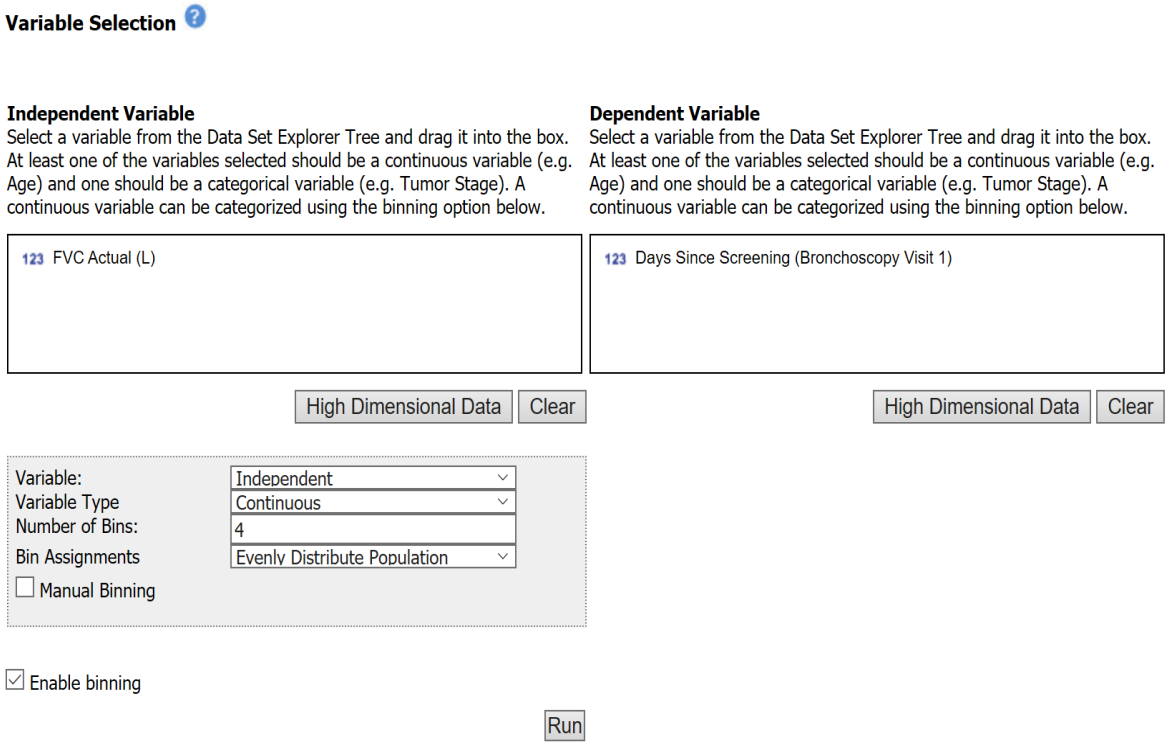


Figure 6.8: Input parameter screen for ANOVA analysis

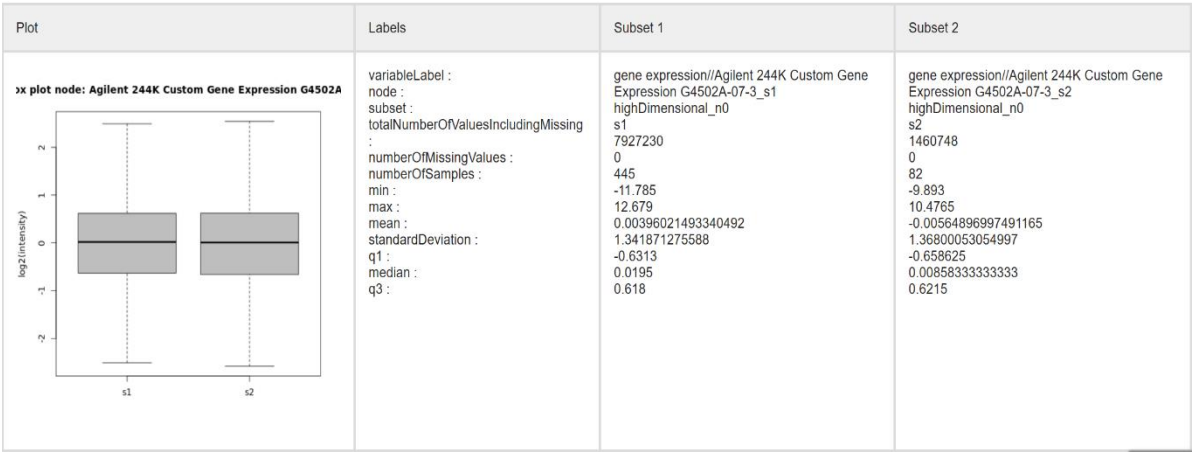
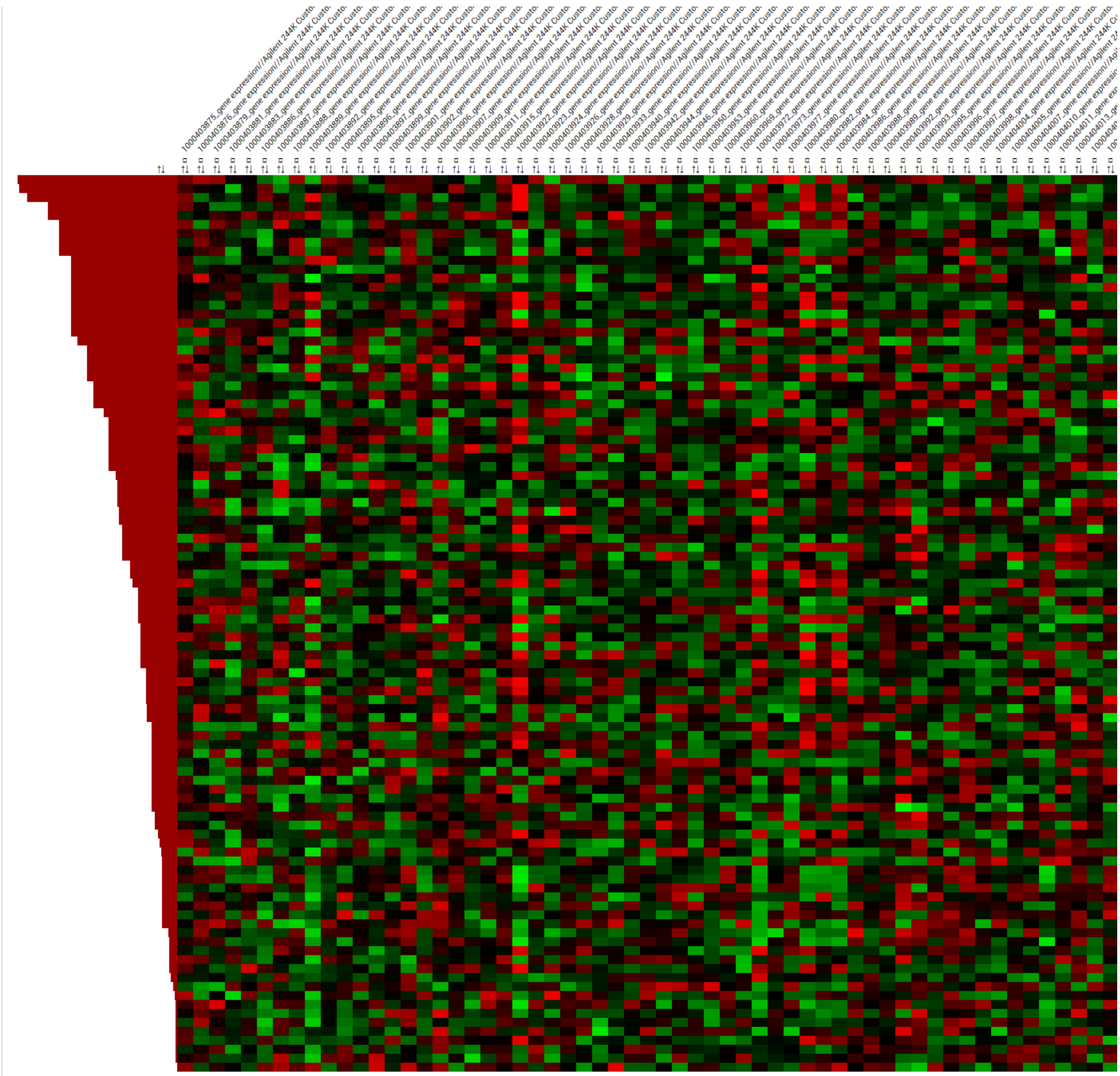
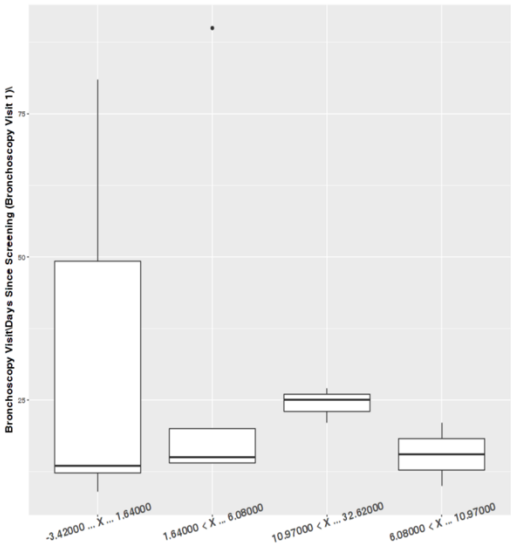


Figure 6.9: Intermediary step of an advance workflow using SmartR



ROWNAME	COEF	VARIANCE	RANGE	MEAN	MEDIAN	TTEST	LOGFOLD	PVAL	ADJPVAL	BVAL
PPP4R2--PPP4R2	4.3402653636828	0.220752113628505	2.37783333333	0.108252055660961	0.127333333333	-5.46018111326686	-0.300782858317396	7.3272881800707e-8	0.00130528311639779	7.50881136733155
RHOXF1--RHOXF1	1.25791656822765	1.38834232084535	4.8618	0.936691208096157	0.8115	5.26840684632904	0.725853621083331	2.00656946410371e-7	0.00178725142167717	6.60984865317153
RBED1--RBED1	0.908837915085335	0.471695921850311	2.471	0.75569165085389	0.68075	5.04252817544097	0.406479398465333	6.32246473986468e-7	0.00375427956253165	5.58789545115567
HSPB6--HSPB6	4.79259192581879	0.42111916998254	1.75675	0.135404174573055	-0.0325	4.7862628504023	0.365493169361469	0.00000220811215614629	0.00921112717833455	4.47744555838657
KIAA1683--KIAA1683	-1.78943176991319	0.850988310855216	4.524625	-0.515521347248577	-0.613375	4.75311765879044	0.515373944916415	0.00000258536184414914	0.00921112717833455	4.33766321576735
ACRC--ACRC	-0.777951154649507	0.48629314828717	3.674	-0.896389468690702	-0.88875	4.57243945669041	0.375751959440943	0.00000600946448923663	0.0122153384908141	3.59131124734962
DEPDC1--DEPDC1	-0.608343730236003	1.66195933808296	6.46	-2.11914737507907	-2.05125	-4.56664415275674	-0.692590725769609	0.00000617144080034393	0.0122153384908141	3.56781048132557
IPO9--IPO9	-1.54033735019961	0.216022030695408	2.3039	-0.301740201349338	-0.2536	-4.58918533955031	-0.252091746444942	0.00000556402982756833	0.0122153384908141	3.6593718394253
SMARCC1--SMARCC1	-3.26139678172147	0.408537619624524	3.10475555556	-0.195980286738361	-0.1588	-4.59844698235617	-0.34644966261684	0.00000533146914171574	0.0122153384908141	3.69711212184519
BCHE--BCHE	-0.360231004991445	2.11667799634458	4.85133333333	-4.03874383301719	-4.38183333333	4.42132253212021	0.757527154471441	0.0000119116983814121	0.0153178150695653	2.98748073319293
C10orf62--C10orf62	-4.5636437170894	0.322252436761692	1.88275	-0.124390259329553	-0.1755	4.42130635432015	0.296505517493332	0.0000119125584760266	0.0153178150695653	2.98741709150942
GPR89A--GPR89A	2.50839935629012	0.426853752761524	3.173	0.26046110056926	0.2315	-4.39910520789046	-0.339290517950123	0.0000131506163576494	0.0153178150695653	2.90028286153199
KCNV2--KCNV2	5.99368631761545	0.427939025591973	1.6955	0.10914326375711	0.026	4.39353455084691	0.339304124417655	0.0000134800675700199	0.0153178150695653	2.87848294262959
LOC55908--LOC55908	-0.373681823833199	1.2114972194609	2.94025	-2.94550142314991	-3.284	4.36231205873607	0.565959824609482	0.0000154777518385638	0.0153178150695653	2.75677113545947
MYH1--MYH1	1.48420228414381	1.47116458055678	3.999	0.817217267552182	0.52	4.37609723381559	0.625463126884077	0.0000145631286663102	0.0153178150695653	2.81040968827857
NEK2--NEK2	-1.16536592149064	2.22370586384546	6.13833333333	-1.27960626185954	-0.937333333333	-4.44212953410433	-0.779944400292439	0.0000108533319464238	0.0153178150695653	3.06951104295926
SGK2--SGK2	-1.63644161084877	1.68122096311666	4.876375	-0.792340552995351	-1.07457142857	4.37300685013421	0.668102640547427	0.0000147635618328642	0.0153178150695653	2.79837129804634
VPRBP--VPRBP	5.25951480470618	0.268330891880194	2.54316666667	0.0984894370651423	0.141	-4.42638579553706	-0.271064890837705	0.0000116454098984689	0.0153178150695653	3.0074096141 8929
MCEE--MCEE	-1.92427777360761	0.272999402903388	2.4365	-0.271527039848197	-0.292	4.32690877491785	0.267460331597698	0.0000180851320625982	0.0169562390822697	2.61973349424214
DAG1--DAG1	2.45520979065208	0.269816803945738	2.65880357143	0.211566007047977	0.26525	-4.27424287171885	-0.262782553439299	0.0000227531167987349	0.0193011439358411	2.41779126794437
GLYAT--GLYAT	1.72874318378592	2.07959347627404	3.85233333333	0.834178020240332	0.2885	4.28068573659366	0.727797769252058	0.0000221257114702665	0.0193011439358411	2.44237267448482

Figure 6.10: Final analysis with Heatmap workflow



Box Plot

ANOVA Result

p-value	0.895
F value	0.199

Group	Mean	n
-3.42000 ≤ X ≤ 1.64000	31.7	6
1.64000 < X ≤ 6.08000	30.6	5
10.97000 < X ≤ 32.62000	24.3	3
6.08000 < X ≤ 10.97000	15.5	2

Pairwise t-Test p-Values

	-3.42000 ≤ X ≤ 1.64000	**1.64000 < X ≤ 6.08000**	**10.97000 < X ≤ 32.62000**
1.64000 < X ≤ 6.08000	0.951	NA	NA
10.97000 < X ≤ 32.62000	0.717	0.764	NA
6.08000 < X ≤ 10.97000	0.492	0.531	0.735

Figure 6.11: Result of an ANOVA computation

The Grid View tab is used to display cohort data in a tabular format (see Figure 6.12). Users can add attributes to the grid via dragging and dropping concepts from the navigation tree and sort the grid by any column in ascending or descending order. Users can remove columns from view. Clinical data presented in the grid view can be exported as a delimited file (.csv, .xls).

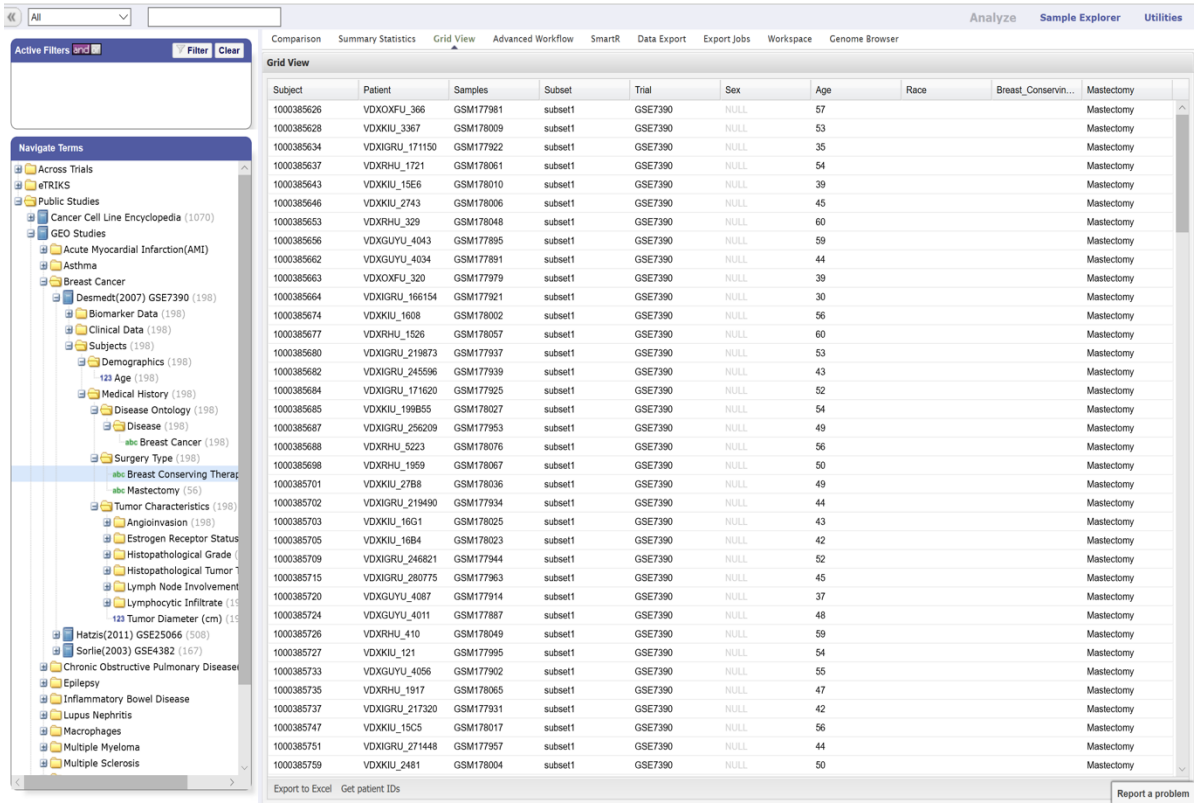


Figure 6.12: Grid view exposing the data values

The Data export tab is used to export the data associated with the cohorts (see Figure 6.13) with options to export either clinical data and/or corresponding high dimensional data, data for all study parameters or a user-defined subset of parameters, to run the export on line or as a background function. These export options place the files into zip archives. There is an export jobs tab that shows the status of each launched export job. The zip archive is accessible from the export jobs tab if the job is run in the background.

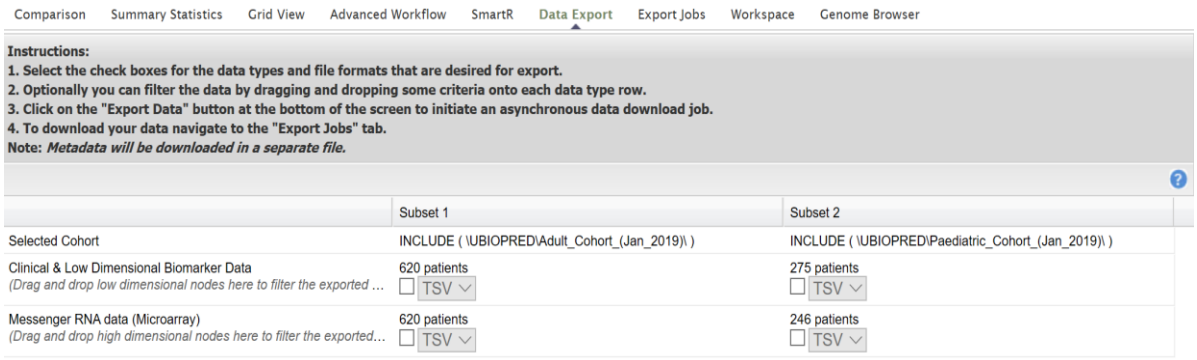


Figure 6.13: Contextual data export panel

The upload data tab allows an end user to upload a set of files to tranSMART (see Figure 6.14). It should be noted that tranSMART administrators often disable this feature and only allow dedicated curators to load data into the system.

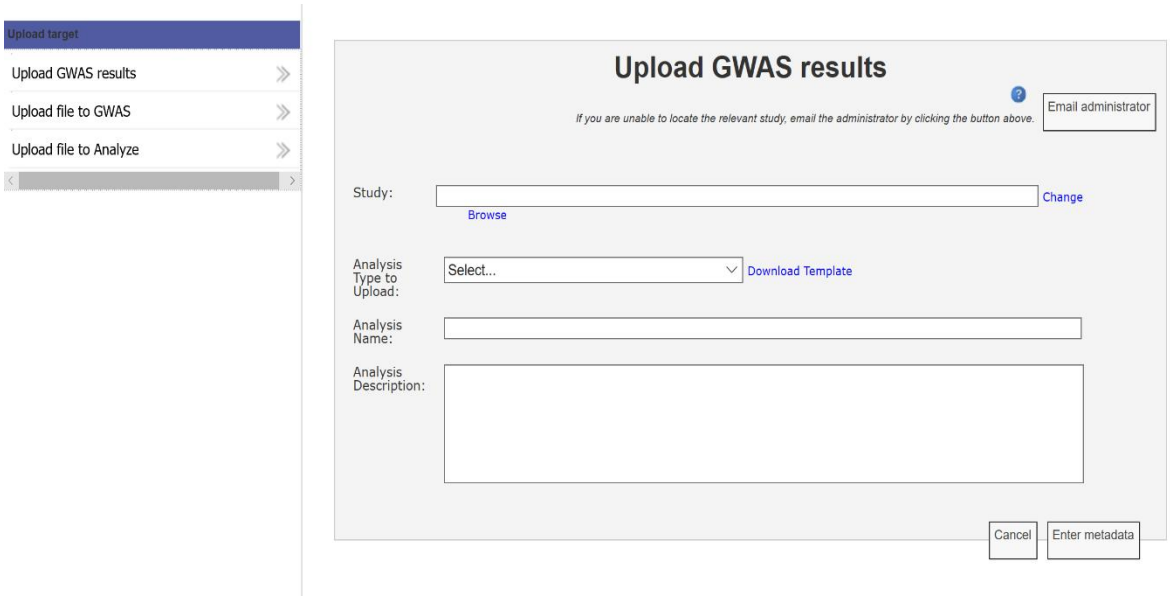


Figure 6.14: Upload of GWAS result data is available in the Upload Tab

Using the Gene Signature/List window, the user can view definitions of existing gene signatures and add new gene signature definitions as shown in Figure 6.15. The gene signatures can be used to find studies that maintain differential expression values for those genes in the signature. Gene signatures can be made private to the creator or shared with other users. Gene signatures can be cloned to facilitate the creation of new signatures and these can be edited to add or delete individual genes. Gene signatures can be exported to a .xls file.

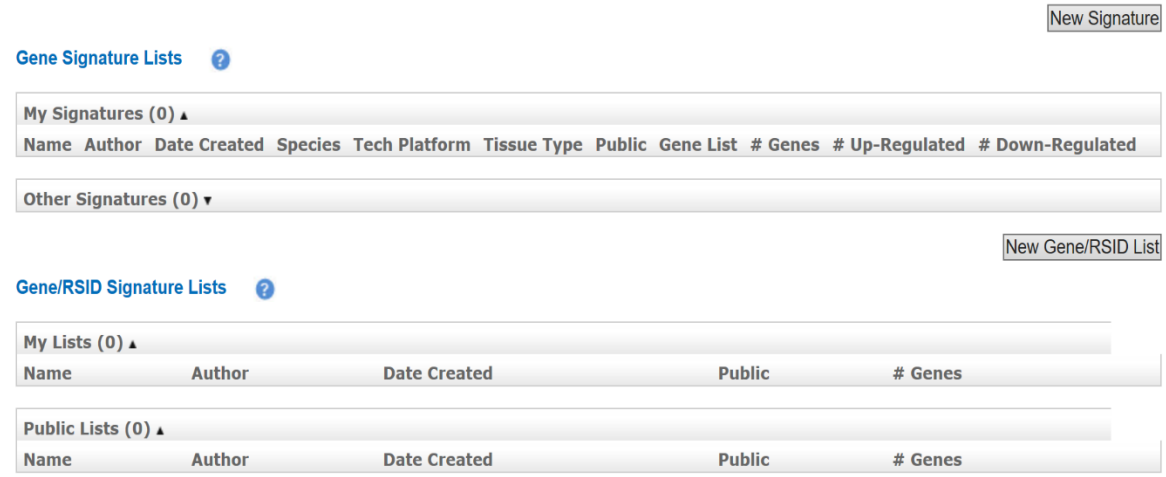


Figure 6.15: List gene panel

The admin panel (see Figure 6.16) allows administrators to view the system access log and to manage groups, users and roles. Administrators can restrict access to individual studies through this interface.

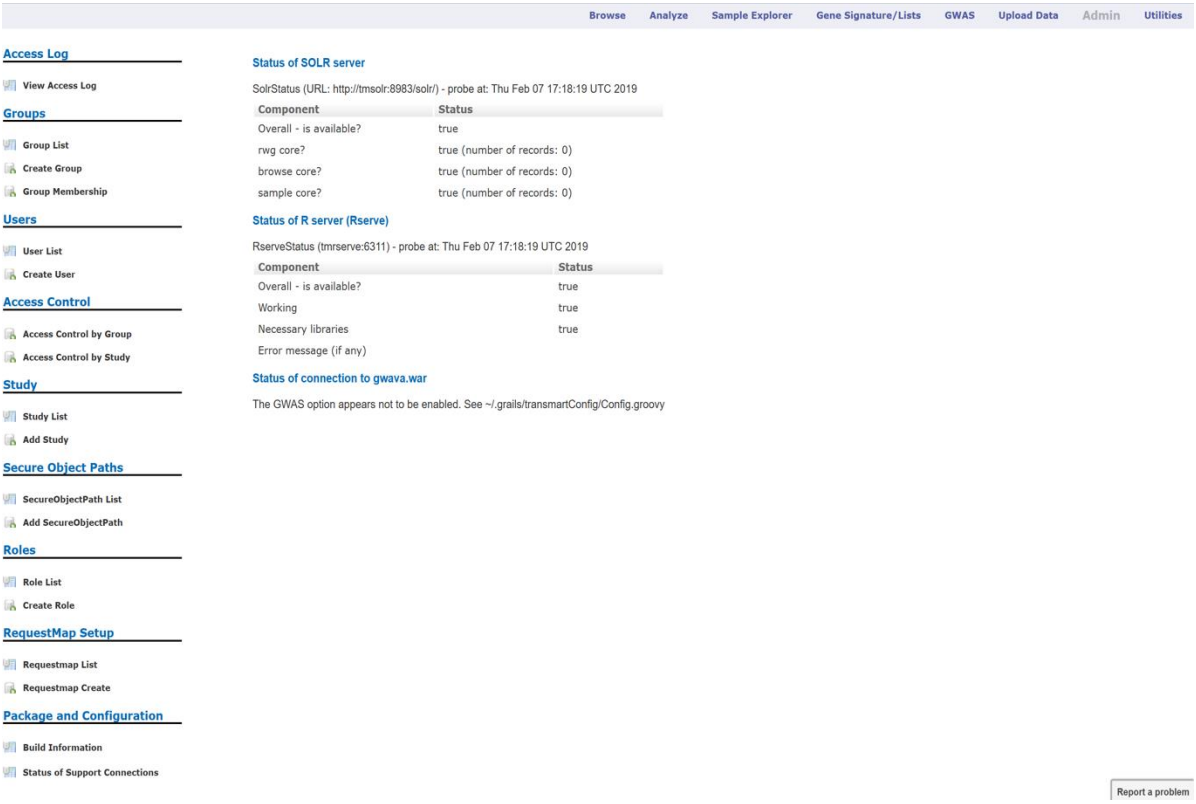


Figure 6.16: Administration Panel

The Utilities options (see Figure 6.17) includes user documentation, access to support contacts, version information and login/logout functions.

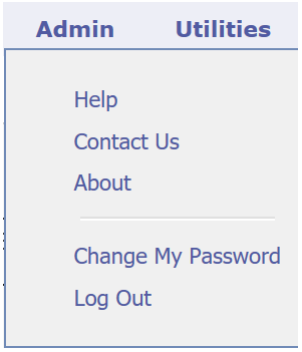


Figure 6.17: Utilities Menu allowing password change

6.3.3 Curation process

The tranSMART database stores clinical attributes separately from high dimensional molecular attributes. Clinical attributes are stored in a flexible data structure capable of accommodating large variations in study designs and collected data elements. However, curators need to map each field of a data file to a data element, or concept, in the database. This is done using a mapping file that, when complete, directs the ETL software in populating the database. Most curators create this mapping file manually. Mapping file development requires both a good understanding of the study and its data as well as the ability to anticipate the navigation and interrogation preferences of the users. Typically, the mapping file will be developed iteratively with user feedback during the development process. Ideally, the input data that is mapped for ETL will be well prepared, standardized datasets as discussed in chapter 4. It should be clear to the reader that he curation process requires diligence as well as both scientific and data management. This process becomes much more intensive if multiple studies will be interrogated in concert.

High dimensional datasets are stored in separate data structures. The mapping is more straight forward provided that the datasets are of good quality but care must be taken to ensure that the corresponding clinical and high dimensional datasets are referenced properly. The gene expression data was normalized using a standard protocol should the raw files be used, or the intensities were downloaded from the source systems. The phenotypes were manually turned into CDISC SDTM concepts that then were stored in a standardized hierarchy.

6.4 Summary

The tranSMART system allows clinicians, translational scientists and discovery biologists to interrogate aligned phenotype/genotype data to enable better clinical trial design or to stratify disease into molecular subtypes with great efficiency. A fine-grained, role-based authorization model throughout the application has been implemented so that study level permissions are enabled and can be controlled by the study owners. During curation the study owners are actively involved in reviewing and approving the loading and standardization of the data from

their studies. This approach greatly enhanced the cooperation of the study owners and the ultimate success of the data warehouse.

This chapter introduced a system that meets many, although certainly not all, of the ideal translational information processing goals described in chapter 4 and can be used freely by anyone. At the time of this writing a hosted implementation of tranSMART, created by the eTRIKS project, is available at https://portal.etriks.org/portal/. There are over 80 public-domain studies available on this tranSMART instance which can be effectively used to assess tranSMART for fit for purpose for projects of interest. Moreover, readers wishing to learn further details should review documentation on the tranSMART/I2B2 Foundation site (https://transmartfoundation.org/) and/or contact the foundation for additional information regarding tranSMART resources.

Chapter 7: eTRIKS Analytical Environment: A Practical Platform for Biomedical Data Analysis

Axel Oehmichen

7.1 Toward large scale data analysis in Life Science

The volumes of data collected in medical sciences is increasing. An example from genomics illustrates the growth very well: next-generation sequencing has led to a rise in the number of human genomes sequenced every year by a factor of $10^{1\ 2}$ far outpacing the development in data analysis capacity. In addition to large scale sequencing facilities and the emergence of medical devices contributing to analytical challenges based on volume, data is also increasingly heterogeneous due to advances in instrumentation leading to a variety of physiologic measures that, with corresponding analysis methods and software, can be used for many purposes. Data may be incomplete, incorrect, inaccurate, and/or irrelevant depending on the type and source and requires substantial preparation prior to analytic use as discussed in prior chapters.

Massive amounts of data require scalable infrastructure to process and analyze. Moreover, the infrastructure needs to accommodate the steady emergence of novel algorithms and data processing, and integration tools. The eTRIKS Analytical Environment (eAE) is an example of such an infrastructure capable of analyzing and exploring massive amounts of medical data. The eAE is a modular framework with which users can rapidly add and replace analytics tools and modules. The system is built upon established technologies and has been demonstrated to manage scale with respect to increases in both the number of users and processed data volumes. The eAE has provided the computational environment for several successful research projects. OPAL[3], a project needing to analyze terabytes of location data for public health research and monitoring, has leveraged the eAE for data processing and analysis.

7.2 Design principles and core concepts

The eTRIKS Analytical Environment aims to enable the analyses of a very broad range of users ranging from biologists with limited computing background to computational specialists. Medical doctors with little or no programming experience must use interactive tools to perform an analysis. However, statisticians and bioinformaticians have an extensive range of highly programmable command line mathematical environments, such as R, as well as complete programming language environments, such as Python, that provide extensive support for numerical computing and machine learning. Regardless of the toolset chosen, large scale

analysis requires seamless and highly efficient transfer of data between data sources and distributed high-performance computational environments.

Cost and administration efficiencies of multitenant operation and the ability of a single computing environment to host multiple unrelated applications and datasets simultaneously while individual users perceive their applications as being run on isolated infrastructure, were important. Collaboration support among users likely distributed geographically, which may seem contrary to multitenancy, was also an imperative.

7.2.1 General Environment

The eAE was designed with four layers - *Endpoints Layer*, *Storage Layer*, *Management Layer*, and *Computation Layer*. These layers aim to provide ease of use, modularity and scalability. These layers are loosely coupled to provide as much flexibility as possible. The modularity of this framework enables the straightforward addition and replacement of architectural components.

The operating system used on both the physical and virtual machines as well as containers within this architecture is Ubuntu 16.04 LTS. This operating system is stable and supports a large spectrum of libraries and drivers. Other Linux distributions such as Centos or Debian can also be used and upgrading to version 18.04 LTS is possible.

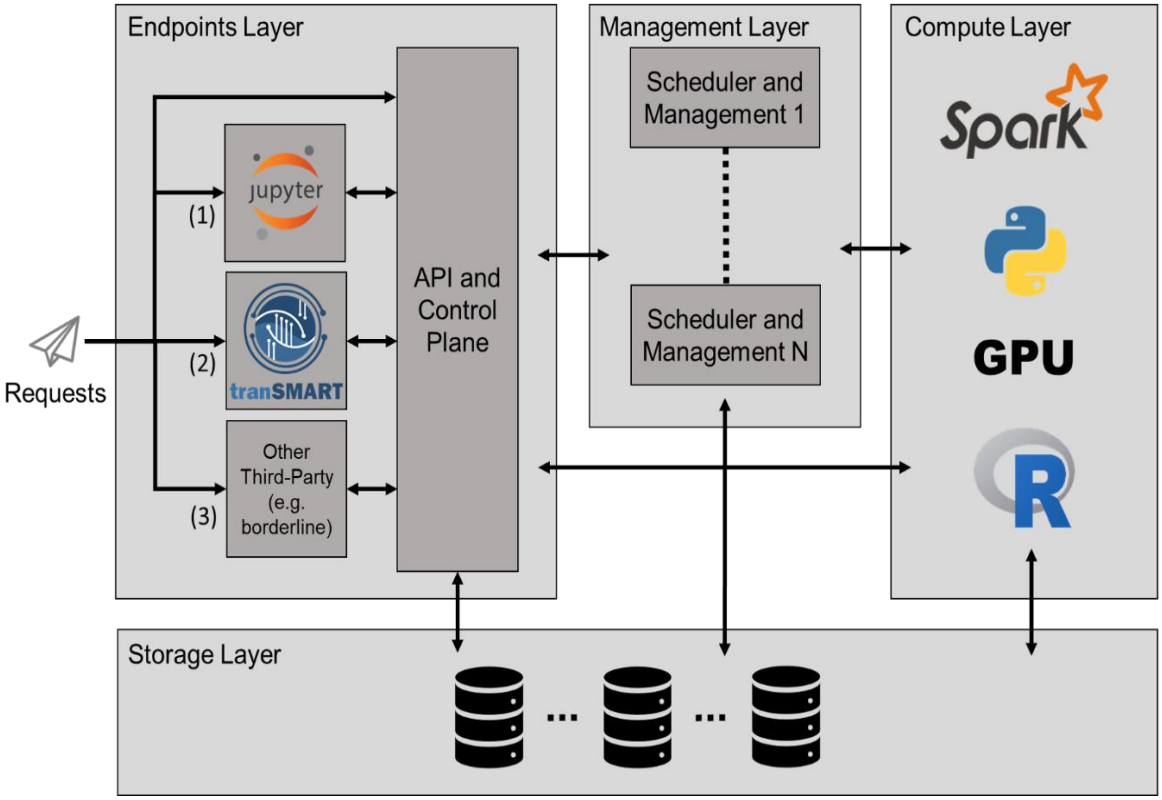


Figure 7.1: A schematic representation of the architecture of the eTRIKS Analytical Environment.

Each service is deployed in a Docker container (Docker is a platform that bundles applications with operating systems such that these can be deployed easily and consistently to a variety of hardware platforms) and the services communicate with each other asynchronously through REST APIs. This architecture supports continuous deployment across different host machines for scalability and resiliency. The platform is hosted behind a firewall and only the *Interface service* in the *Endpoints Layer* is exposed to the internet while all other services are interrelated only via an internal virtual private network.

7.2.2 Endpoints Layer

The *Endpoints Layer* hosts the containers which either provide the User Interface (UI) or the REST API allowing users to integrate third party external tools or interact directly with the platform. The *Endpoints Layer* also contains infrastructure to run small computations locally as well as user authentication, caching and auditing services.

As set up at Imperial College London, the endpoints layer includes tranSMART 16.2, a novel translational research user interface called Borderline[4] and a modified version of the Jupyter notebook for general analysis. The eAE scales the existing tranSMART advanced workflows while Jupyter and Borderline can each be used to create custom analysis processes and visualizations as well as to manage data lineage. The eAE logs all requests and check requests before launching analysis jobs.

The *Interface service* provides the client APIs for queries and user management. All the requests to the eAE are made over HTTPS and are first verified by the *Authentication service*. The authentication method has evolved over progressive versions and will be detailed in the implementation section. Upon successful authentication, the *Interface service* validates the request and, if appropriate, the job request is created for the *Management Layer* to schedule.

The *Logging service* logs the queries made to the platform. The log records all queries, both valid and invalid, for audit purposes and traceability. The invalid queries must be easily accessible to enable administrators for periodic analyses to detect trends of any possible attack on the system. Thus, the valid queries are stored in an append-only text file to avoid tampering.

7.2.3 Storage Layer

The *Storage Layer* maintains analytics results and enables caching (limited term storage of recently used data) to be implemented for specific endpoints (see tranSMART example) to avoid successive computing of the same analysis, thereby making analysis more efficient. All the large-scale data and meta-data associated with the platform (e.g. user details) are saved into the database via the *Storage Layer*. The *Storage Layer* supports all the other layers by providing replicated, distributed and scalable storage resources.

The Storage Layer provides scalable, replicated and sharded (automated partitioning of large data sets to increase retrieval performance) memory structures. Scalability and sharding empowers large-scale analyses while replication protects the platform against data loss due to node failures, network disruptions and other unexpected deleterious events. The eAE uses the MongoDB v3.6.0 NoSQL database, OpenStack for large scale object storage and PostgreSQL as RDBMS.

MongoDB does not enforce rigid data structures (a.k.a schemas) and this flexibility is useful for creating the generalized cache service and for storing simple operational data such as usernames, query parameters, logging information, etc. MongoDB's native support for high throughput read operations coupled with a powerful query language made this database a good choice for implementing the *Scheduling service*.

OpenStack Swift is a high availability, distributed object store that was leveraged for storing large files (up to several terabytes) that will be retrieved in total. Swift enables to store a very large amount of data efficiently, safely, and cheaply.

7.2.4 Management Layer

The *Management Layer* schedules the computation of the analyses on the compute nodes based on the availability of the nodes and type of analyses requested by the platform users. High availability is crucial for the management layer as is extensibility to ensure that new compute nodes can be used as soon as these are introduced to the compute cluster.

The *Scheduling and Management service* guarantees that the system performs efficiently by periodically purging inactive jobs and decommissioning unresponsive compute nodes. Multiple scheduling and management services run concurrently to maximize system availability.

7.2.5 Computation Layer

The *Computation Layer* is responsible for executing the scheduled computations on the scale-out infrastructure which can be a cluster, a cloud service or any other specialized hardware. It also enforces security and privacy as warranted for each analysis. Analyses results are stored in the *Storage Layer* along with the corresponding analysis parameters and other details pertaining to the analysis run.

The *Computation Layer* must efficiently support a broad scope of analytical capabilities ranging from simple statistics to compute heavy deep learning models. Heterogeneous hardware (e.g. CPU, GPU (graphic processing) or ASIC (custom processing chips) are made available with the addition of new nodes possible without downtime.

Jobs are run privacy-preserved while supporting on demand resource allocation (i.e. automated scaling). Scaling is implemented by creating docker containers of analytical tools and using the

Popular opensource Kubernetes orchestration platform to deploy the analytical containers and scale these automatically as needed.

The Spark computation clusters, Hadoop file system (HDFS), Swift and MongoDB are all installed in bare metal environments with RAID-10 storage to obtain the best performances possible and fault-tolerance. Those instances are password protected and running in secure (HTTPS) mode to ensure private network communications. MongoDB connections are encrypted using the transport security layer protocol (TSL/SSL). The eAE currently uses Cloudera CDH 5.9.0 Hadoop deployment.

The eAE would need to be adapted for certain cloud service, such as Amazon Web Services, or in-house compute clusters having alternate specifications.

7.2.6 Interaction between layers

Figure 7.1 illustrates the architecture of the eTRIKS Analytical Environment.

1. Each user owns a Virtual Machine (VM) or a docker container having a modified version of the Jupyter server, a set of kernels (R, Python, Spark, etc.) and a minimal set of standard libraries (Numpy, Scipy, Scikit-learn}, Bioconductor, etc.). This instance is one of the points of access of the eTRIKS Analytical Environment. The users can upload their data sets to the server and write their own analysis scripts. Jupyter, through the selected kernel, sends the requested computations to the local engines which in turn send results back to Jupyter. If the user requires more compute power, they can remotely submit their script to the interface service to be scheduled on a larger centralized cluster. When the required resources become available, the scheduler launches the computation. The Spark clusters are Hadoop stack production clusters installed on physical servers for performance reasons. Each one runs CDH 5.9.0 with the full Hadoop stack. The GPU clusters rely on TensorFlow 1.0 for Deep Learning and Nvidia CUDA processors. The R servers rely on Microsoft R Open, formerly known as Revolution R Open (RRO), which is the enhanced distribution of R from Microsoft Corporation. The results are sent back to Jupyter or MongoDB (depending on the user's choice). The user can explore the results using advanced visualizations (lightning, etc.).
2. The second native entry point to the eAE is through a tranSMART's plugin specifically developed for this integration. The plugin manages and interfaces with the MongoDB cache. The plugin can submit a job to the Interface service using data stored either in MongoDB or in tranSMART. The results are sent to the MongoDB cache. The user can explore the results in tranSMART and compare with previously run computations held in their personal cache history.
3. The third native entry point to the eAE is through Borderline. Borderline is a user facing sets of services responsible for locating data, querying it across multiple heterogeneous

sources, tracking its provenance as it travels through the platform and allowing users to maintain complete control over the process. Borderline makes the eAE and the eTRIKS Data Platform (eDP) including tranSMART and the eTRIKS Harmonization Service interoperable (Chapter 4). Besides allowing seamless data flow and tracking between these components, borderline provides enriched user experience. Borderline provides a dynamic query editor for selecting patient subgroups from the breadth of data available on a tranSMART instance.

7.2.7 Security of the architecture

The eAE manages risk using a combination of server-side security, authentication, audit and network security. However, those protections are only the core layer upon which adopters can build upon and further extend the eAE to meet the most stringent requirements.

Server-side security: Many attacks on privacy and services employ a relatively large number of queries to circumvent protections (e.g. DDOS attacks, data leakage, etc.). To thwart brute-force attacks on the client API, we developed a query rate limitation mechanism. An analyst can submit only a limited number of queries within a set period of time as defined by the curator (e.g. 100 queries in 7 days). The architecture supports secure execution of algorithms in sandboxed environments. This execution isolation comes at a cost to performances but prevents rogue algorithms from inappropriately accessing other computations which might be running at the same time on the platform. The sandboxing relies on AppArmor ("Application Armor") which is a Linux kernel security module. The module supplements the traditional Unix discretionary access control (DAC) model by providing mandatory access control (MAC).

Authentication: Access is provided only for authenticated users having the right authorization. Three levels of users are supported: super admin, admin and standard users. Admins can create, delete and check users through the API as well as monitor the status of the services. Super admins have the additional right to create new admins. In addition to those core levels, the users are given additional rights levels for data and analysis control. For example, in the context of the population density algorithm (see in Privacy section), different users will be authorized different levels of regional access: one user might be authorized to access at commune level while another only at regional level. The granularity can be temporal as well where the access time frame can be larger for some users compared to others. Some further restrictions can also be implemented such as maximum sampling size for a given analysis.

Audit: Auditing is an important part of the security of the platform as it enables system administrators, governance board members for ethical oversight and data owners to review all previous queries and detect tentative of attacks by logging illegal requests. The auditing helps preserve the health of the clusters as well by providing the computation times of the queries and cluster loads to the administrators. Those indicators can help them identify nodes that might be throttling or clusters which are over/under utilized. The administrators could then act on

them by commissioning or decommissioning nodes as pertinent and thus provide the best experience to users.

Network security: To prevent attacks that intercept HTTP packets, all communications with the API and between the different services will be exclusively done in HTTPS. Any non-HTTPS requests will be discarded and logged for auditing purposes. Furthermore, the connection of the services to MongoDB are encrypted using TSL/SSL. To shield the platform from external brute force attacks, the layers are deployed across two different VLANs. This distributed architecture exposes only the *Interface service* of the *Endpoints Layer* to client's applications while the data and compute services remain safely hidden from the rest of the network.

7.2.8 Comparison with similar products

A general-purpose analytical platform usually refers to systems that allow analysts to send many queries of different types using a rich and flexible query language. Many federated and distributed systems have been developed to analyze medical data. Among the most prominent ones (excluding the eTRIKS Analytical Environment) include IBM Platform Conductor, Arvados, Berkeley Open Infrastructure for Network Computing (BOINC) and Petuum. Those platforms share some common features (scalability, scheduling, storage, etc.) although each was designed for specialized user needs.

Platforms	eTRIKS Analytical Environment	IBM platform Conductor	Arvados	BOINC	Petuum
Visualizations capabilities					
Jupyter	●	●			
Zeppelin		●			
Borderline	●				
Custom	●	●	●		●
Analysis support					
Spark	●	●			
Python	●	●	●		
R	●	●	●		
C/C++				●	
Fortran				●	
Java		●	●		
Go/Ruby/Perl			●		
Fixed set of ML pipelines					●
Computation types					
CPU	●	●	●	●	●
GPU	●	●		●	●
Storage capabilities					
SQL	●	●			●
NoSQL	●	●			●
Content-Addressable Storage	○	●	●		
Monitoring and scheduling capabilities					
Jobs Status	●	●	●	●	●
Clusters Status	●	●	○	●	●
Complex batch processing	●	●	○	●	●
Multi-master scheduling	●	●		●	
Workflow capabilities	○	●	●		○
Data security capabilities					
Data provenance	○	●	●		
Extensive platform audit	●	●	●	●	
Privacy	●	○			
Secure computation (sandboxing)	●	○			
Support of GDPR compliance	●	○			
Interoperability					
REST API	●	●	●		●
Distributed Clients	●	●	●	●	●
Platform support					
Installation procedures	●		●	●	●
Configuration documentation	●	●	●	●	●
Support available	●	●	●	●	●
Open Source project	●		●	●	○

● Fully supported
○ Partially supported

Table 7.1: Summary of the differences in the main features provided by comparable existing systems.

It is interesting to notice that beyond the technical capabilities, new types of features (e.g. privacy and provenance) start to emerge as new problematics and legislation are arising.

7.3 Case Studies with the eTRIKS analytical environment: analytics for tranSMART

To illustrate how the eTRIKS Analytical Environment can be used for managing and analyzing large scale translational research data in tranSMART, we have implemented three bioinformatics analysis pipelines: an iterative model generation and cross-validation pipeline for biomarker identification, a general statistical analysis pipeline for hypotheses testing, and a pathway enrichment pipeline using KEGG to demonstrate the performance of the proposed

architecture. Unlike the two others, the pathway enrichment forms part of the iterative model generation pipeline or a pipeline on its own.

Each pipeline was implemented in the same fashion: the code was prototyped locally in a container (to ensure that the code operated as expected using a subset of the data or a smaller number of iterations) and the full computation was then submitted to the central clusters. All those workflows are designed to be highly parallelizable and, to enable their seamless scalability, Spark has been chosen for their implementation.

7.3.1 Iterative Model Generation and Cross-validation Pipeline

The iterative model generation and cross-validation pipeline scales at the same rate as the underlying hardware, a crucial aspect given the massive amounts of data involved. During clinical trials, collecting additional samples, if possible, may be hazardous and costly. In these cases, cross-validation is a powerful approach to prevent from testing invalid hypotheses suggested by the data (called "Type III errors"[5]). Cross-validation is a technique for assessing how a statistical or computational model will generalize to an independent data set. It is mainly used in settings where prediction is the main objective, and one aims to estimate how good a predictive model is in practice. In a prediction problem, a model is usually given a dataset of known outcomes (i.e., training set) on which the model is trained, and a dataset in which outcomes are unknown, against which the model is tested (i.e., testing set). To reduce variability, the dataset can be partitioned for multiple rounds of cross-validation.

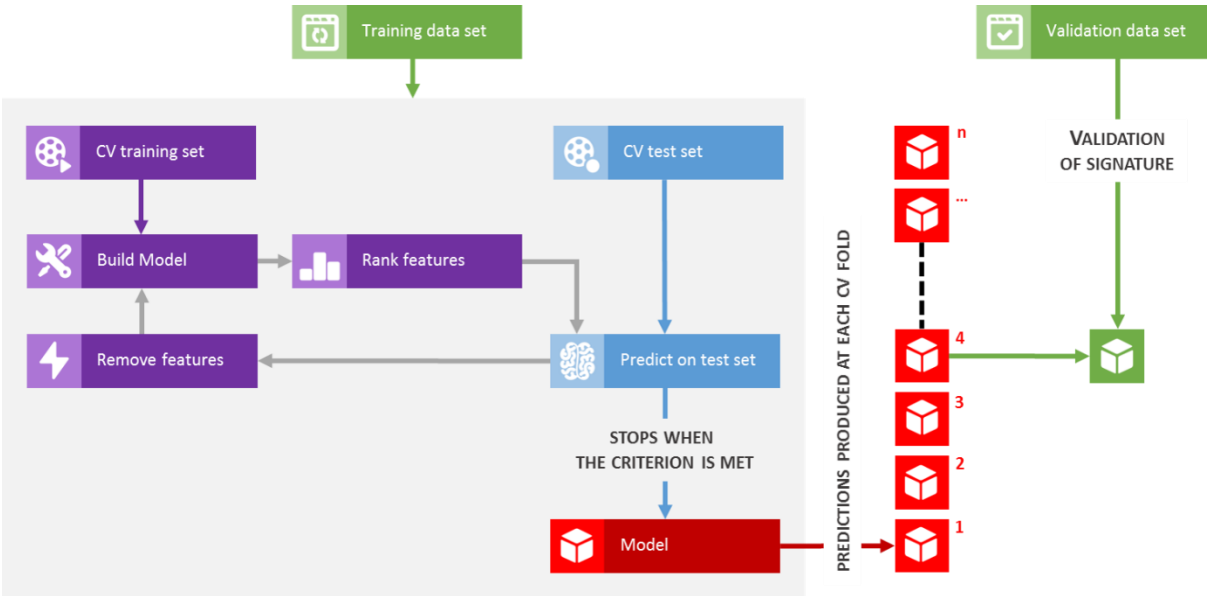


Figure 7.2: Iterative model generation and cross-validation pipeline

The workflow of the pipeline is shown in Figure 7.2, and it has been widely used in translational research, identifying the gene signature for stage II colon cancer as an example[6]. In practice,

many statistical and computational approaches can be used for model generation. Possible candidates include, but not limited to, linear or nonlinear Support Vector Machine (SVM), Logistic Regression, Linear Regression, Alternating Least Squares, Lasso. Indeed, instead of distributing subsets of the data across different nodes (first type of parallelization) to parallelize the computation of the model, we can distribute different sets of parameters (second type of parallelization) on every worker to generate a different model every time.

The drawback with the second type of parallelization compared to the first one is that when dealing with large datasets, the working nodes need to be equivalently large and the network might become a performance bottleneck.

This pipeline allows us to scale by distributing the computation to multiple clusters that work independently to generate models. Each cluster randomly samples the training set and starts generating models using the selected algorithm. Each model is then tested against the test set to evaluate its fitness according to the specified set of indicators. With the increasing number of iterations, it is expected that the model will converge to the optimal solution.

Biomedical datasets are typically comprised of large numbers of features, i.e. individual measurable properties that describe the observed phenomenon. To find the best fit model having the least amount of bias a family of models are generated using the same dataset with different selections of features. Once the model is built, a certain number of features can be removed, and a new model is generated using the remaining features (Figure 7.2). An unbiased approach randomly removes a selected number of features and then determines whether the reduced feature set improves the fitness of the model. If the fitness is improved the process progresses to the next iteration. If the fitness does not improve then another set of features is removed, and the model is again assessed. A fixed small number of features removed at every iteration will promote the predictive accuracy of the model but is computationally expensive. Stepwise feature reduction based on feature set size (e.g., a percentage of the total number of features) may speed up the process.

A more efficient approach is to introduce a small amount of bias by lowering the chances of a feature, which we know, or highly suspect, a priori to be a factor, being removed. By introducing this bias, models will naturally tend to converge to a (presumably) more optimal solution much faster. Vladimir Vapnik's group[7], when working in the context of gene selection, suggested using weight magnitude as ranking criterion for features, computing the ranking criteria: $c_i = (w_i)^2$ for all “i” and find the feature with smallest ranking criterion f = argmin(c).

The scoring used to assess the fitness of models can be done through a wide variety of measures, such as Area Under the Receiver Operating Characteristic (ROC) Curve - AUC, Sensitivity or True Positive Rate (TPR), Specificity or True Negative Rate (TNR), Negative predictive value (NPV), Positive predictive value (PPV) and F1-score. No metric is suitable

for every situation. These metrics can only eliminate obvious "failures" due to performance, complexity, overfitting or stability. The Hazard Ratio (HR) from Cox proportional hazards regression[8] can be used to adjudicate models.

Once distributed processing is complete the results are written to a NoSQL cache on the eAE from which candidate models compared and the information concerning the models are provided to the analyst such that the best model(s) can be selected.

This type of unbiased approach to model generation is not well supported on standard platforms. Model generation typically requires long run times of hours to days, risking loss of intermediate results in cases of unexpected infrastructure failures (i.e. crashes). Mechanisms would need to be implemented to avoid rerunning the entire job in the event of a premature halt due to an infrastructure failure. The eAE leverages versioning mechanisms of Spark to persist intermediate datasets and greatly reduce the impact of infrastructure failure.

The integration of Spark with the eAE enables users to run these large-scale compute intensive experiments easily and seamlessly through the application/interface of their choice. The stability, robustness and fault-tolerance of the platform enables high performance computations, even if a physical machine or a worker fails, as failed tasks are automatically rescheduled. The integration of Docker, Jupyter, Toree and the eAE have enabled users to implement and prototype their algorithms efficiently without the need to prepare custom analysis environments in which these components are integrated separately for each project undertaken. Finally, once the algorithm is ready, submission to the centralized high-performance cluster requires no further development.

7.3.2 General statistics

The general statistical analysis pipeline aims to provide statistical insights about datasets, without any prior statistical knowledge, by performing multiple statistical tests on a given data set. Statistical methods test scientific theories when observations, processes or boundary conditions are stochastic. Performing multiple tests on the same data set at the same stage of analysis increases the chance of obtaining at least one invalid result. The benefit obtained from performing statistical methods across whole datasets, however, far offsets this drawback.

The first step of this pipeline is to divide the data into their basic data types: numerical, binary, categorical, and unknown. Any data element with three or less valid data points and any irrelevant data (e.g. phone numbers and free text) are assigned to the unknown category. These data are not discarded as they might be used to extract insights at a later stage. The numerical data is next subset into groups of normally distributed and non-normally distributed datasets. Two methods are used to determine whether a variable follows a normal distribution: Shapiro-Wilk's test and Anderson-Darling's test. The variable is tagged as normally distributed only if

both tests yield a positive answer. Applicable statistical methods are then applied to the data in each category.

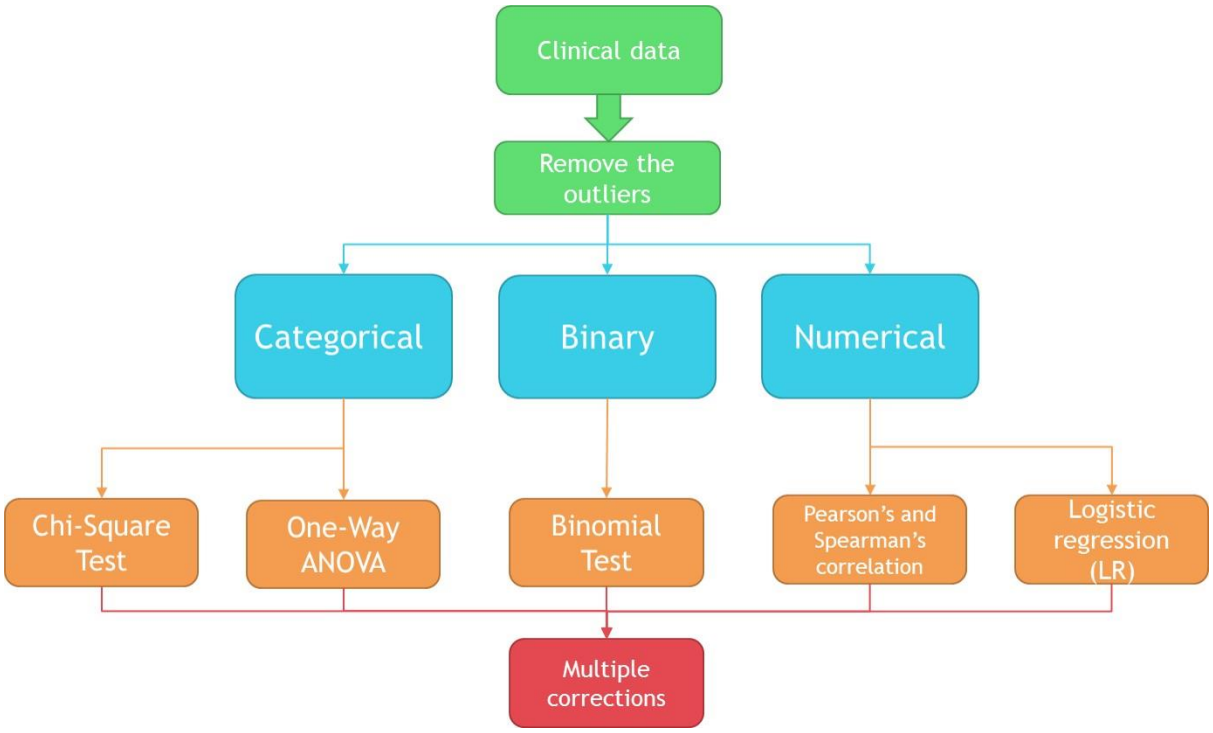


Figure 7.3: Modelling of the pipeline for an unbiased approach to statistical testing of whole datasets.

For the categorical data, the χ^2 test is extensively used for assessing the associations between different clinical variables. The χ^2 test determines whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. If one variable is categorical and one is numerical, the analysis of variance (ANOVA) test is used to statistically determine whether the means of several groups are equal, which complements the χ^2 test. The binomial test, an exact, two-sided test of the null hypothesis that the probability of success in a Bernoulli experiment is p (we chose p=0.5), is used for binary data elements.

We use two methods to analyze the relationships between numerical variables, logistic regression (LR) analysis and correlation analysis. LR has been successfully used to identify independent predictors of prostate cancer to improve the accuracy of diagnosis[9]. LR models are based on the fit of the odds of comparable conditions requires and no specific distribution assumption (e.g., Gaussian distribution). However, LR is often found to be less sensitive than other approaches. For correlation analysis, we choose the Spearman and Pearson correlations. The Spearman correlation between two variables is equal to the Pearson's correlation between the rank values of those two variables; while Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships. If there are no repeated data values, a perfect Spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other. Bonferroni's correction is used for multiple corrections.

7.3.3 Pathway Enrichment

A common approach to interpreting gene expression data is gene set enrichment analysis based on the functional annotation of the differentially expressed genes. This is useful for finding out whether the differentially expressed genes are associated with a certain biological process or molecular function. The Gene Ontology, containing standardized annotation of gene products, is commonly used for this purpose. The approach works by comparing the frequency of individual annotations in the gene list (e.g. differentially expressed genes) with a reference list (usually all genes on the microarray or in the genome).

To help the scientists to select the right model after the list of genes has been output by the Iterative Model Generation pipeline or simply do a pathway enrichment using a list of genes obtained from another analysis, a pathway enrichment using KEGG and GO with multiple test corrections (Bonferroni, Holm-Bonferroni and/or FDR) on the sub selection of high scoring models has been implemented. The enrichment is done using a two-sided Fisher's exact[10] test after building the associated contingency table.

Those enrichments add another insight to the results and an additional quality check to every model generated. The Spark implementation facilitates large-scale enrichments by building the models concurrently unlike traditional platforms that wait for the final model to be output before performing the enrichment. Moreover, the number of pathways and their complexity gradually increases which requires correspondingly increasing compute power. The proposed implementation offers a way to overcome the increasing number of pathways and allows integration of the enrichment.

The following example describes the application of the eAE to a real-world research program.

7.4 DeepSleepNet: An eAE Case Study

This work was carried out in collaboration with researchers at the Data Science Institute at Imperial College London who specialize in researching sleep disorders. This example highlights well the benefits researchers can gain by leveraging the eTRIKS Analytical Environment and TensorLayer. This specific research proposes a new deep learning model, named DeepSleepNet[11], for automatic sleep stage scoring based on raw single-channel Electro Encephalogram (EEG).

7.4.1 Introduction

Sleep plays an important role in human health. Being able to monitor how well people sleep has a significant impact on medical research and practice[12]. Typically, sleep experts determine the quality of sleep using electrical activity recorded from sensors attached to different parts of

the body. A set of signals from these sensors is called a polysomnogram (PSG), consisting of an electroencephalogram (EEG), an electrooculogram (EOG), an electromyogram (EMG) and an electrocardiogram (ECG). This PSG is segmented into 30 second epochs, which are then be classified into different sleep stages by the experts according to sleep manuals such as the Rechtschaffen and Kales[13] and the American Academy of Sleep Medicine[14]. This process is called sleep stage scoring or sleep stage classification. This manual approach is, however, labor-intensive and time-consuming due to the need for PSG recordings from several sensors attached to subjects over several nights.

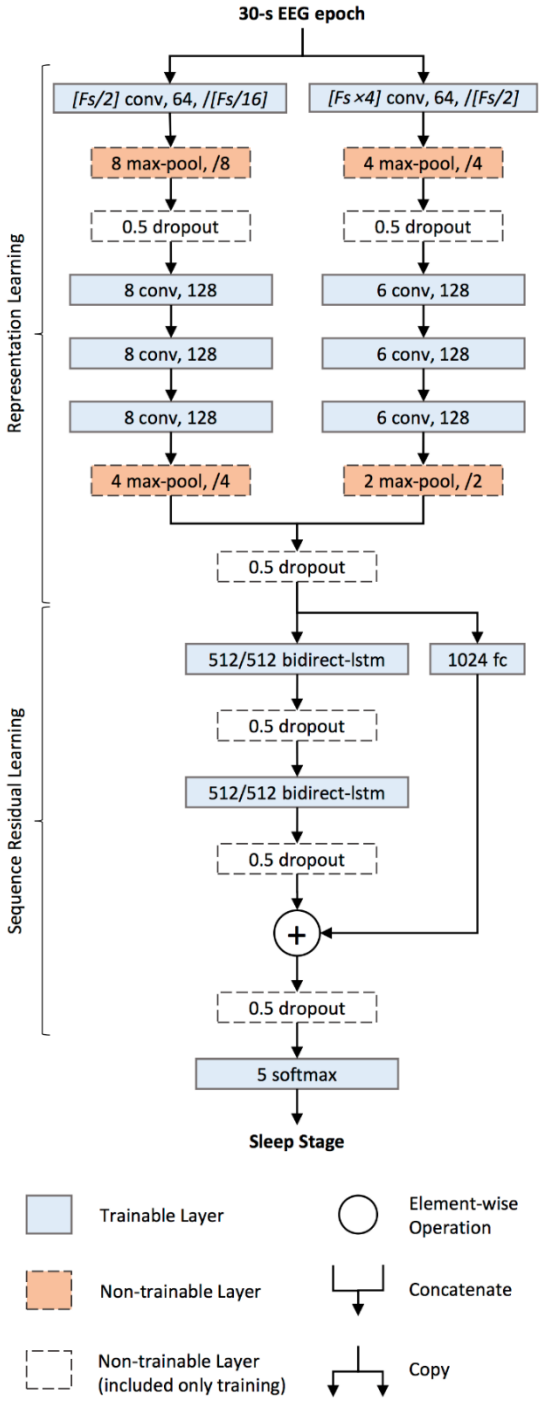


Figure 7.4: An overview architecture of DeepSleepNet from Supratak et al.

The overview is consisting of two main parts: representation learning and sequence residual learning. Each trainable layer is a layer containing parameters to be optimized during a training

process. The specifications of the first convolutional layers of the two CNN depends on the sampling rate (Fs) of the EEG data.

The new approach proposes a model for automatic sleep stage scoring based on raw single-channel EEG by utilizing the feature extraction capabilities of deep learning. The architecture of DeepSleepNet consists of two main parts as shown in Figure 7.4. The first part is representation learning, which can be trained to learn filters to extract time-invariant features from each raw single-channel EEG epochs. The second part is sequence residual learning, which can be trained to encode the temporal information such as stage transition rules from a sequence of EEG epochs in the extracted features.

7.4.2 Representation Learning

Two CNNs with small and large filter sizes were employed at the first layers to extract time-invariant features from raw single-channel 30-s EEG epochs. The small filter is well suited to capture temporal information, while the larger filter is better suited to capture frequency information.

The model has been designed with four convolutional layers and two max-pooling layers for each CNN. Each convolutional layer performs three operations sequentially: 1D-convolution with its filters, batch normalization[15], and applying the rectified linear unit (ReLU) activation (i.e., *relu(x)=max(0, x)*). The max operation has been used in each pooling layer to down sample inputs. Figure 7.4 illustrates the specifications of the filter sizes, the number of filters, stride sizes and pooling sizes. Each {*conv* and *max-pool* block shows a filter size, the number of filters, and a stride size. The dropout blocks have been put in place to help preventing overfitting, more details will be provided in Section regularization.

$$\begin{aligned} \mathbf{h}_t^f, \mathbf{c}_t^f &= LSTM_{\theta_f}(\mathbf{h}_{t-1}^f, \mathbf{c}_{t-1}^f, \mathbf{a}_t) \\ \mathbf{h}_t^b, \mathbf{c}_t^b &= LSTM_{\theta_b}(\mathbf{h}_{t+1}^b, \mathbf{c}_{t+1}^b, \mathbf{a}_t) \\ \mathbf{o}_t &= \mathbf{h}_t^f || \mathbf{h}_t^b + FC_\theta(\mathbf{a}_t) \end{aligned}$$

In order to extract the i-th feature a_i from the i-th EEG epoch X_i, and assuming there are N 30-s EEG epochs X_1, …. X_n from a single-channel EEG, we use two CNNs in the following fashion:

The 30-s EEG epoch X_i are transformed into feature vectors h_i using the function called CNN(X_i) which uses a CNN). θ_s and θ_l are parameters of the CNNs with small and large filter sizes in the first layer respectively. The outputs from the two CNNs are concatenated by the || operation. {a_1, …, a_n} are then forwarded to the sequence residual learning part as linked features.

7.4.3 Sequence Residual Learning

The sequence residual learning part was based on the residual learning framework theory[16]. Bidirectional-LSTMs[17] and a shortcut connection (see Figure 7.4 are the two main components of this part.

Two layers of bidirectional-LSTMs were employed to learn temporal information such as stage transition rules[18] which sleep experts use to evaluate the most likely forthcoming sleep stages based on the previous stages. For example, the AASM manual recommends that if a subject is in sleep stage N2, but scores epochs with low amplitude and mixed frequency EEG activity, will still be classified as N2 even though K complexes or sleep spindles are not present. The bidirectional-LSTMs can learn to remember that specificity it has seen in stage N2 and continue to score successive epochs as N2 even if they still detect the low amplitude and mixed frequency EEG activity. Bidirectional-LSTMs are an extension of LSTM[19] by adding two LSTMs process forward and backward input sequences independently[20]. As the outputs from forward and backward LSTMs are not connected to each other, the model is capable to use information both from the past and the future. In order to inspect their current memory cell before the modification, we used peephole connections[21] [22] in our LSTMs.

With the intent to enable the addition of temporal information to our model, we used a shortcut connect to reformulate the computation of this part into a residual function. Therefore, the model can learn from the previous input sequences into the feature extracted from the CNNs. A fully connected layer in the shortcut connection was used to transform the features from the CNNs into a vector which in turn is added to the output from the LSTMs. The matrix multiplication with its weight parameters, batch normalization, and the application of the ReLU activation are then carried out by this layer sequentially.
Formally, suppose there are N features from the CNNs $\{a_1, \ldots, a_N\}$ arranged sequentially and t=1...N denotes the time index of 30-s EEG epochs, our sequence residual learning is defined as follows:

- LSTM represents a function that processes sequences of features a_t using the two-layers LSTM parameterized by θ_f and θ_b for forward and backward directions.
- **h** and **c** are vectors of hidden and cell states of the LSTMs. h_0^f , c_0^f , $h_{N+1}^b$and c_{N+1}^b of forward and backward LSTMs are set to zero vectors.
- FC represents a function that transform features a_t into a vector that can be added (element-wise) with the concatenated output vector $h_t^f \parallel h_t^b$ from the bidirectional-LSTMs.

Figure 7.4 presents the specifications of the hidden size of forward and backward LSTMs along with the fully connected layers. The *fc* block shows a hidden size, while each *bidirect-lstm* block shows hidden sizes of forward and backward LSTMs.

During the training and testing of the models, the hidden and cell states states h_t^f ,h_t^b , c_t^f and c_t^b are re-initialized to zeros at the beginning of each patient data. This is put in place to ensure that only temporal information from the current subject data is used by the model for both training and testing.

7.4.4 Model Specification

For the representation learning part, we followed the guideline provided by Cohen et al.[23] for capturing temporal and frequency information from the EEG when selecting the parameters of the CNN-1 and CNN-2. On one hand, the filter size of the *conv1* layers of the CNN-1 was set to Fs/2 (half of the sampling rate (Fs)), and its stride size was set to Fs/16 to detect when certain EEG patterns appear. On the other hand, the filter size of the *conv1* layer of the CNN-2 was set to Fs x 4 to better capture the frequency components from the EEG. And, as it is not necessary to perform a fine-grained convolution to extract frequency components, its stride size was also set to Fs/2 (which is higher than the *conv1* layer of the CNN-1). Finally, based on Szegedy et al.[24] observation that the use of multiple convolutional layers with a small filter size (instead of a single convolutional layer with a large filter) can reduce the number of parameters and the computational cost while retaining similar level of model expressiveness, we adopted small fix sizes for the filter and stride sizes of the subsequent convolutional layers *conv2_[1-3]*.

The parameters of the *bidirect-lstm* and *fc* layers were set to be smaller than the output of the representation learning part (set to 1024) for the sequence residual learning part. This method was put in place to prevent overfitting and that only the important features get selected and combined by our model.

7.4.5 Two-Step Training Algorithm

Models built on large sleep datasets usually suffer from class imbalances issues (i.e., learning to classify only the majority of sleep stages). In order to prevent this from happening, we developed the two-step training algorithm as solution to effectively train our model end-to-end via back propagation and prevent the model from suffering class imbalance problem. The representation learning part of the model is first pre-trained by the algorithm, then the algorithm fine-tunes the whole model using two different learning rates. We used the cross-entropy loss to quantify the agreement between the predicted and the target sleep stages in both training steps. The last layer in the DeepSleepNet architecture (see Figure 7.4) is a combination of the softmax function and the cross-entropy loss which are used to train our model to output probabilities for mutually exclusive classes.

Algorithm 1: Two-step Training

Input: $init\_model$, $data$

Output: $model$

Initialization:

1: $init\_CNN_{\theta_s,\theta_l} \leftarrow extract\_cnns(init\_model)$
2: $pre\_model \leftarrow stack(init\_CNN_{\theta_s,\theta_l}, softmax)$
3: $data_{over} \leftarrow oversample(data)$

Pre-training Step:

4: **for** $i = 1$ **to** $n\_pretrain\_epochs$ **do**
5: **for each** $batch$ **in** $shuffle(data_{over})$ **do**
6: $pre\_model \leftarrow adam_{lr}(pre\_model, batch)$
7: **end for**
8: **end for**

Fine-tuning Step:

9: $pre\_CNN_{\theta_s,\theta_l} \leftarrow extract\_cnns(pre\_model)$
10: $model \leftarrow replace\_cnns(init\_model, pre\_CNN_{\theta_s,\theta_l})$
11: **for** $i = 1$ **to** $n\_finetune\_epochs$ **do**
12: **for each** $subject$ **in** $data$ **do**
13: $model \leftarrow reset\_lstm\_cell\_state(model)$
14: $subject\_data_{seq} \leftarrow arrange\_sequence(subject)$
15: **for each** $batch$ **in** $subject\_data_{seq}$ **do**
16: $model \leftarrow adam_{lr_1,lr_2}(model, batch)$
17: **end for**
18: **end for**
19: **end for**
20: **return** $model$

Pre-training

The pre-training step starts with a supervised pre-training on the representation learning part of the model (see lines 1-8 in Algorithm 1) with a class-balanced training set to avoid any overfitting on the majority of sleep stages. Specifically, the two CNNs are extracted from the model and then stacked with a softmax layer (softmax). The sole use of the stacked softmax step is to pre-train the two CNNs and the parameters are discarded at the end of the pre-training. We denote these two CNNs stacked with softmax as pre_model. Similarly, to the pre-training part, the pre_model is trained with a class-balanced training set using a mini-batch gradient-based optimizer called Adam[25] with a learning rate (lr) and the softmax layer is once again discarded at the end of the pre-training. The class-balanced training set is obtained by oversampling the minority sleep stages in the original training set until all sleep stages have the same number of samples.

Fine-tuning

The fine-tuning step starts with a supervised fine-tuning on the whole model (see lines 9-19 in Algorithm 1) with a sequential training set. This step encodes the stage transition rules into the model as well as the necessary adjustments on the pre-trained CNNs. The θ_s and θ_l parameters from the pre_model replace the ones from the CNNs of init_model which in turn result in a model. That model is then trained on the sequence training set using a mini-batch Adam optimizer but this time with two different learning rates (lr_1 and lr_2). The lower learning rate lr_1 is used for the CNNs part (as the CNNs part has already been pre-trained), while the higher learning rate lr_2 is used for the sequence residual learning part. A softmax layer is added after them. During the development phase, we noticed that using the same learning rate to fine-tune the whole network resulted in the excessive adjustment to the sequential data (which were not class-balanced) of the pre-trained CNN parameters. Consequently, the model started to overfit to the majority of the sleep stages toward the end of the fine-tuning. It is from this observation that we decided to use two different learning rates during fine-tuning. Furthermore, exploding gradients is a well-known problem when training RNNs such as LSTMs[26]. Therefore, we use a heuristic gradient clipping technique to prevent the exploding gradients by rescaling the gradients to smaller values using their global norm whenever they exceed a pre-defined threshold. The sequential training set has been constructed by reorganizing the original training set in chronological order across all subjects.

7.4.6 Regularization

As we highlighted before, overfitting has been a major issue we faced. To prevent overfitting problems, we used regularization techniques. During the training phase, and only the training phase, a dropout[27 28] technique that randomly sets the input values to 0 (i.e. dropping units along with their connections) with the specified probability was used. All dropout layers used a probability of 0.5 throughout the model.

Subsequently, we used a L2 weight decay technique which adds a penalty term into a loss function to prevent large values of the parameters in the model (e.g. exploding gradients). This technique was applied only on the first layers of the two CNNs. Indeed, as explained by Pascanu et al.[29], L2 weight decay can limit the model capabilities of learning long-term dependencies. Besides, we found that, without weight decay, the filters of the first layers of the CNNs overfitted to noises or artifacts in EEG data. By using an appropriate amount of weight decay, the model learned smoother filters (e.g. containing less high-frequency elements) which resulted in slightly performance gains. The weight decay parameter that defines the degree of penalty, lambda, was set to 10^{-3}.

7.4.7 Results

Data: Evaluation of the model against other datasets is important in order to access the quality of the model and we evaluated our model using different EEG channels from two public datasets: Montreal Archive of Sleep Studies (MASS)[30] and Sleep-EDF[3132].

MASS: The Mass dataset was organized in five subsets of recordings (SS1-SS5) which followed their research and acquisition protocols. Among those five subsets, we selected only SS3 which contained PSG recordings from 62 healthy subjects (age 42.5 ± 18.9). Each recording contained 20 scalp-EEG, 2 EOG (left and right), 3 EMG and 1 ECG channels. The EEG electrodes were positioned according to the international 10-20 system, and EEG and EOG recordings were pre-processed with a notch filter of 60 Hz, and band-pass filters of 0.30-100 Hz (EEG) and 0.10-100 Hz (EOG). All EEG and EOG recordings had the same sampling rate of 256 Hz. These recordings were manually classified into one of the five sleep stages (W, N1, N2, N3 and REM) by a sleep expert according to the AASM standard [33]. Subject's recordings exhibit movement artifacts at the beginning and the end of each record; those artifacts have been labelled as UNKNOWN. The evaluation was done without any further pre-processing using the F4-EOG channel, which was obtained via montage reformatting[34].

Sleep-EDF: The Sleep-EDF contained only two sets of subjects from two distinct studies. The first study was age effect in healthy subjects (SC), while the second one was Temazepam effects on sleep (ST). We used 20 subjects (age 28.7 ± 2.9) from SC. Each PSG recording contained 2 scalp-EEG signals from Fpz-Cz and Pz-Cz channels (@100Hz), 1 EOG (horizontal @100Hz), 1 EMG, and 1 oro-nasal respiration signal. Unlike in the MASS dataset, the R&K standard[35] was followed for the manual sleep stages classifications resulting in eight classes (W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN). The evaluation was done without any further pre-processing using the Fpz-Cz and Pz-Cz channels and merging the N3 and N4 stages into a single stage N3 to use the same AASM standard as the MASS dataset. Those two studies keep extensive amounts of stage W (awake) at the start and the end of each recording creating an imbalance with the other classes without bringing any valuable information. For that reason, we only included only 30 minutes of such periods just before and after the sleep periods.

Dataset	W	N1	N2	N3 (N4)	REM	Total
MASS	6227	4724	29534	7651	10464	58600
Sleep-EDF	7927	2804	17799	5703	7717	41950

Table 7.2: Number of 30-s epochs from Supratak et al. for each sleep stage from two datasets

Similarly, the MOVEMENT and UNKNOWN labels have been excluded as they did not belong to the five sleep stages[36]. Table 7.1 presents a summary of the number of 30-s epochs for each sleep stage from these two datasets.

Experimental Design and Implementation

The implementation of DeepSleepNet is an illustration of the user-friendliness for highly parallelisable computation and support for hardware accelerators (GPU in this instance) using the eAE. The eAE allows users to seamlessly and quickly configure and launch the complex training of multiple models concurrently.

The implementation of the project had two major constraints: the first one was that the project involved three different researchers working concurrently on the workflow, and the second one was that the GPU cluster was only available outside of working hours. To make things even more complicated, the GPU cluster had to be switched between Windows and Ubuntu daily (with the exception of weekends). This switch happens forcefully in an automated fashion interrupting the computations in the morning. Thus, it was of paramount importance that the eAE restarts the compute services seamlessly and reschedule interrupted jobs whenever a resource becomes available.

In order to overcome those constrains and develop the new workflow, an eAE Jupyter container was deployed on a specific box with two GPU resources available to enable the researchers to prototype their workflow simultaneously, share their code with one another seamlessly and access their data. To overcome the limited availability of the GPU cluster, the eAE's computes were deployed in containers that were configured to restart whenever the machine boots in Ubuntu. If the reported health of a compute node was unavailable, upon restart of the container, the eAE would commandeer the host machine of the failing container and attempt to restart up to three times the container automatically to ensure the good health of the cluster.

In order to build and assess the quality of our model, we used a k-fold cross-validation scheme on the eAE, where k was set to 31 for the MASS and Sleep-EDF datasets. In each fold, we used recordings from 60 subjects to train the model and use the two remaining subjects to test the model. This process is repeated 31 times so that all recordings are tested. Finally, we combine the predicted sleep stages from all folds and compute the performance metrics. We ran a large number (several hundreds) of 31-fold cross-validation iterations for hyperparameters tuning of

the model and various experiments. Each cross-validation task takes roughly 6-7h to execute and the total execution time consequently is 170.5 hours. The eAE enabled the researchers to schedule during the day two iterations to run every night without any intervention necessary as the tasks would be triggered as soon as the compute nodes become available thanks the eAE's scheduler and management services. Then, we combined the predicted sleep stages from all folds and computed the performance metrics, which will be discussed in Section Performance Metrics.

The only alternatives to this would be either to schedule the tasks on each machine of the cluster individually or sequentially on one machine. The former is tedious and far from practical as one needs to give the user access to all machines, and the latter simply takes too long. The eAE provides a user-friendly web UI, which allows to train multiple models with different configurations concurrently across a cluster of high-performance machines. The scheduling of these tasks through the eAE takes approximately 2-3 minutes compared to an hour if done manually. Another benefit is the possibility to queue jobs to be run once machines become available. For this experiment, the GPU resources were only available at night as they were used for other projects during the day. The option to schedule two iterations at a time for 31-fold cross-validation tasks to be run during the night, without any external intervention, is a key feature for the timely delivery of DeepSleepNet. In the case of this workflow, the experiments spanned almost an entire year and was made possible only thanks to the eAE.

Performance Metrics

The performances of the model were done using per-class precision (PR), per-class recall (RE), per-class F1-score (F1), macro-averaging F1-score (MF1), overall accuracy (ACC), and Cohen's Kappa coefficient (κ)[37] [38]. The per-class metrics are computed by selecting a single class as a positive class, and then combining all other classes as a single negative class. The MF1 and ACC are calculated as follows:

$$\mathrm{ACC} = \frac{\sum_{c=1}^{C} \mathrm{TP}_c}{N}$$

$$\mathrm{MF1} = \frac{\sum_{c=1}^{C} \mathrm{F1}_c}{C}$$

where TP_c is the true positives of class c, $\mathrm{F1}_c$ is per-class F1-score of class c, C is the number of sleep stages, and N is the total number of test epochs.

Training Parameters

The representation learning part was pre-trained using the oversampled training set with a mini-batch of size 100. The Adam optimizer's parameters lr, beta1, and beta2 were set to 10^{-4}, 0.9 and 0.999 respectively. Then, we equally split the sequences of 30-s EEG epochs from each subject data in the sequential training set into 10 sub-sequences (e.g. batch size was 10) to fine-tune the whole model. For each step training, we fed a sequence length (e.g. epochs) of 25 from each sub-sequence yielding 250 epochs per step. The Adam optimizer's parameters were similar to the pre-training step except that the learning rate of each part of the model were lr1 set to 10^{-6} and lr2 to 10^{-4}, while the threshold for the gradient clipping was set to 10. The pre-training and the fine-tuning steps were set to 100 epochs and 200 epochs respectively. Finally, as no validation was set in our evaluation scheme, no stopping had been put in place.

We relied on existing literature recommendations for the default values of the parameters such as beta1, beta2 and the threshold of the gradient clipping. We evaluated different mini-batch sizes (from 50 to 200) during the pre-training, batch sizes (from 5 to 40), sequence lengths (from 5 to 40) during fine-tuning, and the learning rates (from 10^{-3} to 10^{-6}) to obtain optimal performances in our model. For the batch normalization in *conv* and *fc* blocks, the ε constant of 10^{-5} was added to the mini-batch variance for numerical stability. The mean and variance of the training set, which were used as fixed parameters during testing, were estimated by computing the moving average with a decay rate of 0.999 from the sampling mean and variance of each mini-batch.

Initial Experiments

We initially conducted experiments for the design of the architecture and the parameters for DeepSleepNet with the first fold of the 31-fold cross-validation using the MASS dataset. For model architecture, we tried several configurations such as increasing/decreasing convolutional layers, changing the number of filters, the stride sizes, changing the number of hidden sizes in the bidirectional-LSTMs and the fully connected layer. The architecture in Figure 7.4 gave us the best performance. For regularization parameters, we tried several values for the weight decay parameters ranging from 10^{-1} to 10^{-5}. The value of 10^{-3} gave us the best performance.

For training parameters, we tried several values of learning rates ranging from 10^{-3} to 10^{-8}. We also experimented with the mini-batch size (from 50 to 200) during the pre-training, the batch size (from 5 to 40) and sequence length (from 5 to 40) during fine-tuning. Other parameters such as beta1, beta2 and the threshold of the gradient clipping were chosen from the default values reported in the literature. The training parameters mentioned in Section Training Parameters gave us the best performance. With these settings, the pre-training and fine-tuning steps started to converge after 100 and 200 epochs respectively.

Sleep Stage Scoring Performance

Tables 7.3 and 7.4 show confusion matrices obtained from the cross-validation on the F4-EOG and the Fpz-Cz channels from the MASS and Sleep-EDF datasets respectively. Fpz-Cz channel

yielded the best performance when compared with the Pz-Oz channel from the Sleep-EDF dataset, thus we did not include the confusion matrix obtained from the Pz-Oz channel. Each row and column represent the number of 30-s EEG epochs of each sleep stage classified by the sleep expert versus our model respectively. The numbers in bold indicate the number of epochs that were correctly classified by our model. The last three columns in each row indicate per-class performance metrics computed from the confusion matrix.

	Predicted					Per-class Metrics		
	W	N1	N2	N3	REM	PR	RE	F1
W	**5433**	572	107	13	102	87.3	87.2	87.3
N1	452	**2802**	827	4	639	60.4	59.3	59.8
N2	185	906	**26786**	1158	499	89.9	90.7	90.3
N3	18	4	1552	**6077**	0	83.8	79.4	81.5
REM	132	356	533	1	**9442**	88.4	90.2	89.3

Table 7.3: Confusion matrix from Supratak et al. obtained from the cross-validation on the F4-EOG channel from the MASS dataset

	Predicted					Per-class Metrics		
	W	N1	N2	N3	REM	PR	RE	F1
W	**6614**	745	181	81	306	86.0	83.4	84.7
N1	295	**1406**	631	30	442	43.5	50.1	46.6
N2	391	618	**14542**	1473	775	90.5	81.7	85.9
N3	29	9	291	**5370**	4	77.1	94.2	84.8
REM	360	457	419	7	**6474**	80.9	83.9	82.4

Table 7.4: Confusion matrix from Supratak et al. obtained from the cross-validation on the Fpz-Cz channel from the Sleep-EDF dataset

The poorest performance came from the stage N1, with the F1 less than 60, while the F1 for other stages were significantly better, with the range between 81.5 and 90.3. It is also important to notice that the confusion matrix is almost symmetric via the diagonal line (except for the pair of N2-N3), which indicates that the misclassifications were less likely to be due to the imbalance-class problem

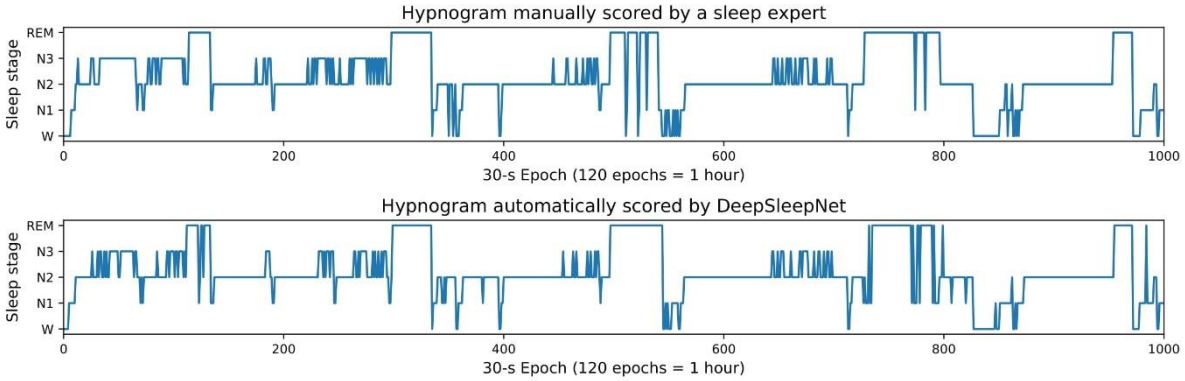


Figure 7.5: Examples from Supratak et al. of the hypnogram manually scored by a sleep expert (top) and the hypnogram automatically scored by DeepSleepNet (bottom) for Subject-1 from the MASS dataset.

The figure 7.5 presents an example of a manually scored hypnograms done by a sleep expert against an automatically scored one by our DeepSleepNet model for Subject-1 from the MASS dataset.

7.4.8 Comparison with state-of-the-art approaches

Methods	Dataset	EEG Channel	Overall Metrics			Per-class F1-Score (F1)				
			ACC	MF1	κ	W	N1	N2	N3	REM
Non-independent Training and Test Sets										
Ref. [HYWH13]	Sleep-EDF	Fpz-Cz	90.3	76.5	-	77.3	46.5	**94.9**	72.2	**91.8**
Ref. [SPU17]	Sleep-EDF	Pz-Oz	**91.3**	77	**0.86**	**97.8**	30.4	89	**85.5**	82.5
Ref. [HS17]	Sleep-EDF	Pz-Oz	90.8	**80**	0.85	96.9	**49.1**	89	84.2	81.2
Independent Training and Test Sets										
Ref. [TMG16]	Sleep-EDF	Fpz-Cz	78.9	73.7	-	71.6	**47.0**	84.6	84.0	81.4
Ref. [TMGZ16]	Sleep-EDF	Fpz-Cz	74.8	69.8	-	65.4	43.7	80.6	**84.9**	74.5
DeepSleepNet	Sleep-EDF	Fpz-Cz	**82.0**	**76.9**	**0.76**	84.7	46.6	**85.9**	84.8	**82.4**
DeepSleepNet	Sleep-EDF	Pz-Oz	79.8	73.1	0.72	**88.1**	37	82.7	77.3	80.3
Ref. [DSP+16]	MASS	F4-EOG	85.9	80.5	-	84.6	56.3	**90.7**	**84.8**	86.1
DeepSleepNet	MASS	F4-EOG	**86.2**	**81.7**	**0.80**	**87.3**	**59.8**	90.3	81.5	**89.3**

Table 7.5: Comparison from Supratak et al. between DeepSleepNet and other sleep stage scoring methods that utilizes hand-engineering features across overall accuracy (ACC), macro-F1 score (MF1), Cohen's Kappa, and Per-class F1-Score (F1)

Table 7.5 presents a comparison between our method and the state-of-the-art sleep stage scoring methods from the literature across ACC, MF1, kappa and F1. These methods include

the ones that utilize hand-engineered features[39] [40] [41], CNNs only[42] or LSTMs only[43]. The other methods' metrics were computed using the confusion matrices reported in their papers. The methods have been classified in two groups: *non-independent* and *independent* training and test sets.

The non-independent methods included parts of the test subjects' epochs in the training data, while the independent ones excluded all epochs of the test subjects from the training data. Non-independent methods are usually bad practice as it has been demonstrated multiple times in the literature to be prone to overfitting while it has been shown that the non-independent scheme resulted in an improvement of the performance[44]. Thus, we only compared the performances of our method with the non-independent group and the numbers in bold indicate the highest performance metrics of all methods in each dataset for each group.

Similar performances between our model and to the state-of-the-art methods have been achieved when using the same EEG channel and dataset. However, that performance has been achieve without compromising on the performances on the stage N1, which is the most difficult sleep stage to classify. This fact highlights that our method was not biased toward the majority of the sleep stages to the detriment of the minority ones. The kappa coefficient showed that the agreement between the sleep experts and our model were meaningful (between 0.61 and 0.80)[45]. Interestingly, our model performed better when applied on the Fpz-Cz channel compared to the Pz-Oz, which is similar to Tsinalis et al. [46].

7.4.9 Sequence Residual Learning

	Predicted					Per-class Metrics		
	W	N1	N2	N3	REM	PR	RE	F1
W	**5215**	709	94	19	190	84.5	83.7	84.1
N1	468	**2582**	747	11	916	40.8	54.7	46.8
N2	241	1846	**24140**	2435	872	93.4	81.7	87.2
N3	19	3	472	**7156**	1	74.3	93.5	82.8
REM	227	1181	383	5	**8668**	81.4	82.8	82.1

Table 7.6: Confusion matrix from Supratak et al obtained from 31-fold cross-validation on the F4-EOG channel from the MASS dataset using DeepSleepNet without Sequence Residual Learning

In order to assert the important of the sequence residual learning part, we ran a 31-fold cross-validation on the F4-EOG channel of the MASS dataset without the sequence residual learning part (e.g. the pre model in Algorithm 1) in DeepSleepNet. Table 6 shows a confusion matrix obtained from the cross-validation and it can be observed that the F1 of all sleep stages, except the stage N3, were lower than the ones in Table 7.6. The root cause for this is an increase in the misclassifications between the pairs of N1-N2, N2-N3 and N1-REM. We assume that it may be caused by the effects of oversampling the training set to have balanced-class samples which resulted in the model tending to predict more of stages N1 and N3. From those results, we can conclude that the process to stack the pre-trained representation learning part with the sequence residual learning part, and then fine-tune both parts with sequential training set helped improve the classification performance.

7.4.10 Model Analysis

Now, we will attempt to better understand the underlying structure of the model by analyzing and comparing: 1) the learned filters in the first convolutional layers of the two CNNs in the representation learning part; and 2) the memory cells inside the bidirectional-LSTMs in the sequence residual learning part.

The MASS dataset was used for the analysis with all of the 31 cross-validation folds.
We first tried to determine which filters were mostly active for each sleep stage (in the first convolutional layers of the two CNNs) by computing the average of the sum of the activations of all filters across samples of each sleep stage. We name $\{X_1, ..., X_n\}$ the N 30-s EEG epochs from each validation fold and we fed them to our model to obtain activations Z from the first convolutional layer of each CNN: $\{Z_1, ..., Z_n\}$, where Z_i in R^{pxq} and p and q are the activation output size and the number of filters of the first convolutional layer.

The average of the sum of the activations of the filter k for the sleep stage c is computed as follows:

$$u_{c,k} = \frac{\sum_{i=1}^{N_{y_{pred}=c}} \sum_{j=1}^{q} z_{i,j,k}}{N_{y_{pred}=c}}$$

where $u_{c,k}$ is the average of the sum of the activation of the filter k for sleep stage c, $Z_{i,j,k}$ is the j-th index of the activation vector Z of the filter k, and N is the number of EEG epochs that our model predicted as stage c. After we computed the $u_{c,k}$ of all filters for sleep stage c, we rescaled them into a range of 0 and 1. We denote this scaled k-dimension vector u_c as *filter activations* for stage c. This process was reiterated for all sleep stages until we got the filter activations from all sleep stages. Once this was done, we stacked them together, and rearranged the order of the filters in a way that the filters that were most contributing for each sleep stage were grouped together.

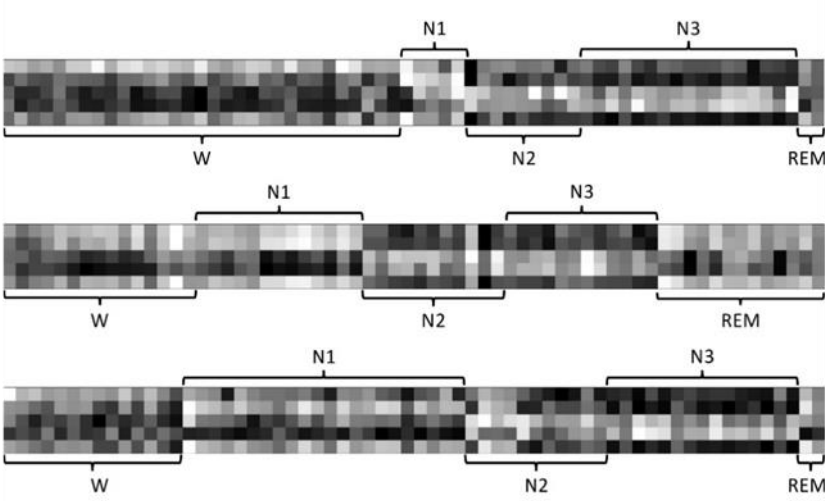


Figure 7.6: Examples from Supratak et al of the filter activations from the first convolutional layers of the two CNNs obtained by feeding our model with data from 3 subjects.

The filter activations from the small filters are on the left (a), and the larger filters are on the right (b). Each image has 5 rows and 64 columns, corresponding to 5 sleep stages and 64 filters respectively. Each pixel represents the scaled value of $u_{c,k}$ from (6.9), where 1 (e.g. active) is white and 0 (e.g. inactive) is black. Each row corresponds to the 64-dimension vector (e.g. k is 64) for each sleep stage c. The first row is from stage W and the last row is from stage REM. Each image also has labels indicating which filters are mostly active for which sleep stages.

Figure 7.6 shows an example of the filters activations with data from 3 subjects and from the small (a) and large (b) filters obtained by feeding our model. The five rows in the images correspond to the five sleep stages while the 64 columns correspond to the 64 filters. Each pixel represents the value of $u_{c,k}$ from equation 2 scaled into a range of 0 and 1, where 1 is active and white and 0 is inactive and black. Each row corresponds to the 64-dimension vector (e.g. k is 64) for each sleep stage c. The first type of filter that appeared in this analysis were the ones that were mostly active for each sleep stage. For example, some of the small and large filters were mostly active for both sleep stages N2 and N3. The second type of filters that appeared was the ones that were mostly active for multiple sleep stages. Even though a global trend appears across subjects for the types of filters, we found that the number of active filters for different sleep stages varied across subjects and, for a small number of subjects, no small filter was active for stage N1. The latter could be linked to the fact that there were only a few stage N1 in the dataset.

Then, in a second time, we analyzed how the bidirectional-LSTMs were used in our model to extract the temporal information from a sequence of EEG epochs. In order to do so, we looked at how the bidirectional-LSTMs managed their memory cells (e.g. c in (6.4) and (6.5)) using the visualization technique from Karpathy et al.[47]. It has been discovered that several memory cells of the forward LSTMs that were interpretable. Several cells were keeping track of the

wakefulness (stage W) or the sleep onset of the patient (stage N1), which would result in setting their values to active (e.g. positive values). Those cells would then become inactive (e.g. negative values) during sleep stages (stages N2, N3 and REM).

Figure 7.7 illustrates the changes of a LSTM cell value according to a sequence of sleep stages predicted by our model. Other interpretable cells have been identified, such as the ones that started with a high value at the beginning of each subject data and then slowly decreased with each sleep stage until the end of the subject data, or the ones that only activated when a continuous sequence of stages N3 and REM appeared. The capacity to correctly identify the next sleep stages relies on the current status of each subject and stage transition rules[48]. The existence of these cells showed that the LSTMs inside the sequence residual learning part indeed learned to do those tasks accurately.

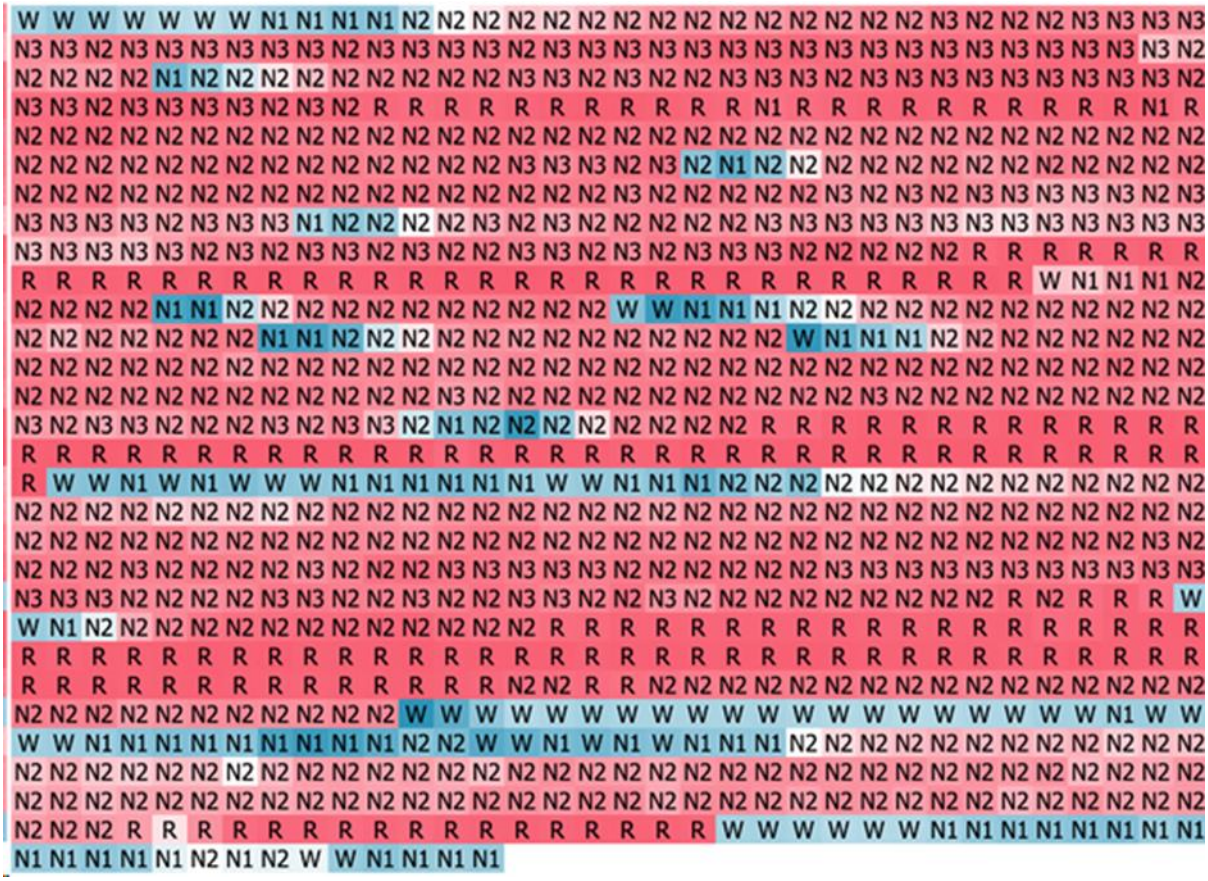


Figure 7.7: An example from Supratak et al of the LSTM cell that is active at the beginning of wakefulness (e.g. stage W) or the sleep onset (e.g. stage N1). The sequences of sleep stages are the predictions from DeepSleepNet on one subject data, arranged through time from left-to-right and top-to-bottom. The background color of each stage corresponds to tanh(c), where +1 is blue and -1 is red.

7.4.11 Conclusion

The results demonstrated that the model could flexibly be applied on different EEG channels (F4-EOG, Fpz-Cz and Pz-Oz) without any change both in the model architecture and the training algorithm. Also, the model achieved similar overall accuracy and macro F1-score compared to the state-of-the-art hand-engineered methods on both the MASS and Sleep-EDF datasets despite having different properties such as sampling rate and scoring standards (AASM and R&K). It is interesting to note that temporal information learned from the sequence residual learning part helped improve the classification performance. We can conclude that our proposed model was capable to automatically learn features for sleep stage scoring from different raw single-channel EEGs. This work has moved us one step closer to the possibility of remote sleep monitoring from home environments which would be less costly, less stressful for the patients and at a larger scale than current hospital setups. Remote monitoring could potentially help elder people and people with stress or sleep disorders on a daily basis and doctors to easily follow up on their patients.

Conversely, the eAE has benefited as well from that close collaboration with the DeepSleepNet project. Firstly, the researchers have provided valuable feedback on the user experience side of the first implementation of the eAE. That feedback has been included in the design of the second version bringing more value to the users. Secondly, this project acted as a testbed for validating the architecture and identify shortcomings of the implementation which have been addressed in the second version (adopted by OPAL). As this use case illustrates, all those innovations have opened the way for better science and deep learning to build better applications.

Chapter 7 References:

[1] Marx V. 2013. Drilling into big cancer-genome data. Nature Methods. PMID: 23538863

[2] Costa FF. 2014. Big data in biomedicine. Drug Discov. Today. PMID: 24183925

[3] A. Oehmichen, S. Jain, A. Gadotti and Y. d. Montjoye, "OPAL: High performance platform for large-scale privacy-preserving location data analytics," 2019 IEEE International Conference on Big Data (Big Data), doi: 10.1109/BigData47090.2019.9006389.

[4]A. Oehmichen, F. Guitton, P. Agapow, I. Emam and Y. Guo, "A Multi Tenant Computational Platform for Translational Medicine," 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), doi: 10.1109/ICDCS.2018.00167.

[5] Mosteller F. 1948. A k-Sample Slippage Test for an Extreme Population. The Annals of Mathematical Statistics. ISBN: 978-0-387-20271-6

[6] Kennedy RD et al. 2011. Development and independent validation of a prognostic assay for stage ii colon cancer using formalin-fixed paraffin-embedded tissue. Journal of Clinical Oncology. PMID: 22067406

[7] Guyon I et al. 2002. Gene Selection for Cancer Classiffication using Support Vector Machines. Machine Learning, Vol. 46.

[8] Cao D et al. 2009. Expression of HIF-1 alpha and VEGF in colorectal cancer: association with clinical outcomes and prognostic implications. BMC Cancer. PMID: 20003271

[9] Virtanen A et al. 1999. Estimation of prostate cancer probability by logistic regression: Free and total prostate-specific antigen, digital rectal examination, and heredity are significant variables. Clinical Chemistry.

[10] Fisher RA. 1935. The Logic of Inductive Inference. Journal of the Royal Statistical Society.

[11]Supratak A et al. 2017. DeepSleepNet: a Model for Automatic Sleep Stage Scoring based on Raw Single-Channel EEG. ieee transactions on neural systems and rehabilitation engineering, doi: 10.1109/TNSRE.2017.2721116

[12] Wulff K, et al. 2010. Sleep and circadian rhythm disruption in psychiatric and neurodegenerative disease. Nature Reviews Neuroscience, PMID: 20631712

[13] Hobson JA. 1969. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. Electroencephalography and Clinical Neurophysiology.

[14] Iber C et. al. 2007. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. American Academy of Sleep Medicine.

[15]Ioffe S, Szegedy C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

[16] He K et al. 2016. Deep Residual Learning for Image Recognition. In Cvpr. doi: 10.1109/CVPR.2016.90

[17]Schuster M, Paliwal KK. 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process.

[18] Iber C et. al. 2007. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. American Academy of Sleep Medicine.

[19] Hochreiter S, Schmidhuber J. 1997. Long Short-Term Memory. Neural Computation.

[20] Schuster M, Paliwal KK. 1997. Bidirectional recurrent neural networks. IEEE Trans. Signal Process.

[21] F.A. Gers FA, Schmidhuber J. 2000. Recurrent nets that time and count. In Proc. IJCNN'2000, Int. Joint Conf. on Neural Networks, Vol. 3.

[22] Sak H et al. 2014. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition.

[23] Cohen MX. Analyzing Neural Time Series Data: Theory and Practice. Cambridge (MA). MIT Press, 2014. ISBN: 978-0262019873

[24] Szegedy C et al. 2015. Going deeper with convolutions. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[25]Kingma D, Ba J. 2014. Adam: A method for stochastic optimization.

[26] Pascanu R et al. 2012. On the difficulty of training Recurrent Neural Networks.

[27] Srivastava N et al. 2014. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. J. of Machine Learning Research. doi: 15:1929–1958

[28] Zaremba W et al. 2014. Recurrent Neural Network Regularization.

[29] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio. 2012. On the difficulty of training Recurrent Neural Networks.

[30]Christian O'Reilly, Nadia Gosselin, Julie Carrier, Tore Nielsen. 2014. Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research. J. of Sleep Research.

[31] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, H. Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation.

[32] Bastiaan Kemp, Aeilko H. Zwinderman, Bert Tuk, Hilbert A C Kamphuisen, Josefien J L Obery´e. 2000. Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG. IEEE Trans. Biomed. Eng.

[33] Iber C, Ancoli-Israel S, Chesson AL Jr., et. al. 2007. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. In AASM Manual for Scoring Sleep.

[34] T. D. Lagerlund. 2000. Manipulating the magic of digital EEG: Montage reformatting and filtering. Amer. J. of Electroneurodiagnostic Tech.

[35] J. Allan Hobson. 1969. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. Electroencephalography and Clinical Neurophysiology.

[36] Iber C, Ancoli-Israel S, Chesson AL Jr, et. al. 2007. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. AASM.

[37]Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement.

[38] Marina Sokolova, Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. Inform Process. and Manage.

[39] Yu-Liang Hsu, Ya-Ting Yang, Jeen-Shing Wang, Chung-Yao Hsu. 2013. Automatic sleep stage recurrent neural classifier using energy features of EEG signals. Neurocomputing.

[40] Rajeev Sharma, Ram Bilas Pachori, Abhay Upadhyay. 2017. Automatic sleep stages classification based on iterative filtering of electroencephalogram signals. Neural Computing and Applicat.

[41]Ahnaf Rashik Hassan, Abdulhamit Subasi. 2017. A decision support system for automated identification of sleep stages from single-channel EEG signals. Knowledge-Based Syst.

[42] Orestis Tsinalis, Paul M. Matthews, Yike Guo, Stefanos Zafeiriou. 2016. Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks.

[43] Hao Dong, Akara Supratak, Wei Pan, Chao Wu, Paul M. Matthews, Yike Guo. 2016. Mixed Neural Network Approach for Temporal Sleep Stage Classification.

[44]Orestis Tsinalis, Paul M. Matthews, Yike Guo. 2016. Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders. Ann. of Biomedical Eng.

[45] Ahnaf Rashik Hassan, Abdulhamit Subasi. 2017. A decision support system for automated identification of sleep stages from single-channel EEG signals. Knowledge-Based Syst.

[46] Orestis Tsinalis, Paul M. Matthews, Yike Guo. 2016.Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders. Ann. of Biomedical Eng.

[47] Andrej Karpathy, Justin Johnson, Li Fei-Fei. 2015. Visualizing and Understanding Recurrent Networks.

[48] Iber C, Ancoli-Israel S, Chesson AL Jr, et. al. 2007. The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications. AASM.

Chapter 8: Select Case Studies

8.1 eTRIKS-Associated Case Studies

The DeepSleepNet case study, detailed in Chapter 7, is an exceptional example of the application of high-performance computation and analytic methodologies to address human health disorders. This chapter will detail two additional case studies in which eTRIKS colleagues, collaborating with client researchers, applied products, services and expertise described in earlier chapters to real-world translational science projects. Both case studies are peer-reviewed articles reprinted (for convenience to the reader) and cited in adherence to the licenses applied to each.

Readers seeking additional case studies may find the following articles of interest which represent work undertaken by three leading translational research informatics teams, eTRIKS, the Avillach Laboratory at Harvard University and the Netherlands-based TraIT (Translational IT) consortium.

1. Sijin He, May Yong, Paul M Matthews, Yike Guo, tranSMART-XNAT Connector tranSMART-XNAT connector—image selection based on clinical phenotypes and genetic profiles, Bioinformatics, Volume 33, Issue 5, 1 March 2017, Pages 787–788, https://doi.org/10.1093/bioinformatics/btw714
2. Murphy, S. N., Avillach, P., Bellazzi, R., Phillips, L., Gabetta, M., Eran, A., McDuffie, M. T., & Kohane, I. S. (2017). Combining clinical and genomics queries using i2b2 - Three methods. PloS one, 12(4), e0172187. https://doi.org/10.1371/journal.pone.0172187
3. Zhang C, Bijlard J, Staiger C, et al. Systematically linking tranSMART, Galaxy and EGA for reusing human translational research data. F1000research. 2017 ;6. DOI: 10.12688/f1000research.12168.1.
4. Venkata Satagopam, Wei Gu, Serge Eifes, Piotr Gawron, Marek Ostaszewski, Stephan Gebel, Adriano Barbosa-Silva, Rudi Balling, and Reinhard Schneider.Big Data.Jun 2016.97-108.http://doi.org/10.1089/big.2015.0057

8.2 Data and Knowledge Management in Translational Research: Implementation of the eTRIKS Platform for the IMI OncoTrack Consortium

Wei Gu, Reha Yildirimman, Emmanuel Van der Stuyft, Denny Verbeeck, Sascha Herzinger, Venkata Satagopam, Adriano Barbosa-Silva, Reinhard Schneider, Bodo Lange, Hans Lehrach, Yike Guo, David Henderson, and Anthony Rowe

Gu, W., Yildirimman, R., Van der Stuyft, E. et al. Data and knowledge management in translational research: implementation of the eTRIKS platform for the IMI OncoTrack consortium. BMC Bioinformatics 20, 164 (2019). https://doi.org/10.1186/s12859-019-2748-y

8.2.1 Background

The data coordination activities of large multi-stakeholder research collaborations are becoming more complex. In- creasingly, projects are citing the use of specialist know- ledge management technologies such as the tranSMART platform [1] as used by the IMI UBIOPRED, ABIRISK and OncoTrack projects [1 2 3 4]. In reality, however, a

The motivation to improve such technologies is therefore twofold: Firstly, the system provides a single place where data from all partners participating in the project can be deposited, collated, linked and then published back to the whole consortium. Secondly, the data are not just made *available* in curated form, but are also made *accessible*. This is achieved by the use of flexible user interfaces, combined with analytical and visualization tools that can be used by all stakeholders in the consortium and not just those with the specialist data handling skills such as bioinformaticians and statisticians. A consortium that provides a data coordination capability accelerates the work of the specialist data scientist who can access the raw data from a single location for specialist analysis. If this data coordination capability additionally includes a knowledge management technology, this can empower the wider community of scientists who are able to browse and generate hypotheses from all of the data in an accessible format.

In this paper, we present the broad overall systems architecture developed by the eTRIKS consortium to accommodate the data management requirements of translational research consortia, using the IMI OncoTrack project as a use case. Additionally, we present a novel plug-in for tranSMART developed by the IMI eTRIKS consortium to overcome some of the limitations in cross-linking related datasets, such as those found when exploring and conducting correlation analyses using clinical data, experimental data from patient derived ex vivo models and high dimensional “omics” data. The data linking solution presented here is capable of handling and integrating the majority of data types encountered in translational medicine research, independent of the medical indication, and should therefore be generally useful for other consortia faced with similar data management challenges.

In line with the challenges and requirements mentioned above, this knowledge management platform intends to provide a common point to access and share the accumulated, curated and pre-processed data sets as well as testing hypotheses and facilitating exchange of ideas.

The intended users and usages are:

1. 1) All “end-users” that do not necessarily have advanced IT skills to be able to explore the integrated datasets with dynamic visual-analytics to test new hypotheses immediately, without asking bioinformaticians for every (explorative) analysis.

2. 2) Bioinformaticians to select and download data (curated or raw) for specific analyses.

3. 3) Data managers as well as researchers to collect, organise, store and disseminate data during the course of the project.

4. 4) Project managers to oversee project progress in terms of available data and metadata.

We would like to emphasis that the analytical tools provided on the platform are not meant to replace all advanced analyses that might be carried out by trained bioinformaticians and biostatisticians, who nevertheless can benefit from the reduced time and effort needed for data preparation.

8.2.2 Implementation: The IMI OncoTrack consortium

The IMI OncoTrack Consortium[2] is an ambitious international consortium that is focused on advancing “Methods for systematic next generation oncology biomarker development”. As one of the Innovative Medicines Initiative (IMI) oncology projects, it brings together academic and industry scientists from more than twenty partner institutions in a research project to develop and assess novel approaches for identification of new markers for the treatment response of colon cancer.

At the core of OncoTrack are two patient cohorts that, either prospectively at the point of primary colon cancer surgery or retrospectively at the point of metastasis surgery are sampled in order to build a colon cancer tissue bank containing both primary and metastatic tumour samples, together with associated normal tissues and biofluids. A part of each tissue sample is also used to develop in vitro 3D cell cultures and in vivo xenograft models that are used to study response to standard and experimental therapies.

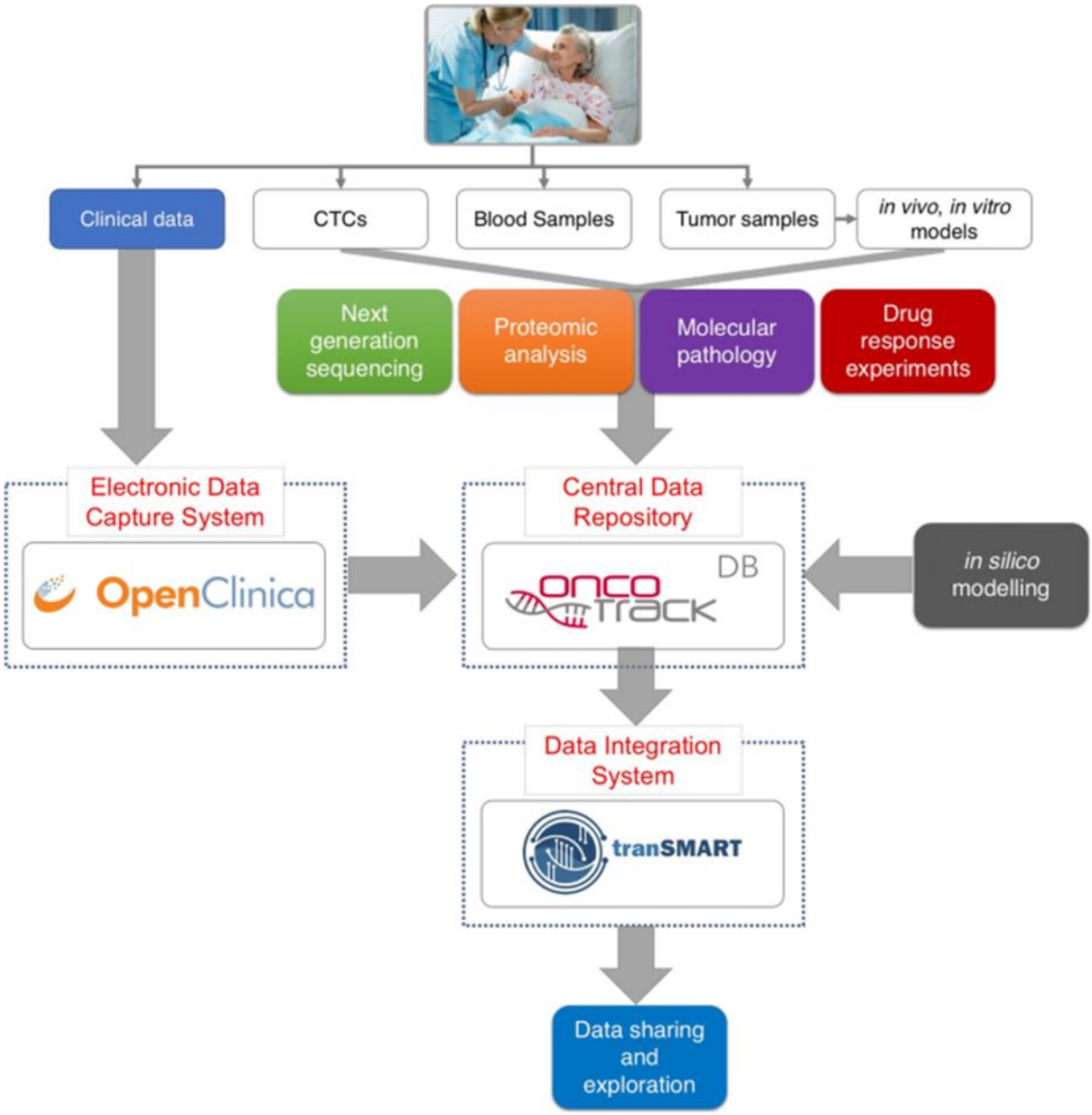
The tissue samples are processed to build collections of DNA, RNA, serum and circulating tumour cells that are then analysed to generate an in-depth description of the genome, transcriptome, methylome and proteome both of the tumour and the biological models. This approach uses a broad panel of methods such as next generation sequencing, proximity extension assays, reverse phase protein arrays, methylation arrays and mass spectrometry. The

patient-derived models also provide platforms to study the role of tumour progenitor or 'cancer stem cells' in the pathogenesis and evolution of colon cancers.

Finally, data from all of these platforms are combined using a systems biology approach that can be used to make personalised predictions about how an individual may respond to therapy. The systems biology model of the cancer cell incorporates the combined results of genome, transcriptome, methylome and proteome analyses [5].

The coordination of these different collections of data requires core systems to be used to perform the data collection and integration tasks. We would like to note that the "data integration" related to the work reported here are the steps and procedures to transform and store data from subject level, sample level and derived animal models as well as across different data types (drug response, different molecular and 'omics data) in an interlinked manner in a data warehouse. In this way users are able to filter data in any layer/type and query related data in the same or different layer/type with a few mouse clicks and subsequently test their new hypotheses. As shown in Fig. 1 and detailed below, the OncoTrack data management work package implemented OpenClinica[6] and developed the OncoTrack DB[7] as central repositories for clinical and biological data, respectively. Here, we describe the collaborative effort to interface these data repositories with tranSMART, to provide an interactive user interface for exploration and preliminary data analysis.

Fig.1 OpenClinica: electronic data capture



The components of the OncoTrack data coordination operation. The platform comprises three major components: the Electronic Data Capture System (EDC, OpenClinica), the Central Data Repository (OncoTrack DB), and the Data Integration System (tranSMART). The OpenClinica EDC system is used to collect medical history and observational patient data from clinical sites during the studies and feeds the structured data to the Central Data Repository. The Central Data Repository, OncoTrack DB is a sample indexed content management system. Data and results generated in the laboratories (before integration) are deposited and exchanged here. In order to link the different data types and layers, the data collected in the OncoTrack DB are integrated in the Data Integration System, tranSMART. The tranSMART data warehouse provides deep linking and integration between the clinical and laboratory data and a set of tools for the exploratory analysis of the integrated data

The first component of the data coordination platform is the OpenClinica Electronic Data Capture system (EDC, https://www.openclinica.com/; https://github.com/OpenClinica/OpenClinica).
OpenClinica provides the capability for the clinical sites to record electronically all of the patient data from different visits and to deposit these in a central database. The system enables the design of specific data entry conventions and data validation checks. These features ensure high data quality by providing all clinical sites with identical case report forms and by flagging data entry errors so they can be rapidly fixed. The user interface is made available through a standard web browser technology so that it requires no installation of software, allowing it to be readily adopted by all clinical sites. In order to ensure data privacy and compliance with data protection legislation, access to OpenClinica is IP-restricted and each clinical site can access only to the data for their own patients. In compliance with the institutional ethics committee and patient data privacy regulations, only a subset of the clinical data is made available to all consortium scientists through OncoTrack DB.

OncoTrack DB: sample indexed content management

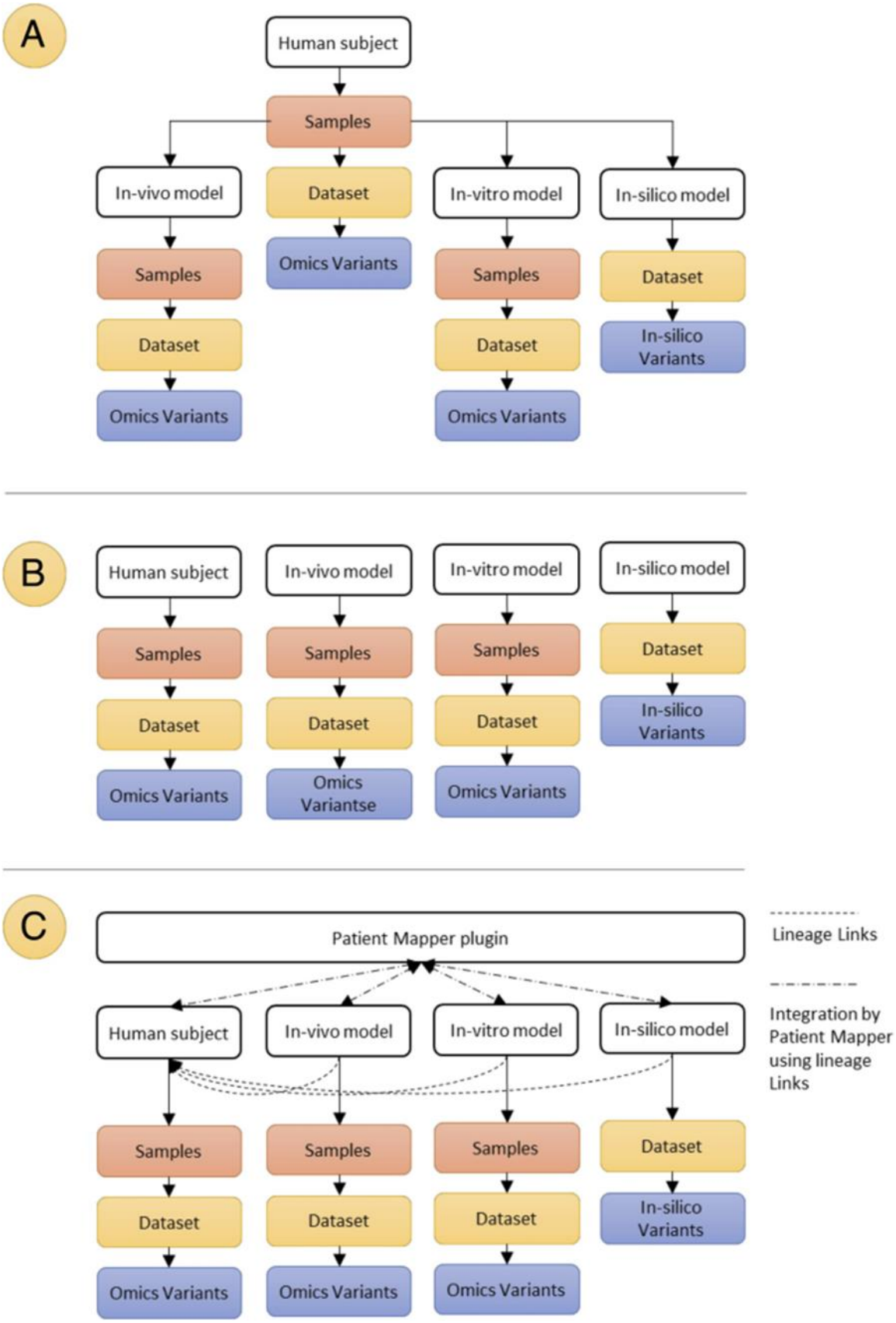
The Oncotrack DB is software based on DIPSBC (data integration platform for systems biology collaborations), further developed by Alacris Theranostics and adapted to the specific needs of the OncoTrack project[8]. It is best described as a "Sample Indexed" Content Management System (CMS). It supports the typical features of a CMS to store, version control and manage collections of files and also enables project management, dissemination and progress tracking as well as allowing multiple channels for data access (eg. web interface, RESTful API). File formats were developed to store the results of the different laboratory analyses including the NGS based genome and transcriptome analysis, the ex vivo drug response experiments and the molecular characterisation of tumour samples. For each experimental data type, a unique upload interface was deployed to handle specific requirements with regard to data production frequency, volume and format as well as transfer method (i.e. web interface, RESTful API). Additionally, the OncoTrack DB indexes each of these data files with unique sample identifiers, so that each file can easily be filtered to locate and sort all data by cohort, experimental platform or patient. Throughout this work, we have adopted generally accepted data standards for 'omics, clinical data etc. where applicable, inter alia CDISC compliant terminology for clinical data using Study Data Tabulation Model (SDTM), high-throughput sequencing data standards (e.g. FASTQ, BAM), gene sequence variations data format (VCF) or Systems Biology Markup Language (SBML) for computational models. In addition, data was loaded into a relational database and mapped to respective reference standards (e.g. Ensembl, UniProt, miRBase) to allow comparability and ensure compatibility. This allowed for more advanced data access and querying of available data sets.

tranSMART: knowledge management data warehouse

To make the data collected in OpenClinica and the OncoTrack DB accessible to the entire consortium in a systematic way, the tranSMART knowledge management platform was used. tranSMART is an open-source data warehouse designed to store data from clinical trials, as well as data from pre-clinical research, so that these can be interrogated together in translational research projects. tranSMART is a web-based system, designed for use by multiple users, across organizations. Prior to uploading data into tranSMART, a curation step (to adapt formats and define the data tree) needs to be performed. The data pre-processing is handled during this curation phase and ensures that the end-user is presented with data sets upon which valid hypotheses can be based. To ensure data integrity, it is recommended that the pre-processing and uploading be restricted to a limited group of data curators, working with uniform ETL scripts (https://github.com/transmart/tranSMART-ETL).

The data were organised in 3 core collections: 1) the observational clinical cohorts, 2) the drug response data from the cell-line models and 3) the drug response data from the xenograft models (see Fig. 2). The high dimensional data from the molecular analyses were linked to these collections so that users could browse and analyse:
Variants among germline, primary and metastatic tumour material
Confirmatory genomic analyses of xenograft and cell cultures
Quantification of RNA transcripts from clinical and preclinical samples
Quantification of small non-coding RNA (miRNA)
Analysis of DNA Methylation

Fig. 2


A
Human subject
Samples
In-vivo model
Dataset
In-vitro model
In-silico model
Samples
Omics Variants
Samples
Dataset
Dataset
In-silico Variants
Omics Variants
Omics Variants
B
Human subject
In-vivo model
In-vitro model
In-silico model
Samples
Samples
Samples
Dataset
Dataset
Dataset
Dataset
In-silico Variants
Omics Variants
Omics Variantse
Omics Variants
C
Patient Mapper plugin
Lineage Links
Integration by Patient Mapper using lineage Links
Human subject
In-vivo model
In-vitro model
In-silico model
Samples
Samples
Samples
Dataset
Dataset
Dataset
Dataset
In-silico Variants
Omics Variants
Omics Variants
Omics Variants

The OncoTrack dataset structure. **a** The complex OncoTrack data hierarchy with OMICS datasets directly generated from patient material and datasets generated from patient derived pre-clinical in vivo, in vitro and in silico models. **b** Due to constraints in tranSMART (v16.1) unable to represent this hierarchical use of samples, data has been organised as a series of different independent collections. One collection for data derived directly from patient samples and other collections for data derived from the pre-clinical models. **c** A solution we provided with linkage back to human subject and a tool to automatically map data using this linkage

The implementations of the functions reported in this manuscript have been integrated into the tranSMART main release, starting with version 16.2 (https://wiki.transmartfoundation.org/pages/viewpage.action?pageId=10126184). The code can be accessed under:
https://github.com/transmart/transmartApp and https://github.com/transmart/SmartR
The documentation can be found at: https://transmart-app.readthedocs.io/en/latest/
A description of and link to a public demonstration version of the tranSMART instance can be found at https://wgu.pages.uni.lu/etriks-oncotrack/

Dynamic dataset linking

The Oncotrack consortium based its approach to biomarker discovery on the innovative experimental design of creating collections of patient derived pre-clinical models. Tumour tissue collected during surgery from both the primary and metastatic tumours was used to create in vitro 3D-cell line models and xenograft in vivo models that could be linked back to the original patient. Cell lines and xenografts were used to study the response to a standard panel of established and experimental colon cancer drugs. The combination of deep molecular characterization of the tumours and their associated models with data on drug response provides the scientist with the necessary information for identification of candidate biomarkers for prediction of response to treatment.

Data generated in the OncoTrack study is organised so that each sample can be linked back to the patient from whose tissue it was generated, as shown in Fig. 2a.

The primary data level is the human cohort, with the primary entity being the subject. Patient tissue samples collected from subjects are profiled using omics and NGS technologies creating datasets directly attributable to the subject. A second data level is generated from the three disease modelling platforms used by OncoTrack: xenograft based in vivo models, 3D cell line based in vitro models (‘biological models’) and cell simulation based in silico models. Each of these is used to explore the tumour samples in different experiments such as response to standard clinical or novel experimental therapies. The biological models are then profiled using NGS and omics analysis technology, generating their own dataset and variants. The primary entity of this data is the model used in the experiment (e.g. cell line) with a lineage to the original patient. This two level lineage hierarchy of the datasets is shown conceptually in Fig. 2a.

This approach contrasts with the data model of tranSMART that has (by design) been developed with constraints regarding data organization. These constraints are required in order to achieve the required interactions of a flexible data model to a suite of analysis tools. These constraints mean that when modelled in tranSMART the data has to be modelled as 4 independent data sets (Fig. 2b) or coerced to a structure resembling Fig. 2a but at the loss of being able to use the analysis and visualisation tools.

Our objective was to create a mechanism where 1) data sets could be analysed independently *and* 2) we were able to respect the lineage of the samples to enable integrated analysis between the different levels in the hierarchy in the dataset. Our solution, shown in Fig. 2c is to maintain the basic tranSMART structure shown in Fig. 2b, augmented with additional metadata about lineage, mapping all level two datasets to their "parent" in the cohort dataset.

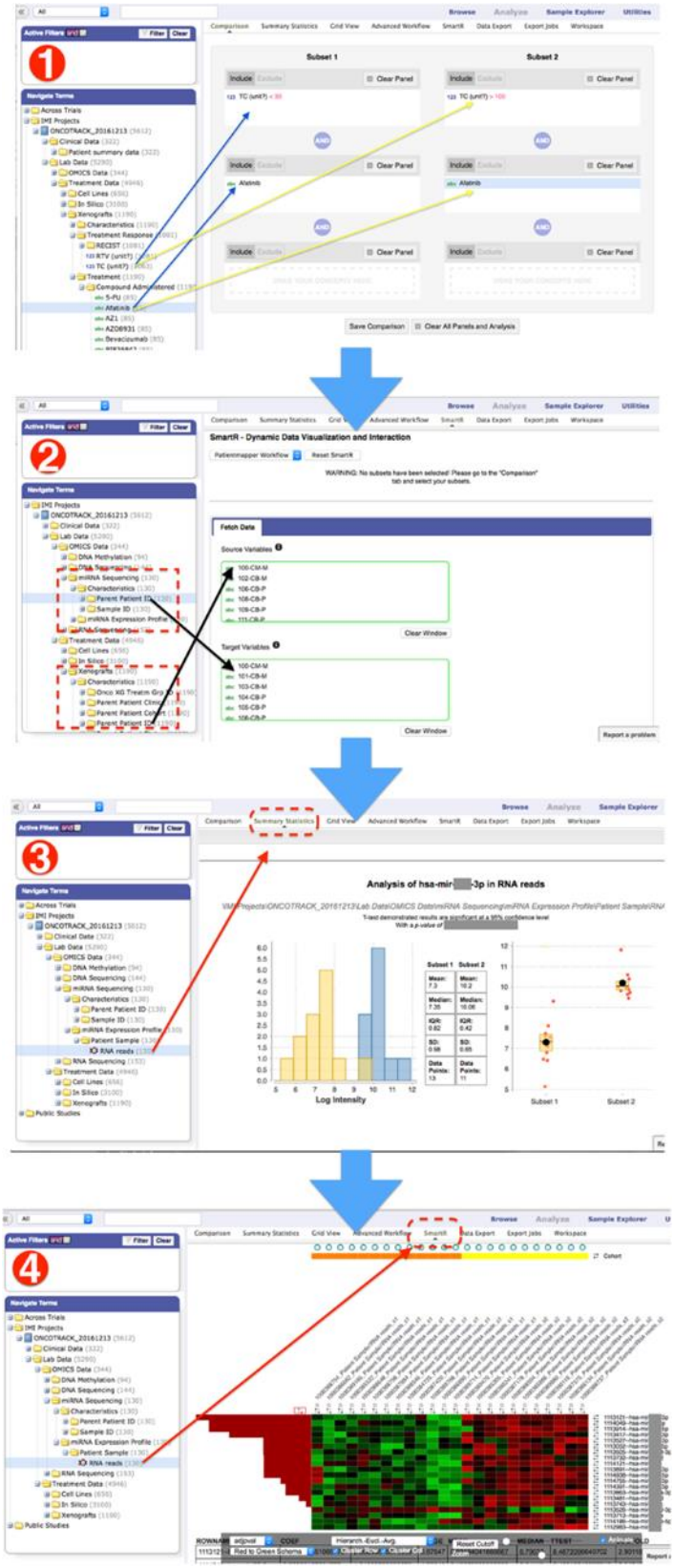
Additionally, we developed PatientMapper, a plugin-tool for tranSMART designed to integrate data sets from different levels of the hierarchy referring to these mapped lineage relationship metadata. When applied across datasets with the lineage mapping, Patient Mapper uses the back-links to correctly integrate and reshape the data to be compatible with the tranSMART analytics suite.

Data curation for dynamic data linking

To support dynamic data-linking among datasets, we developed an enhanced curation process to create a data model that includes lineage relationships between different entities. To achieve this, we developed a new mapping logic, in which the parent-child relationships are kept for all levels of datasets to the patient from which the samples/derived model are derived (see Fig. 2c). For example: a patient is a parent of *n* patient samples. Those samples can again be a parent of *m* in vitro models (like e.g. xenografts or xenograft treatment groups). Those in turn can be parents of *p* samples used for 'omics measurements, or even of 'child' in vitro models, etc.)

In tranSMART, variables are represented in a tree structure (i2b2 tree, see Fig. 3 and see also Additional file 1)[9]. The design of the data tree structure should organise the data to allow easy exploration of datasets. In line with the above considerations, in the OncoTrack-tranSMART integration, we separated different data levels and data types into separate study-trees to better organise the different categories (clinical data and lab data). Under the Clinical Data tree, general subject information (e.g. Clinical site, Cohort, etc.) of the participating subject are stored. The Lab Data stores data generated in the lab (e.g. Treatment Data, OMICS Data). In each subtree under the "Treatment Data" and the "OMICS Data", the subject/sample information as well as the interrelationships to other subtrees are organized in the "Characteristics", and the corresponding measured data are stored within the subtree labelled with the data type (e.g. Xenografts, DNA_Methylation, etc.)

Fig. 3



Integration of OncoTrack data into tranSMART: (1) Left panel: Overall data representation in the TranSMART data tree. Right panel: easy customized cohort building with drag-and-drop. (2) Cascaded querying with cohort linking/selection tool PatientMapper. (3) Generating

summary statistics of a miRNA of choice by dragging the miRNA-Seq node to the right panel and providing miRNA ID using the HiDome plugin. (4) Performing miRNA-ome wide heatmap analysis between the two sub-cohorts (here responder vs. non-responder for a selected drug treatment) using SmartR workflows.

Data curation and transformation are a prerequisite for the implementation of the data model described above. These steps are sometimes time consuming and require detailed knowledge regarding the necessary pre-processing of each data type as well as familiarity with tranSMART ETL requirements and scripting skills. Within the work reported in this paper, however, the curation need only be performed once and periodic updates (while new data of the same data type are generated) can be done automatically with pipelines developed during the manual curation. Data contributed by the different partners contributing to OncoTrack were collected centrally in OncoTrack DB. To avoid the risk of variability in the process, curation and transformation were performed centrally using one uniform set of ETL scripts. Details of each curation step are described in the Additional file 1.

Dynamic cross-layer data link tool (PatientMapper)

One typical query/analysis that requires the above-mentioned data model could be: what are the differences between xenograft models that respond to a certain drug and those that do not respond to the same drug: how do their parent samples differ in transcriptome and/or epigenome? To enable users to easily explore such a data model with dynamic cross-layer data, we have developed a user-friendly data linking tool (PatientMapper. see Fig. 3 (2)) that allows users to easily link sub-cohorts they have built on any level of data to datasets in other levels for the corresponding parent/children sample/subjects. This tool is integrated into tranSMART and updates cohort selection automatically based on the linking parameters selected by the user. From this point on, the other analysis and exploration of the updated cohorts can be performed within the same platform. This tool is not limited to mapping sample level data to patient level data but can be used to map data across any levels as long as they share a common lineage.

Results visualization

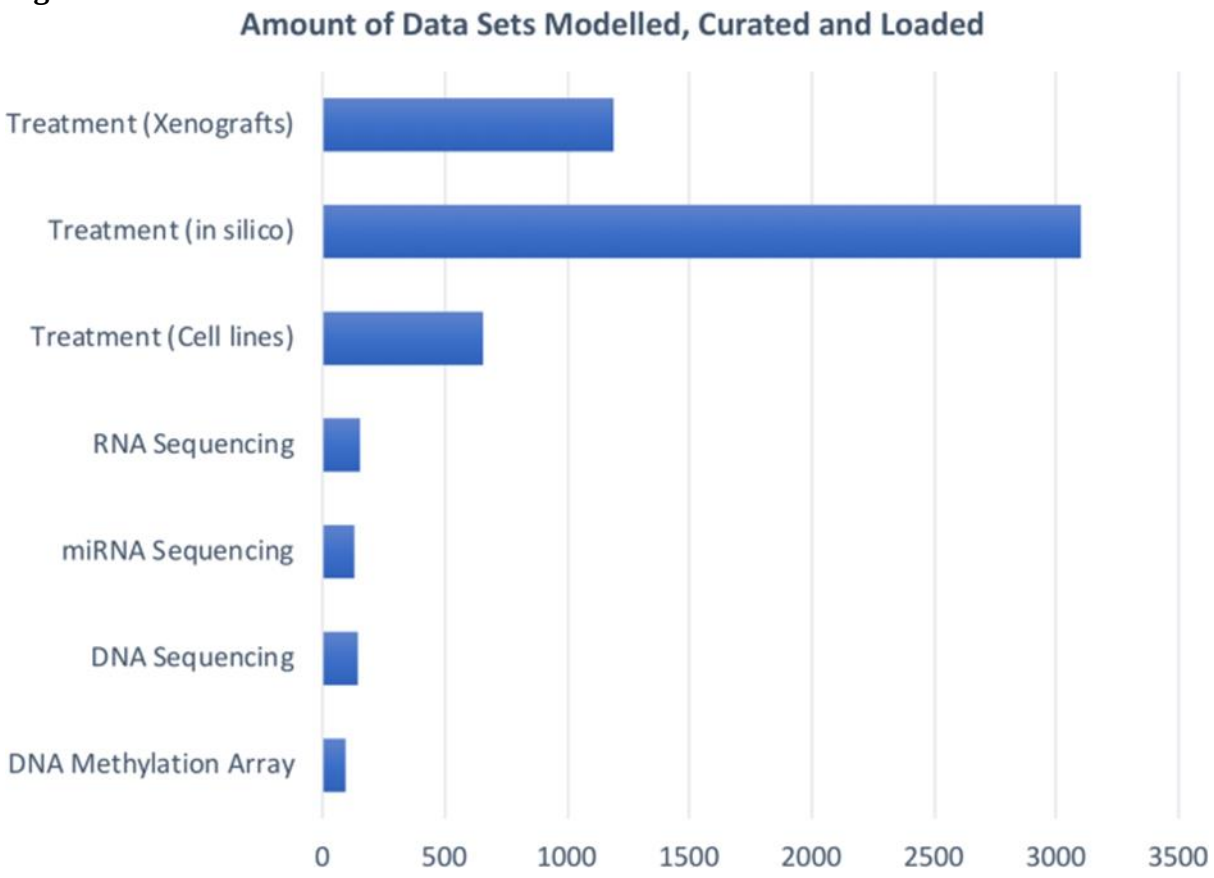
High Dimensional and Omics Exploration (HiDome) is a novel functionality for tranSMART that was developed through eTRIKS Labs[10]. It extends the platform's core capabilities with regard to handling omics data. HiDome allows the visualization of individual components of these data sets, for example the read count distribution for a given miRNA (see panel 3 in Fig. 3). It also enables creation of cohorts based on omics data set components, for instance comparing patients with a high versus a low read count for a specific miRNA. Details about the development of HiDome are described in a separate paper[11].

SmartR is another new functionality for tranSMART that was also developed through eTRIKS Labs [12]. This functional module enables the user of tranSMART to perform interactive visual analytics for translational research data, including both low-dimensional clinical/phenotypic data and high-dimensional OMICS data (see panel 4 in Fig. 3).

8.2.3 Results: Oncotrack TranSMART

The current Oncotrack TranSMART deployed to the consortium is based on the eTRIKS distribution (eTRIKS V3) of tranSMART 16.1. A summary of data that have been modelled, curated and loaded in the OncoTrack tranSMART server is shown in Fig. 4.

Fig. 4



An overview of OncoTrack data that have been modelled, curated and loaded in the OncoTrack tranSMART Server

Case study

To illustrate how the OncoTrack TranSMART can facilitate the exploration and analysis of data, we present here the use case already introduced in the discussion of the PatientMapper (see above). We would like to emphasise that this paper is not meant to focus on any specific scientific questions within the OncoTrack project, which have been reported in a separate paper [13], but rather to demonstrate the advantage of the tranSMART platform in solving data integration problems in general. For this reason, the marker annotations are blanked out.
The use case: For two xenograft groups, one whose tumours respond to treatment with Afatinib, the other one whose tumours are resistant, what biomarkers (e.g. miRNA) are different in their

parent patient tumor samples? And how to check whether a marker of interest is differentially presented?

The steps: Researchers who use the OncoTrack-tranSMART can achieve this goal easily by first building the two cohorts (xenografts Afatinib responders vs xenografts Afatinib non-responders) by dragging the Afatinib data-node and treatment response TC values (with filters, here < 30 and > 100) from the data tree into cohort selection (See Fig. 3 (1) for details). In order to get the miRNA data of the corresponding source patient, users can link the cohorts that were built using the xenograft level data to patient level data (here: miRNA sequencing data) using the GUI tool PatientMapper (Fig. 3 (2)) that will automatically handle the many-to-one relationship across the different data layers. In this example, the patient level miRNA expression profile (from miRNA-Seq) is linked to the xenograft level treatment response data by simply dragging-and-dropping their Parent Patient ID branch on the i2b2 tree to the PatientMapper tool.

With this new cohort after data mapping, researchers can easily check and visualize the corresponding miRNA sequencing data between the two sub-cohorts via the Summary Statistics function in tranSMART, by dragging the miRNA sequencing data node into it (See Fig. 3 (3)).

Researchers can extend the same steps to analyze the differences across the complete miRNA data set, using a few mouse-clicks to run the SmartR workflow (Fig. 3 (4)) to explore and identify differential biomarkers between the responders and non-responders. In all these steps, data mapping, linking and preparation are handled automatically by the OncoTrack-tranSMART integration platform. Therefore, researchers can focus directly on the scientific questions, without spending any effort on processing the data and data-integration, which is otherwise a burden and the most time-consuming part of translational research data analysis.

8.2.4 Discussion

Recent reviews have summarized many of the existing computing and analytical software packages designed to ease integrated analysis of 'omics and/or clinical data[14 15 16]. Those platforms are either repositories with an existing infrastructure or solutions requiring deployment. The advantage of the first type of solutions is their out-of-the-box usability, but this sacrifices the flexibility of configuration and toolset management. This type is represented by technologies like STRIDE[17], iDASH[18], caGRID and its follow up, TRIAD[19 20] or BDDS Center[21]. Many platforms in this category focus on a specific disease, like cBioPortal[22] or G-DOC[23 24] for cancer, or COPD Knowledge Base[25] for pulmonary dysfunction. The second family of solutions requires deployment on the user's infrastructure, often requiring substantial storage or High-Performance Computing (HPC) capabilities, but allows more flexibility in the setup and easier development. As a result of their configurable nature, such solutions provide support to ongoing projects as (part of) their data management platform to handle complex data. Examples in this group are BRISK[26], tranSMART[1] or Transmed[27]. Informative use cases

of such platforms are SHRINE[28]and DARiS[29], where well-defined demands of clinical research projects drove the design and implementation of infrastructure supporting translational medicine.

Besides these platforms, there are also many solutions that target web-based integrated analysis of 'omics data. Some well-known examples are EuPathDB (a eukaryotic pathogen genomics database resource[29]), the DNA Microarray Inter-omics Analysis Platform[30], Mayday SeaSight (combined analysis of deep sequencing and microarray data,[31]), GeneTrail2 (multi-omics enrichment analysis[32]), OmicsAnalyzer (a Cytoscape plug-in suite for modeling 'omics data[33]), PathVisioRPC (visualise and analyse data on pathways[34]), 3Omics (analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data[35]) and PaintOmics (joint visualization of transcriptomics and metabolomics data[36]).

Among the above-mentioned solutions, tranSMART stands out as a community-driven, rapidly growing, web-based data and visual-analytics platform for clinical and translational research [1 5]. TranSMART is being used by many (> 100) organizations and consortia around the world [2 3 4 5 15 37 38 39]. It enables the integrated storage of translational data (clinical and 'omics) by providing interlinks between different data-types and it allows researchers to interactively explore data as well as to develop, test and refine their hypotheses. These features are essential in order to support multi-party consortia like OncoTrack, that involve researchers with very diverse background working together on the datasets generated during the project. In the eTRIKS consortium, the platform has been further developed to incorporate more advanced, user-friendly and portable functionalities [39 40 41 42 43].

This paper describes the approach used by eTRIKS to provide an interface between the data architecture in the OncoTrack consortium and tranSMART. We also highlight the development of a new plug-in for the tranSMART platform to support dynamic data-linking among different datasets and datatypes in tranSMART.

The consortium model approach to research problems is becoming increasingly successful, as seen by the continuation of the European Innovative Medicines Initiative and the similar programs such as CPATH and the Accelerated Medicines Partnerships in the USA. There is increasing awareness among both funding agencies and the coordinators of large consortia, that data coordination and knowledge management capabilities are prerequisites for data to be integrated and used by all stakeholders in the collaboration and therefore constitute a key part of a project's operational design. Developing a strong data coordination capability enables:
Project Coordinators to understand the progress of data generation by different laboratories within the project, to help manage the scientific deliverables of a project and to identify in an early stage any data quality problems
Clinical and Laboratory scientists, as by interacting with a knowledge management platform they have access to all of the data from across the consortium, not just the sections they generated themselves

Data Scientists, Bioinformaticians and Statisticians to have access to clean, curated and linked datasets that represent the master version of data, saving them time in performing their own data preparation

While there are significant advantages to the investment in such a capability it should be recognised that there is no gold standard for data and knowledge management. As we have shown here, 3 key components (Open Clinica, OncoTrack DB, tranSMART) are used to collect, organise, publish and support analysis of the data generated in the OncoTrack consortium. While all of the software is Open Source and does not require a license for its implementation, there are operational costs in both the underlying IT hardware and the multi-disciplinary skill sets of people acting as data coordinator.

8.2.5 Conclusions

The authors suggest that results generated from exploratory analysis as described here provide a useful approach to hypothesis generation, but that such results should be scrutinized by a qualified statistician or bioinformatician prior to publication.

During the course of OncoTrack, we were confronted by the reality of the maxim "Scientific research and data production in life sciences move faster than development of the technical infrastructure". We developed patient derived pre-clinical models on a large scale and amassed large data sets from the analysis both of these models as well as the biological characteristics of the clinical samples. Consequently, new technology had to be developed to support the dynamic data linking across different datasets to enable the users to formulate the queries and analyses they wanted to explore. The approach described here is generally applicable to data collected in typical translational medicine research projects.

References

[1] Szalma S, Koka V, Khasanova T, Perakslis ED. 2010. Effective knowledge management in translational medicine. Brief Bioinform.

[2] Wheelock CE, Goss VM, Balgoma D, Nicholas B, Brandsma J, Skipp PJ, Snowden S, Burg D, D'Amico A, Horvath I, Chaiboonchoe A, Ahmed H, Ballereau S, Rossios C, Chung KF, Montuschi P, Fowler SJ, Adcock IM, Postle AD, Dahleń SE, Rowe A, Sterk PJ, Auffray C, Djukanović R. 2013. Application of 'omics technologies to biomarker discovery in inflammatory lung diseases. Eur Respir J.

[3] Henderson D, Ogilvie LA, Hoyle N, Keilholz U, Lange B, Lehrach H. 2014. Personalized medicine approaches for colon cancer driven by genomics and systems biology: OncoTrack. Biotechnol J.

[4] Bachelet D, Hässler S, Mbogning C, Link J, Ryner M, Ramanujam R, Auer M, Jensen PEH, et al. 2016. Occurrence of anti-drug antibodies against interferon-beta and natalizumab in multiple sclerosis: a collaborative cohort analysis. PLoS One.

[5] Link J, Ramanujam R, Auer M, Ryner M, Hässler S, Bachelet D, Mbogning C, Warnke C, et al. 2017. Clinical practice of analysis of anti-drug antibodies against interferon beta and natalizumab in multiple sclerosis patients in Europe: a descriptive study of test results. PLoS One.

[6] Wierling C, Kühn A, Hache H, Daskalaki A, Maschke-Dutz E, Peycheva S, Li J, Herwig R, Lehrach H. 2012. Prediction in the face of uncertainty: a Monte Carlo-based approach for systems biology of cancer treatment. Mutat Res Toxicol Environ Mutagen.

[7] www.openclinica.com. Copyright © OpenClinica LLC and collaborators, Waltham, MA, USA, The data collection and management for this paper was performed using the OpenClinica open source software, version 3.1.

[8] Dreher F, Kreitler T, Hardt C, Kamburov A, Yildirimman R, Schellander K, Lehrach H, Lange BMH, Herwig R. 2012. DIPSBC - data integration platform for systems biology collaborations. BMC Bioinformatics.

[9]Gainer V, Hackett K, Mendis M, Kuttan R, Pan W, Phillips LC, Chueh HC, Murphy S. 2007. Using the i2b2 hive for clinical discovery: an example. AMIA Annu Symp Proc.

[10] The eTRIKS Consortium, eTRIKS Labs. (available at https://www.etriks.org/etriks_labs/).

[11] Verbeeck D, Elefsinioti A. Hidome: Unlocking high dimensional data in TranSMART (manuscript in preparation).

[12] Herzinger S, Gu W, Satagopam V, Eifes S, Rege K, Barbosa-Silva A, Schneider R. 2017. SmartR: an open-source platform for interactive visual analytics for translational research data. Bioinform.

[13] Schütte M, Risch T, Abdavi-Azar N, Boehnke K, Schumacher D, Keil M, Yildiriman R, Jandrasits C, et al. 2017. Molecular dissection of colorectal cancer in pre-clinical models identifies biomarkers predicting sensitivity to EGFR inhibitors. Nat Commun.

[14] Canuel V, Rance B, Avillach P, Degoulet P, Burgun A. 2015. Translational research platforms integrating clinical and omics data: a review of publicly available solutions. Brief Bioinform.

[15] Zeng IS, Lumley T. 2018. Review of Statistical Learning Methods in Integrated Omics Studies (An Integrated Information Science). Bioinform Biol Insights.

[16] Dunn W Jr, Burgun A, Krebs MO, Rance B. 2017. Exploring and visualizing multidimensional data in translational research platforms. Brief Bioinform.

[17] Lowe HJ, Ferris TA, Hernandez Nd PM, Weber SC. 2009. STRIDE – an integrated standards-based translational research informatics platform. AMIA Annu Symp Proc.

[18]Ohno-Machado L, Bafna V, Boxwala AA, Chapman BE, Chapman WW, Chaudhuri K, Day ME, Farcas C, et al. 2012. iDASH: integrating data for analysis, anonymization, and sharing. J Am Med Informatics Assoc.

[19]Oster S, Langella S, Hastings S, Ervin D, Madduri R, Phillips J, Kurc T, Siebenlist F, Covitz P, Shanbhag K, Foster I, Saltz J. 2008. caGrid 1.0: An enterprise grid infrastructure for biomedical research. J Am Med Informatics Assoc.

[20] Payne P, Ervin D, Dhaval R, Borlawsky T, Lai A, Payne. 2001. PRO. TRIAD: the translational research informatics and data management grid. Appl Clin Inf.

[21] Toga AW, Foster I, Kesselman C, Madduri R, Chard K, Deutsch EW, Price ND, Glusman G, Heavner BD, Dinov ID, Ames J, Van Horn J, Kramer R, Hood L. 2015. Big biomedical data as the key resource for discovery science. J Am Med Informatics Assoc.

[22]Cerami E, Gao J, Dogrusoz U, Gross BE, Sumer SO, Aksoy BA, Jacobsen A, Byrne CJ, Heuer ML, Larsson E, Antipin Y, Reva B, Goldberg AP, Sander C, Schultz N. 2012. The cBio Cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. Cancer Discov.

[23] Madhavan S, Gauba R, Song L, Bhuvaneshwar K, Gusev Y, Byers S, Juhl H, Weiner L. 2013. AMIA Jt Summits Transl Sci Proc.

[24] Bhuvaneshwar K, Belouali A, Singh V, Johnson RM, Song L, Alaoui A, Harris MA, Clarke R, Weiner LM, Gusev Y, Madhavan S. 2016. G-DOC plus - an integrative bioinformatics platform for precision medicine. BMC Bioinformatics.

[25] Cano I, Tényi Á, Schueller C, Wolff M, Huertas Migueláñez MM, Gomez-Cabrero D, Antczak P, Roca J, Cascante M, Falciani F, Maier D. 2014. The COPD Knowledge Base: enabling data analysis and computational simulation in translational COPD research. J Transl Med.

[26] Tan A, Tripp B, Daley D. 2011. BRISK-research-oriented storage kit for biology-related data. Bioinformatics.

[27] Saulnier Sholler GL, Ferguson W, Bergendahl G, Currier E, Lenox SR, Bond J, Slavik M, Roberts W, et al. 2012. A pilot trial testing the feasibility of using molecular-guided therapy in patients with recurrent neuroblastoma. J Cancer Ther.

[28] Natter MD, Quan J, Ortiz DM, Bousvaros A, Ilowite NT, Inman CJ, Marsolo K, McMurry AJ, et al. 2013. An i2b2-based, generalizable, open source, self-scaling chronic disease registry. J Am Med Informatics Assoc.

[29] Nguyen TD, Raniga P, Barnes DG, Egan GF. 2015. Design, implementation and operation of a multimodality research imaging informatics repository. Heath Inf Sci Syst.

[30] Aurrecoechea C, Barreto A, Basenko EY, Brestelli J, Brunk BP, Cade S, Crouch K, Doherty R, Falke D, Fischer S, Gajria B, Harb OS, Heiges M, Hertz-Fowler C, Hu S, Iodice J, Kissinger JC, Lawrence C, Li W, Pinney DF, Pulman JA, Roos DS, Shanmugasundram A, Silva-Franco F, Steinbiss S, Stoeckert CJ Jr, Spruill D, Wang H, Warrenfeltz S, Zheng J. 2017. EuPathDB: the eukaryotic pathogen genomics database resource. Nucleic Acids Res.

[31]Waller T, Gubała T, Sarapata K, Piwowar M, Jurkowski W. 2015. DNA microarray integromics analysis platform. BioData Min.

[32] Battke F, Nieselt K. 2011. Mayday SeaSight: combined analysis of deep sequencing and microarray data. PLoS One.

[33] Stöckel D, Kehl T, Trampert P, Schneider L, Backes C, Ludwig N, Gerasch A, Kaufmann M, Gessler M, Graf N, Meese E, Keller A, Lenhof HP. 2016. Multi-omics enrichment analysis using the GeneTrail2 web service. Bioinform.

[34]Xia T, Hemert JV, Dickerson JA. 2010. OmicsAnalyzer: a Cytoscape plug-in suite for modeling omics data. Bioinformatics.

[35] Bohler A, Eijssen LM, van Iersel MP, Leemans C, Willighagen EL, Kutmon M, Jaillard M, Evelo CT. 2015. Automatically visualise and analyse data on pathways using PathVisioRPC from any programming environment. BMC Bioinformatics.

[36] Kuo TC, Tian TF, Tseng YJ. 2013. 3Omics: a web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data. BMC Syst Biol.

[37]García-Alcalde F, García-López F, Dopazo J, Conesa A. 2011. Paintomics: a web based tool for the joint visualization of transcriptomics and metabolomics data. Bioinformatics.

[38] Rance B, Canuel V, Countouris H, Laurent-Puig P, Burgun A. 2016. Integrating heterogeneous biomedical data for cancer research: the CARPEM infrastructure. Appl Clin Inform.

[39] Bauer CR, Knecht C, Fretter C, Baum B, Jendrossek S, Rühlemann M, Heinsen FA, Umbach N, Grimbacher B, Franke A, Lieb W, Krawczak M, Hütt MT, Sax U. 2017. Interdisciplinary approach towards a systems medicine toolbox using the example of inflammatory diseases. Brief Bioinform.

[40] Satagopam V, Gu W, Eifes S, Gawron P, Ostaszewski M, Gebel S, Barbosa-Silva A, Balling R, Schneider R. 2016. Integration and visualization of translational medicine data for better understanding of human diseases. Big Data.

[41] Herzinger S, Grouès V, Gu W, Satagopam V, Banda P, Trefois C, Schneider R. 2018. Fractalis: a scalable open-source service for platform-independent interactive visual analysis of biomedical data. Gigascience.

[42] Bussery J, Denis LA, Guillon B, Liu P, Marchetti G, Rahal G. 2018. eTRIKS platform: conception and operation of a highly scalable cloud-based platform for translational research and applications development. Comput Biol Med.

[43] Pandis I, Guo Y, Guitton F, Yang X, Sun K, Wang S, Jullian N, Sousa AR, Bansal AT, Corfield J, Pavlidis S, Hekking PP, Fleming LJ, Shaw D, Roberts G, Fitch N, Riley JH, Wagers SS, Rowe A, Adcock IM, Chung KF, Auffray C, Sterk PJ. 2015. eTRIKS IT platforms for large-scale biomedical research. Eur Respir J.

[44] A. Oehmichen, F. Guitton, K. Sun, J. Grizet, T. Heinis and Y. Guo, "eTRIKS analytical environment: A modular high performance framework for medical data analysis," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 353-360, doi: 10.1109/BigData.2017.8257945.

[45] Abend A. et al. 2009. Integrating clinical data into the i2b2 repository. Summit Transl. Bioinform.

[46] Acmg BoD. 2017 Laboratory and clinical genomic data sharing is crucial to improving genetic health care: a position statement of the American College of Medical Genetics and Genomics. Genet. Med.

[47] Athey B.D. et al. 2013. tranSMART: an open source and community-driven informatics and data sharing platform for clinical and translational research. AMIA Jt. Summits Transl. Sci. Proc.

[48] Begley C.G., Ellis L.M. 2012. Drug development: raise standards for preclinical cancer research. Nature.

[49] Dunn W.Jr. et al. 2017. Exploring and visualizing multidimensional data in translational research platforms. Brief Bioinf.

[50] Elefsinioti A. et al. 2016. Key factors for successful data integration in biomarker research. Nat. Rev. Drug Discov.

[51] FDA. 2014. Providing regulatory submissions in electronic format – standardized study data, guidance for industry. In: Eletronic Submission: US Department of Health and Human Services, and Food and Drug Administration.

[52] Satagopam V. et al. 2016. Integration and visualization of translational medicine data for better understanding of human diseases. Big Data.

[53] Wilkinson M.D. et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data, 3.

[54] Yamamoto K. et al. 2017. A pragmatic method for transforming clinical research data from the research electronic data capture 'REDCap' to Clinical Data Interchange Standards Consortium (CDISC) Study Data Tabulation Model (SDTM): development and evaluation of REDCap2SDTM. J. Biomed. Inform.

8.3 Presenting and sharing clinical data using the eTRIKS Standards Master Tree for tranSMART

Adriano Barbosa-Silva, Dorina Bratfalean, Wei Gu, Venkata Satagopam, Paul Houston, Lauren B Becnel, Serge Eifes, Fabien Richard, Andreas Tielmann, Sascha Herzinger, Kavita Rege, Rudi Balling, Paul Peeters, Reinhard Schneider

Barbosa-Silva A, Bratfalean D, Gu W, et al. Presenting and sharing clinical data using the eTRIKS Standards Master Tree for tranSMART. Bioinformatics. 2019;35(9):1562-1565. doi:10.1093/bioinformatics/bty809

8.3.1 Introduction

The European Translational Information and Knowledge Management Services (eTRIKS, https://www.etriks.org/, 2017) is tasked with providing tools and services to support data management and analysis for >60 diverse biomedical research projects which have been funded by the Innovative Medicines Initiative (IMI). As Europe's largest public-private partnership, IMI funds projects ranging from molecular and systems biology to clinical trials and full translational research projects. The community translational research system under use is tranSMART (Athey *et al.*, 2013; Dunn *et al.*, 2017), first developed by the pharma industry, and then gifted to a global translational research development community. The tranSMART system has undergone extensive development extended by its own community and the eTRIKS project, which has focused on an implementation that serves IMI projects users based in the European Union (EU). The flexibility and capability of tranSMART is well presented in a recent paper showing the availability of workflows within a sandbox environment (Satagopam *et al.*, 2016). tranSMART serves as the central knowledge management system for eTRIKS, while other tools and complimentary services applicable to the data value chain, such as data harmonization, sharing, analysis, visualization and preservation, have been developed. To expedite medical breakthroughs the sharing of clinical research data is vital owing to legislative incentives and increased public pressure, many clinical trial registries are expanding their remit to share not only basic summary trial registration data but also results. Wider data sharing is one way of tackling reporting bias by increasing visibility of successful studies as well as failed ones. Additionally, data standards play a pivotal role in tackling the omnipresent problem of reproducibility. Begley *et al.* reproduced 53 experiments from landmark publications to find 47 out of 53 could not be replicated; a very worrying trend for preclinical studies that are used as the scientific basis for target identification for new drug development (Begley and Ellis, 2012).

The Data FAIRport initiative in 2014 prescribed a set of guiding principles known as FAIR: Findable, Accessible, Interoperable, Reusable which should be applied where data is deemed

scientifically valuable (Wilkinson *et al.*, 2016). Those principles have gained official recognition from G20, NIH and the Directorate General for Research and Innovation of the European Commission. The consistent application of common semantics and data structures, as outlined within data standards, is a key factor to ensure interoperability and reusability of data. The eTRIKS Data Standards Work Package created a Standards Starter Pack (https://doi.org/10.5281/zenodo.50398/, 2016), which outlines the FAIR principles and recommendations for the main clinical and genomic standards as well as supporting vocabularies and minimum information guidelines that should be applied in the entire translational research landscape. eTRIKS has also produced the IMI Data catalogue which centralizes metadata of ongoing and past IMI projects. It is part of the service that eTRIKS provides in its key knowledge management performance with a focus on the findability of project level study description metadata. Furthermore, this well received initiative facilitates broader sharing and accessibility of data (http://datacatalog.elixir-luxembourg.org/ckan/, 2017).

For clinical research data, The Clinical Data Interchange Standards Consortium (CDISC, https://www.cdisc.org/, 2018) data standards have been implemented in over 90 countries, and are now mandated by Food and Drug Administration of the United States (FDA, 2014) and Pharmaceuticals and Medical Devices Agenda (PMDA) in Japan (https://www.pmda.go.jp/files/000206449.pdf, 2018) in order to increase the uptake of data standards, which, when applied, contribute to higher data quality. The lack of implementing standards will render datasets from different cohorts inadequate when integrating with complementary research data for meta-analyses (Elefsinioti *et al.*, 2016). A recent paper by the American College of Medical Genetics and Genomics (Acmg, 2017) discussed the importance of using the information from one patient cohort to benefit other patients. The ACMG's framework for data sharing will work best if standards are implemented within the framework, as within tranSMART, and datasets are gathered by utilizing those standards from the beginning of the research, as is also recommended by CDISC.

8.3.2 Implementation

The eTRIKS Standard Master Tree is based on the standards for clinical data representation developed by CDISC, mainly the Study Data Tabulation Model (SDTM) standard. The proposal of eTRIKS was to create a hierarchical navigation tree in which the raw data, collected at the multiple cohorts, should be promptly mapped to the elements of this tree so that data are loaded automatically with the correct topology into tranSMART i2b2 (Informatics for Integrating Biology and the Bedside) framework. The requirement for this is that all the data collected from a patient will be organized and formatted using the SDTM model. SDTM modeling increases the ability to compare information among systems and/or organizations, whilst also decreasing the time to initiate a new research study. The use of these data standards improves the data quality, their interoperability and their management, which allows easier, faster and more reliable data aggregation.

The eTRIKS Standard Master Tree presents the clinical data within tranSMART i2b2. eTRIKS has radically updated the original tranSMART engine that sorts and presents the clinical data within the system. Users can choose to map their clinical data content to a favorite terminology prior to the SDTM modelling using global standards such as OMICS, NCI (https://www.cancer.gov/digital-standards, 2017) or LOINC (https://loinc.org/, 2017), as long as the SDTM variable names as maintained. Further, the clinical data is mapped to a 'clinical mapping file', which requires a good working knowledge of the CDISC foundational standards, in order to represent the SDTM structure of the clinical data correctly in the hierarchy of the tranSMART i2b2 repository (Abend *et al.*, 2009).

In practice if one thinks about the outcome of a 'glucose test', this test may be named 'sugar test' or 'glucose test' in different differing cohorts, which may be well understood by experts but not a machine as the same concept. The use of standard name 'Glucose Tolerance Test' (NCBI's, MeSH Unique ID: D005951) would avoid any confusion or wrong interpretation and enable data query across cohorts. Further to this, considering that the metabolite 'glucose' could be measured in different samples (e.g. blood, urine), the test results could be reported in different units (mg/dl or mmol/l) and/or the test could be performed at different periods of the time (screening, visit 1, visit 2, etc.), error prone aspects during the data analysis. If the problem is proposed, 'How to standardize the manner by which this information should be organized and formatted for effective and precise cohorts comparisons?' One answer should be: 'Use a Standard Master Ontology Tree' or in this case, the eTRIKS Standard Master Tree. The application of this tool coupled with a good application of controlled vocabularies will increase greatly the Reusability and Interoperability Principles mentioned above.

In the 'Glucose Tolerance Test' example, upon mapping to the tranSMART Standard Master Tree, the outcome of this test would already be represented as displayed in Figure 1A below. The test result is reported in this example by means of 14 variables (columns A-N) for the subject CDISC01.100008 (column C). Note that column G collects one variable called LBTEST (Lab Test Examination Name), which is filled with the standard value 'Glucose' and another variable LBSPEC (Specimen Type) is used to distinguish 'BLOOD' from 'URINE' samples. In terms of readout values, the variable LBORRES (Result or Finding in Original Units) records the original values as collected reported units in LBORRESU (Original Units) the unit itself (e.g. mg/dl). The example shows results converted to numeric type and this reported value to a standard unit, which is achieved by using the pair variables LBSTRESN (Numeric Result/Finding in Standard Units) and LBSTRESU (Standard Units), for values and units (e.g. mmol/l), respectively. The SDTM Implementation Guide provides a comprehensive description including four sessions: 1—Overview of topics for specific general observation class associated with specific domains, 2—Specification for table of variables, 3—Rules for correct implementation of standards and 4—Examples.

Fig. 1.

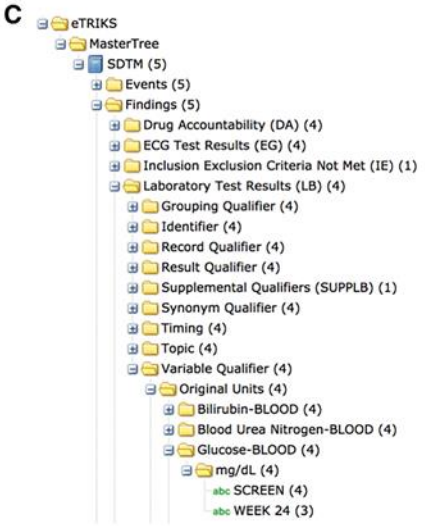
A

STUDYID	DOMAIN	USUBJID	LBSEQ	LBREFID	LBTESTCD	LBTEST	LBORRES	LBORRESU	LBSTRESN	LBSTRESU	LBNRIND	LBSPEC	VISIT
CDISC01	LB	CDISC01.100	5	B232115	GLUC	Glucose	100	mg/dL	5.5	mmol/L	NORMAL	BLOOD	SCREEN
CDISC01	LB	CDISC01.100	6	B290028	GLUC	Glucose	87	mg/dL	4.8	mmol/L	NORMAL	BLOOD	WEEK 24
CDISC01	LB	CDISC01.100	19	B232115	GLUC	Glucose	NEGATIVE				NORMAL	URINE	SCREEN
CDISC01	LB	CDISC01.100	20	B290028	GLUC	Glucose	TRACE				NORMAL	URINE	WEEK 24
CDISC01	LB	CDISC01.100	21	B232115	GLUC	Glucose	10	mg/dL	0.6	mmol/L	NORMAL	URINE	SCREEN
CDISC01	LB	CDISC01.100	22	B290028	GLUC	Glucose	30	mg/dL	1.7	mmol/L	NORMAL	URINE	WEEK 24

B

Seq. For Order	Observation Class	Domain Prefix	Domain	Variable Name	Role	Variable Label	Data Label	Visit Name
1	Findings	LB	Laboratory_Test_Results_(LB)	STUDYID	Identifier	Study_Identifier		
2	Findings	LB	Laboratory_Test_Results_(LB)	DOMAIN	Identifier	Domain_Abbreviation		
3	Findings	LB	Laboratory_Test_Results_(LB)	USUBJID	Identifier	Unique_Subject_Identifier		
4	Findings	LB	Laboratory_Test_Results_(LB)	LBSEQ	Identifier	Sequence_Number		
5	Findings	LB	Laboratory_Test_Results_(LB)	LBGRPID	Identifier	Group_ID		
6	Findings	LB	Laboratory_Test_Results_(LB)	LBREFID	Identifier	Specimen_ID		
7	Findings	LB	Laboratory_Test_Results_(LB)	LBSPID	Identifier	Sponsor-Defined_Identifier		
8	Findings	LB	Laboratory_Test_Results_(LB)	LBTESTCD	Topic	Lab_Test_or_Examination_Short_Name		
9	Findings	LB	Laboratory_Test_Results_(LB)	LBTEST	Synonym_Qualifier	Lab_Test_or_Examination_Name		
10	Findings	LB	Laboratory_Test_Results_(LB)	LBCAT	Grouping_Qualifier	Category_for_Lab_Test		
11	Findings	LB	Laboratory_Test_Results_(LB)	LBSCAT	Grouping_Qualifier	Subcategory_for_Lab_Test		
12	Findings	LB	Laboratory_Test_Results_(LB)	LBORRES	Result_Qualifier	Result_or_Finding_in_Original_Units		
13	Findings	LB	Laboratory_Test_Results_(LB)	LBORRESU	Variable_Qualifier	Original_Units		
14	Findings	LB	Laboratory_Test_Results_(LB)	LBORNRLO	Variable_Qualifier	Reference_Range_Lower_Limit_in_Orig_Unit		
15	Findings	LB	Laboratory_Test_Results_(LB)	LBORNRHI	Variable_Qualifier	Reference_Range_Upper_Limit_in_Orig_Unit		
16	Findings	LB	Laboratory_Test_Results_(LB)	LBSTRESC	Result_Qualifier	Character_Result/Finding_in_Std_Format		
17	Findings	LB	Laboratory_Test_Results_(LB)	LBSTRESN	Result_Qualifier	Numeric_Result/Finding_in_Standard_Units		
18	Findings	LB	Laboratory_Test_Results_(LB)	LBSTRESU	Variable_Qualifier	Standard_Units		
19	Findings	LB	Laboratory_Test_Results_(LB)	LBSTNRLO	Variable_Qualifier	Reference_Range_Lower_Limit-Std_Units		
20	Findings	LB	Laboratory_Test_Results_(LB)	LBSTNRHI	Variable_Qualifier	Reference_Range_Upper_Limit-Std_Units		
21	Findings	LB	Laboratory_Test_Results_(LB)	LBSTNRC	Variable_Qualifier	Reference_Range_for_Char_Rslt-Std_Units		
22	Findings	LB	Laboratory_Test_Results_(LB)	LBNRIND	Variable_Qualifier	Reference_Range_Indicator		
23	Findings	LB	Laboratory_Test_Results_(LB)	LBSTAT	Record_Qualifier	Completion_Status		
24	Findings	LB	Laboratory_Test_Results_(LB)	LBREASND	Record_Qualifier	Reason_Test_Not_Done		
25	Findings	LB	Laboratory_Test_Results_(LB)	LBNAM	Record_Qualifier	Vendor_Name		
26	Findings	LB	Laboratory_Test_Results_(LB)	LBLOINC	Synonym_Qualifier	LOINC_Code		
27	Findings	LB	Laboratory_Test_Results_(LB)	LBSPEC	Record_Qualifier	Specimen_Type		
28	Findings	LB	Laboratory_Test_Results_(LB)	LBSPCOND	Record_Qualifier	Specimen_Condition		
29	Findings	LB	Laboratory_Test_Results_(LB)	LBMETHOD	Record_Qualifier	Method_of_Test_or_Examination		
30	Findings	LB	Laboratory_Test_Results_(LB)	LBBLFL	Record_Qualifier	Baseline_Flag		
31	Findings	LB	Laboratory_Test_Results_(LB)	LBFAST	Record_Qualifier	Fasting_Status		
32	Findings	LB	Laboratory_Test_Results_(LB)	LBDRVFL	Record_Qualifier	Derived_Flag		
33	Findings	LB	Laboratory_Test_Results_(LB)	LBTOX	Variable_Qualifier	Toxicity		
34	Findings	LB	Laboratory_Test_Results_(LB)	LBTOXGR	Variable_Qualifier	Standard_Toxicity_Grade		
35	Findings	LB	Laboratory_Test_Results_(LB)	VISITNUM	Timing	Visit_Number		
36	Findings	LB	Laboratory_Test_Results_(LB)	VISIT	Timing	Visit_Name		VISIT_NAME
37	Findings	LB	Laboratory_Test_Results_(LB)	VISITDY	Timing	Planned_Study_Day_of_Visit		
38	Findings	LB	Laboratory_Test_Results_(LB)	LBDTC	Timing	Date/Time_of_Specimen_Collection		
39	Findings	LB	Laboratory_Test_Results_(LB)	LBENDTC	Timing	End_Date/Time_of_Specimen_Collection		
40	Findings	LB	Laboratory_Test_Results_(LB)	LBDY	Timing	Study_Day_of_Specimen_Collection		
41	Findings	LB	Laboratory_Test_Results_(LB)	LBTPT	Timing	Planned_Time_Point_Name		
42	Findings	LB	Laboratory_Test_Results_(LB)	LBTPTNUM	Timing	Planned_Time_Point_Number		
43	Findings	LB	Laboratory_Test_Results_(LB)	LBELTM	Timing	Planned_Elapsed_Time_from_Time_Point_Ref		
44	Findings	LB	Laboratory_Test_Results_(LB)	LBTPTREF	Timing	Time_Point_Reference		
45	Findings	LB	Laboratory_Test_Results_(LB)	LBRFTDTC	Timing	Date/Time_of_Reference_Time_Point		
46	Findings	LB	Laboratory_Test_Results_(LB)	LBTEST-LBSPEC-LBMETHOD	DATA_LABEL		DATA_LABEL	

C



Content representation of the eTRIKS Master Tree Package. (**A**) SDTM data file for one patient (USUBJID) for the LB domain. (**B**) SMOT_Lite definition session for the LB domain. (**C**) Hierarchical (i2b2) tree created for the LB domain and displayed in the tranSMART web app

To avoid the all too common pitfalls of redundant data eTRIKS developed the Standard Master Tree, using the comprehensive SDTM domain structure, to support and give structure and context to the data so it can be easily identified. The Standard Master Tree follows a basic and easy-to-understand logic, which was built upon the premise of tranSMART rules for data loading. This means that multiple data collected for one patient for the same domain (e.g. Laboratory Test Results—LB) should be distinguished based on *Data Labels*. This way, for the LB domain, results of 'Glucose' and 'Creatinine' tests for example, could be loaded in the same run. Moreover, multiple results for the same test should also be distinguished based of the *Visit Names*. Respecting these two basic rules, any results from any sort of laboratory tests and even results for any other domains, can be easily represented in tranSMART via the Standard Master Tree.

8.3.3 Features

The eTRIKS Standard Master Tree model consists of a package of three main components: (i) a CDISC clinical dataset reported as define.xml metadata and converted in *.txt* tabulate files composing of 16 SDTM domains represented as review data as collected for 4 fictitious subjects (Fig. 1A); (ii) the tranSMART standard master ontology tree as TM SMOT-

SDTM_lite.txt definition file, where all the information concerned about the correct positioning of the SDTM variables can be found (Fig. 1B); and the (iii) Mapper script where users can map their data files to the TM SMOT-SDTM_Lite definition, avoiding manual work. The script reads a target directory containing the input SDTM files and maps all the collected variables against the SMOT-SDMT_Lite.txt master tree file mentioned above. This is achieved with a single command line: 'php mapper.php SMOT_Lite.txt CDISC01_ClinicalData' (further details are explained on the package's README file). Figure 1A depicts an example for the Glucose test of one such subject. Figure 1Bdepicts part of the TM SMOT-SDTM_Lite file where definitions for the domain LB is displayed (Note the seven columns required for the annotation of each of the SDTM variables used in this domain). This information can be found easily on the SDTM implementation guide as should be adopted by the data curators. Finally, Figure 1Cdisplays the graphical hierarchy tree, known as tranSMART i2b2 tree, where the loaded data can be further queried and used to create comparison subsets on the tranSMART i2b2 web app, these can be visualized in a sandbox implementation available at http://public.etriks.org/transmart/datasetExplorer under the eTRIKS—Master Tree branch.

The strategy for clinical research data standards representation proposed above offers a readily available method to integrate multiple translational research datasets while meeting the Interoperability and Reusability aspects of the FAIR principles. Once the data is within the eTRIKS Standard Master Tree, it can then take advantage of the tranSMART environment, where it will receive a unique study and server specific identifier and the metadata can be given greater and essential specificity. With an effective tranSMART search tool where multiple datasets and/or studies can be pooled and queried, coupled with an entry within the eTRIKS data catalogue the data has undergone FAIR-ification to a satisfactory degree. Now the data is Findable and also Accessible, and it can begin its hopefully long life adding scientific value to any number of future studies or aggregated data comparisons.

8.3.4 Conclusions

The tranSMART Standard Master Tree presented here adds to other efforts to make other software data interoperable with tranSMART. Projects such as 'ODM to i2b2' converts data stored in XML/ODM based systems such as OpenClinica and REDCap into i2b2 format (https://github.com/CTMM-TraIT/trait_odm_to_i2b2, 2018); and 'REDCap2SDTM' converts electronic data capture system data to SDTM (Yamamoto *et al.,* 2017). Taken together, this software could benefit from the Master Tree concept in order to standardize the manner that SDTM studies should appear within a tranSMART navigation tree to users.

If the tools and processes above are adopted in the scope of the NIH funded projects, it will contribute greatly to creating an overseas bridge for data sharing initiatives with the EU/EFPIA-funded (IMI) translational medicine research projects, of which over 60 are being supported by the eTRIKS project. While not all of the eTRIKS supported projects have implemented the tranSMART Standards Master Tree they have all received the appropriate guidance and advice from eTRIKS experts or as laid out in the eTRIKS standards starter pack.

Tremendous curation efforts were necessary to guarantee that IMI data was collected in good quality, once that, frequently, the big challenge for translational research projects lies on the quality of the data itself, not only its metadata. The adoption of the technologies and standards developed and presented in this paper will support a significant step towards a position where IMI data can be shared, and the findings reproduced to benefit the health care research community, allowing a standardized representation of SDTM data across multiple tranSMART servers. The eTRIKS Standard Master Tree package can be downloaded at https://doi.org/10.5281/zenodo.1009098.

Chapter 9: Meditations on the Nature of Open Source Software

Jay Bergeron

9.1 Open Source Software and Scientific Research

Although robust commercial solutions existed to support molecular profiling, particularly gene expression, in the mid to late 2000's, these solutions did not comprehensively support corresponding low dimensional clinical datasets including demographics, labs, study endpoints and assessments. Johnson and Johnson (J&J) created the tranSMART platform, as described in chapter six, in order to integrate these corresponding low dimensional clinical and high dimensional biomarker patient datasets. Developing a system such as tranSMART is an enormous undertaking requiring substantial investment in funding and personnel time. Although J&J likely viewed tranSMART as a competitive advantage, the cost to develop and release subsequent versions of the software caused J&J's leadership to consider open licensing the product to, hopefully, take advantage of community investment (per my remembrance of a conversation with Sandor Szalma, a leader in tranSMART's development). As a first step, tranSMART was used to manage the translational data of the IMI U-BIOPRED consortium as described in chapter one. Based on the U-BIOPRED success, J&J released tranSMART under the GNU Public License version 3 (GPLv3) leading to commissioning of the eTRIKS consortium and use by other scientific programs. Notable programs using tranSMART include the Translational IT (TraIT) public private partnership based in the Netherlands and Zachary Cohane's group at the Harvard Medical School (this team developed *Informatics for Integrating Benchtop to the Bedside* (I2B2), the foundational application upon which tranSMART was built). Paul Avillach joined the Cohane group at roughly the time tranSMART was licensed as open source. Paul's team has consistently developed and utilized the Harvard tranSMART version to great advantage for many largescale disease studies, leveraging the best complementary capabilities of both I2B2 and tranSMART.

In anticipation of the open source tranSMART release, several Pharmaceutical and scientific informatics companies, following a pattern associated with many highly successful open source systems, formed the *tranSMART Foundation* (tF) in order to guide and harmonize tranSMART feature development and promote the use of tranSMART.

However, the initial GPLv3 release of tranSMART was dependent on the *Oracle* relational database management system (RDBMS), a commercial product. As such, Oracle licenses were required to use the initial tranSMART public release and the Oracle RDBMS could not be modified or developed outside of the Oracle product roadmap. As academic groups generally preferred to use a fully open source stack to reduce cost and provide code visibility eTRIKS software developers, working with the University of Michigan Computational Biology

department, created a tranSMART version that could be used with either the Oracle or the open source RDBMS PostgreSQL. This version was co-released by eTRIKS and the tF, becoming the base implementation to power many translational research projects and consortia.

Subsequent releases attempted to imbue tranSMART with advanced capabilities such as cross-study analysis and robust longitudinal/temporal support, integrating these capabilities developed in isolation by disparate teams. Based on highly successful open source projects such as Linux, Star Office, MySQL, the R statistical program, PostgreSQL and others, this model of feature enhancement was expected to be highly enabling. However, to a great extent these version integrations proved problematic to harmonize and typically failed to be widely adopted. Feature enhancement was, by contrast, usually implemented by creative study configurations conceived by expert data curators or by custom enhancements specific to individual projects or development teams. The result being version segregation with the pursuit of a single harmonized feature-consolidated tranSMART codebase became economically infeasible.

As to why Linux, Star Office and many other open source systems were strategically enhanced for community-wide adoption and utilization while tranSMART suffered from version fragmentation is an exceedingly important question. Although there are multiple contributing factors, the most straight-forward compelling explanation lies in client-based variability of seemingly shared requirements. For example, cross study analysis and longitudinal study support appear to be reasonably well understood as general concepts. However, the high degree of study to study dataset variability coupled with the correspondingly variable and complex data use and analysis patterns greatly complicates the detailed implementation of such features. A cross-study analysis or longitudinal implementation that satisfies a specific research team may be wholly inadequate for other teams, although each team of researchers describe a consistent general functionality need. The dissonance between high level feature description and explicit user implementation requirements was consistently observed in practice.

This dissonance was also observed for even apparent simplistic custom features. The company for which I work created a tranSMART enhancement that allowed company researchers to create and utilize a large collection of genome wide association (GWA) summary statistics. This collection eventually comprised tens of billions of records. Now migrated to another system as part of an evolving information strategy, the GWA collection resided on tranSMART effectively and cost efficiently for seven years of consistent use and expansion. Other groups attempted to adopt this feature and failed due to variable detailed requirements and tolerance for investing in updating their own company tranSMART systems. GWA summary statistics can be described by a straight-forward data structure relating individual genetic variants to computed summary values (i.e. a grid of values with the number of records equivalent to individual variants tested for a specific phenotype projected across a set of ~50 summary statistics, the number of records will grow substantially with each phenotype tested but the data relationship remains rather simple). However, varying data quality processes, raw data

integration requirements, dominant query patterns (intra vs. cross dataset) and a variety of other factors important to individual work groups seriously complicated what was expected to be an easily distributable feature set.

In a more cogent example, two academic groups implemented two disparate integrations of tranSMART and the open source image utility XNAT to enable queries of low dimensional clinical data with corresponding medical images. Each implementation was presented side by side at the October 2015 tranSMART Foundation community meeting (see also Chapter 8 references). The apparent redundancy of effort was logically justified through detail level distinctions in the types of queries needed by each client group.

Building Linux or Star Office is (really, really) difficult. However, the development process is much facilitated by modeling these open source systems on well-established commercial counterparts with thriving customer communities. With clearly understood use patterns coupled with an underserved user population which, for example, may not be able to afford, or may need to customize, the commercial product there emerges an economic incentive to create the open source substitute. In highly successful cases such as Star Office and PostgreSQL the open source competitor may eventually encroach on the market of the commercial incumbent.

If existing applications, having clearly established use patterns, do not exist to guide the open source development, expect risks with respect to open source solution development and adoption and plan pertinent risk avoidance or mitigation measures.

I can attest that the tranSMART community (eTRIKS, TraIT, Harvard, tF and beyond) was populated by astoundingly talented, intelligent, educated, experienced and enthusiastic people who genuinely strived to work together to build impactful software for translational research. The success in distributing the base tranSMART platform and the difficulty in consolidating the various functionally “decorated” tranSMART flavors into successful harmonized releases led to a gamut of community reactions from elation, solidarity, intellectual debate and lively disagreement. In the end, a community-developed core system which is customized by/for specific client communities is a productive model for solution deployment. Granted this model diverges from that of efficient evolutionary integration associated with some of the most powerful, distributed and famous open source products. However, astute communities possessing self-awareness can achieve enviable results with community development models that diverge from the perceived ideal. Applying public-private funding in such circumstances (eTRIKS as an example) is a competent consideration for meeting public goals by accountable corporate and government officials.

Given the high regard and broad use of open source systems within the scientific community, it is important for researchers to understand the basic concepts underlying open source economics given their likelihood of engaging open source software products and communities. It is likely that a utopian vision of altruistic developers building, through shear virtuous passion,

open solutions to rival commercial juggernauts is somewhat naïve. However, there is clearly demonstrable and important economic potential and positioning for open source software.

Scientists may very well need to answer questions of the following nature.

1. As a principal investigator, how do I determine whether a commercial or open source application is best for my laboratory, what factors and risks are pertinent beyond the basic comparison of existing capabilities and cost?

2. As a government funding decision maker, is it proper or ethical to fund an open source development effort that may compete with commercial products?
    a. What conditions would contribute to an anti-competitive environment?
    b. Is there open source potential to simplify/optimize procedural compliance?
    c. Is there open source potential to anchor productive competitive marketplaces?

3. As a software developer, is it in my customer's best interest to provide a custom solution or is there benefit to contributing to and delivering an open source solution? What explicit factors should I assess to better understand the risks and benefits?

I invite those wishing to explore these concepts more to read the full chapter, the content of which was originally accepted as a dissertation by the faculty at the Lally School of Management, Rensselaer Polytechnic Institute in April of 2013, six months following the launch of eTRIKS. To have a theoretical viewpoint of the nature of open source development is useful. To experience this nature unfold as a leader in a sizeable international open source endeavor, living the successes, challenges and disappointments, has been uniquely satisfying. Further exploration into this subject matter, which I continue to find profoundly fascinating, begins with this abstract.

> *Motivations that give rise to voluntary participation by software developers in Open Source Software (OSS) projects have been well analyzed[9 40 37 4]. Socio-Psychological factors that include the potential for individual development and personal recognition, as well as the opportunity to contribute to self-selected high value efforts, have been promoted as drivers of OSS contribution. Substantial work has been conducted to relate architectural aspects of OSS, including the extent of design modularity and differentiated option value for modular components, with the free-rider tolerance associated with successful OSS initiatives[9 6]. However, empirical evidence to support the hypothesized relationship between architecture and OSS participation is limited. Moreover, the extent to which Socio-Psychological factors promote OSS participation is difficult to quantify given substantial OSS investments by governments and commercial enterprises. Additionally, open source licensing models do not preclude business development activities such as commercial software extensions, consultancies and discretionary pricing models[31 23]. To extend the traditional dialog regarding OSS success, patterns of business implementation, and the*

subsequent impact of such patterns on OSS end use, are considered. The high degree of imitation in the design of end user components that is apparent between OSS projects and corresponding pre-existing, often commercial, competitors cannot be ignored, nor can the impact of Information Protection policies that allow such imitation. Moreover, it is proposed that evolutionary development of new OSS features may coincide with OSS product suitability for an initial niche market that ultimately can expand to traditional market segments and challenge/supplant established commercial competitors.

9.2 Impact of Business Patterns on Voluntary Production: Imitation and Open Source Software Success

9.2.1 Introduction

Computer *software* products (alternatively *programs* or *applications*) control the physical elements of a computer (*i.e. hardware*) such that these elements can be coordinated to perform valuable tasks for humans [10]. Software products are comprised of instructions (*source code*) that conform to language contexts that can be interpreted by machine hardware. Software instructions must be written by a, typically human, *developer* having the requisite knowledge of the software language constructs as well and the goals and outcomes (i.e. software *requirements*) that future users plan to achieve by using the software [9 10].

Computer programs are typically produced using a phased development approach referred to as the software product *life cycle* [31 23]. The software product life cycle includes business analysis, product design, creation (build) and test/verification phases. Business analysis involves eliciting and documenting software requirements from future users [4 13 19]. These requirements include the capabilities that the software must provide (*functional* requirements) as well as performance, security, regulatory and other necessary operational characteristics to which the software must conform (*nonfunctional* requirements). Given the results of the business analysis, technical staff will prepare the *design* of a future software product capable of meeting the documented software requirements.

Design encompasses many elements, including the selection of hardware and network specifications, supporting software products such as operating and database management systems as well as programming language(s) [9 17]. Software is often constructed in a *modular* fashion in which related data and operations are grouped together into logical subunits. Rules and methods (termed *interfaces*) that describe the nature of interactions between the subunits are designed using standardized modeling techniques [29 17 20]. This *functional decomposition* of software programs into subunits allows for separation of responsibilities among the development team. Additionally, the isolation of software functions inherent in efficient modular design leads to advantages in maintaining and enhancing the software as program defects will be easier to identify and their correction both less laborious and less likely to result

in unintended consequences (such as the introduction of new defects due to unanticipated interactions within the code)[29]. Moreover, the design specification will include planning for *interoperability* or how the system will approach interactions, such as information exchange or the submission and receipt of instructions, with associated or dependent software products that may or may not be within the development team's control [31 25]. Application capabilities are generally exposed to other software using an *application programming interface* (API) that allows one application to use the capabilities of another without needing in depth knowledge of the design or code of the supplier application [10]. As with internal application interfaces, APIs establish the rules and mechanisms for cross program interaction and have been likened to a contractual agreement [10 17]. Potential future expansion may be considered, and planned for in advance, as part the design phase and can include preparations for managing potential new requirements or scaling to meet anticipated increases in future demand (such as greater numbers of users or higher data volumes).

The *build* phase encompasses the actual software development effort that produces a working application [31 23]. The team of software developers creates and integrates the various modules that, together, deliver against the functional and nonfunctional requirements. The *test or verification* phase of the software lifecycle ensures that the software product actually conforms to the stated requirements. Typically, authorization of the test/verification phase is contingent upon a group of end users conducting testing and authorizing the product as fit for purpose [23 13].

The linear phased software lifecycle described above (sometimes referred to as a *Waterfall* approach) is a traditional model [23 8]. There are a number of software lifecycle approaches, each having specific variations on techniques, including incremental models that divide the definition and delivery of software into smaller units of effort and versioning models that encompass long term development strategies. All life cycle approaches incorporate the basic activities of requirements analysis, design, build and verification [23 8].

Software source code can be written, modified and extended by developers but, typically, *cannot* be *executed* on hardware. Source code is written in languages that are interpretable by humans having the requisite skills in the use of the language [9 10]. However, a transformed, or *compiled*, version of the source code can be executed on a machine but, in this form, is extremely difficult for developers to modify [9]. In order to protect the intellectual property, commercial software applications are generally delivered as compiled code ready for execution while the source code is withheld. *De-compilation* of distributed compiled code (back to source code, should tools for de-compilation exist) is generally forbidden by software licensing agreements [9].

Open Source Software (OSS) is the name applied to computer applications that are licensed for unrestricted distribution and use [9 10 31]. More specifically, Open Source refers to open access to source code. OSS may be used per the discretion of the consumer and, if desired, modified by

consumers to extend the software's capabilities or to leverage the code base for alternative uses [9]. Certain OSS licenses, such as the original version of the GNU (*Gnu's Not Unix!*) Public License (GPLv1) compel developers to freely distribute products derived from OSS under the same license. This enforced distribution licensing model is known as *copyleft* or viral licensing [9]. However, some OSS licenses, such as LGPL (Lesser GPL) impart property rights onto consumers who modify OSS allowing for commercial distribution of value-added application updates and derived products. Apache, the ubiquitous OSS web server, is distributed under such a "permissive" license as are all software products licensed under the prominent *Berkley Software Distribution* (BSD) [9].

OSS products are generally perceived as being developed and maintained by a community of volunteers and freely distributed via public networks [31]. This is the case with many high-profile OSS products, such as the Linux operating system and the Apache web server [10 31 23]. The motivations for such mass volunteer efforts confounded economists although convincing explanations, including socio-psychological drivers of volunteer developers and system architecture-based enablers have emerged [30 31]. These reasons for mass volunteer participation have tended to address the design/build/test phases of the software development life cycle [31].

However, many OSS projects include commercial and government benefactors [9]. Furthermore, OSS models do not preclude the prospect of financial reward for participants. The generally cited motivational aspects of OSS success will be discussed relative to sources of financial investment and opportunity of financial gain associated with these projects. The paper will discuss alternative drivers of OSS success predicated upon the business analysis phase of the software development life cycle.

9.2.2 The Nature of OSS as a Public Good

The economist Paul Samuelson formalized the concept of *collective consumptive goods* (i.e. *public* goods) as products that can be consumed by an individual without lessening the product's consumption by others [33 26]. Public goods are characterized as both *non-rival* and *non-exclusive*. As such, public goods are differentiated from individual fee for service offerings as well as member-based (club) services, both of which are exclusive offerings that can be limiting with respect to the number of customers that can be supported [10 39]. Pragmatically, most public goods, such as public spaces, are likely to have limits in their capacity that may create situations of rivalry [39].

Public goods provide governments with an alternative to fee for service-based allocations of civic offerings. The public good model allows governments to feasibly tax public services that can be used by anyone although such services may not be used by everyone [33 9]. Samuelson also described the inherent inability to enforce equity of individual investment in public goods, later termed, and generally recognized as, the *"Free-Rider"* problem, such that individuals could, per Samuelson, "*selfishly"* conceal their true demand to take advantage of a wider community investment [33]. Out of town persons using a town-funded public park, for which

they do not pay taxes, being a simple free rider example.

The free rider problem complicates the definition of apparent public goods. The economist Ronald Coase, in a classic article popularly referred to as *"Coase's Lighthouse"* [15], demonstrates these classification challenges. Lighthouses were commonly used as examples of public goods by economists who assumed fee for use schemes with regards to these structures would not be feasible. Consequently, government support was generally presumed to be a necessity for establishing lighthouses. With these assumptions, "Free-riding" shipping lines not subject to local taxation would be expected to take advantage of the presence of light houses. However, Coase presented substantial evidence that commercial sources, as opposed to government/public sources, dominated lighthouse funding. Moreover, such funding entities were able to develop business models (port "landing fees" for example) that charged individual shipping lines for their use of these navigational installments. Lighthouse builders were able to introduce an unexpected element of exclusivity with regards to the use of lighthouses that effectively limited free-riders to tolerable levels [15].

OSS code may be downloaded and used by anyone with the requisite hardware and dependent software required to operate the OSS product and, as such, is non-exclusive by nature. OSS is generally also non-rival as there is no limitation on the number of people who can download and use the software (of course, distributed applications that use shared networks or hardware can be susceptible to increased use, note services below). Therefore, OSS licensed under permissive terms meets the definition of a public good, albeit a public good promoted by seemingly altruistic volunteerism rather than by tax allocation [9].

On-line software *services* that are provided freely/openly via the internet, such as Wikipedia, also generally conform to the definition of a public good although clearly these services may be limited (i.e. rival) relative to the number of concurrent online users or volumes of data that can be supported [10]. Although the sustainability of open internet services is worthy of study, this paper will focus on open code development and distribution inherent in OSS products rather than the delivery of open internet services.

As potential public goods, OSS projects are expected to be susceptible to free riders given the facility and anonymity associated with software downloads. That successful OSS projects are able to tolerate free riders (and given the ease of replication and distribution of software, potentially enormous discrepancies between free-rider OSS users relative to volunteer OSS developers) is therefore economically intriguing and has driven substantial interest in the value propositions associated with OSS participation[9 6]. The potential of OSS, like the lighthouse, to benefit from non-obvious business models and elements deserves investigation.

9.2.3 Motivations for Open Source Volunteers

The motivations that drive OSS volunteers have been substantially analyzed [30 31]. Social/psychological rewards have been proposed. Popularized by writers such as Dan Pink [30],

the prospect of working autonomously, the pursuit of technical mastery and the opportunity to be involved in goal drive communities that deliver value far greater than that achievable by isolated individuals are purported drivers for participation [30]. Pink notes that developers who contribute to OSS projects typically have paid commercial positions and their contributions to OSS are supplemental to their employment responsibilities. There may be no relationship between the employment goals of OSS participants and the open source projects to which they contribute [30]. The ability to select projects and the manner of contribution is attractive to professionals who view their work as artistry. Open source achievements lead to recognition outside of closed commercial environments and establish a reputation in an extended professional network [30 31]. Open source projects offer opportunities to develop new skills using cutting-edge technologies. OSS developers are able to select projects for which they have a personal interest or that are of meaning to them. For an individual developer, contribution to an OSS project allows their creative expression to be distributed far more broadly than contributions to proprietary software products [27].

Beyond altruistic motives, the opportunity for unfettered creation and the pursuit of personal development, there exist financial opportunities for those involved in OSS projects [23 10]. Although OSS code may be distributed for free, consulting positions, the creation of technical and user documentation, premier *for profit* versions of OSS applications, OSS integration with commercial applications or processes, software packaging for facilitated installation and OSS hosting services are all wage-earning opportunities for those having expertise with OSS projects that are widely adopted. Furthermore, employment opportunities are manifested with commercial and government/academic investment in OSS projects [23 31].

9.2.4 Government Interest in OSS Projects

Government entities play a key role in the advancement of OSS. By 2004, there were 44 countries and 99 (local and national) government entities having established pro-OSS policies [10]. These policies were primarily enacted through administrative functions including software procurement and via grants and subsidies aimed at training software developers to use OSS technologies. OSS has also been promoted through direct legislation albeit to a lesser extent and often restricted to local or regional government entities [10]. The perception that OSS provides lower cost alternatives to commercial software is, as can be expected, a primary factor driving government interest [10].

However, licensing is only one element that contributes to the total cost of ownership for software products. Hosting environments, support and maintenance activities, software customization and enhancement, the availability of skilled technicians and the interoperability of OSS applications with the myriad of allied software solutions required to support the breadth and scale of government activities must be considered [10 23]. Service level commitments for application support are typically included in time-bound licensing schemes and can be exceedingly valuable for software customers such as government entities. For OSS, service level agreements will need to be internalized or sourced from independent commercial ventures that have either built the requisite OSS expertise as part of business development strategies or

have been spun off from the OSS community [10 23].

It is common for OSS products to be created by academic research groups [9]. The release of software developed through public investment may be an outcome mandated by the academic funding organization. However, software developed by academic groups is commonly provided freely to public researchers while requiring paid licenses for commercial users. This discretionary pricing model provides income to the academic developers should the software become widely adopted as a standard solution. As an example, the Genome Analysis Tool Kit (GATK), an open source genomics analysis package developed by the *Broad Institute*, has become a de facto industry standard. Recently, the Broad Institute partnered with *Appistry*, a commercial genomics solution provider, to deliver a for profit version (V2.0) of the GATK that includes licensing and support services. The prior GATK version was branded as a "Lite" version and remains free for both commercial and academic users. Academic users may update to GATK V2.0 for free while commercial users must purchase a license from Appistry [see http://www.broadinstitute.org/gatk/]. Such academic discretionary pricing arrangements are proper in the United States under the *Bayh-Dole* act of 1980. The Bayh-Dole act allows recipients of federally funded grants to retain title to *"any invention of the contractor that is conceived or first actually reduced to practice in the performance of work under a funding agreement"* [http://www.csurf.org/enews/bayhdole_403.html]. Legislated to spur commercialization of academic research patents, there are many examples of commercialization of products created by academic researchers using federal funding [http://www.csurf.org/enews/bayhdole_403.html]. For OSS software, enhancements created can be bundled and commercialized. The author is not aware of case law regarding the ownership of OSS contributions to products that are distributed under discretionary licensing policies. The author suspects that the integration of OSS contributions, when carried out by the licensing academic or commercial entity, establishes the licensing entity as the inventor of the new capabilities [See [23] and Raymond for a description of the value added distribution (RedHat)].

Although governments may benefit from no-cost licensing and discretionary pricing practices of the academic recipients of government grants, government support or production of public goods, such as OSS, is generally applicable to those products for which there is no proprietary incentive or capability to produce [10]. However, there are instances of government support for OSS projects for which there are commercial competitors. The suitability of government-sponsored OSS development that directly competes with established proprietary software is certainly debatable and is, in fact, fiercely debated, especially in democratic nations. However, if historical trends persist, the use of OSS by government agencies will continue to increase [[10], see a detailed government business case for OSS in [23]].

9.2.5 Commercial Interest in OSS Projects

Contrary to the general perception that OSS projects are commissioned, managed and delivered by communities of altruistic agents, commercial entities, and their employees, play a

substantial role in OSS development [10]. Large firms such as IBM and Hewlett Packard are dominant contributors to OSS efforts such as Linux and Apache [10 44]. The landscape of enterprise software comprises a complicated business environment, served by networks of specialized suppliers, that enables a multitude of customers ranging from individual private consumers to corporate clients. Furthermore, computerized systems are aggregations of various interdependent products. End user applications depend upon well planned hardware architectures as well as intermediate software applications, such as operating systems, that control hardware elements, manage network protocols and secure systems from unauthorized access and use [17 20 9]. As a result, providing value to end users depends upon layers of disparate products and services that must operate in concert [23 13]. Often these products are generated or manufactured by a variety of organizations. Many of these products are integrated through formal or informal diversity strategies undertaken by product suppliers. The well-described relationship between IBM, Microsoft and Intel provides a straightforward example of cooperative advantage within the computer industry [35 15]. IBM's licensed PC architecture preferably uses Intel processors and Microsoft's operating system. PC Models using IBM-licensed architectures generally deliver systems to users as a bundled package having Intel microprocessors and Windows preloaded. Both Intel and Microsoft utilize their own branding schemes as part of the package. Although the consumer may be aware of the distinctions between the various components of their PC, the system that they purchase is an integrated set of software and hardware that, from the perspective of the customer, functions as a single unit [14]. The integration of IBM/Microsoft/Intel products imparts substantial marketing advantage for the participating companies and encourages steps to ensure seamless integration of these complementary products. The Windows operating system serves as a platform for customization through the selection of user software that meets the individual needs of customers. The cooperative situation has been hypothesized to be mutually advantageous for driving advances in technologies as, for example, new generations of software demand higher performing PC architectures that, in turn, demand higher performing processors designed to interoperate with the next generation PC architecture [[14] and this author]. Moreover, the domination of the Microsoft IBM PC model drives further adoption of other Microsoft offerings, including Microsoft-branded end user software (such as MS Office) and software development environments (such as C# and .Net). The Microsoft consumer software offerings provide real or perceived advantages given that the platform vendor would be expected to best optimize consumer software for their own operating platform (the author neither supports nor refutes the reality of this specific proposition).

Of note is the so-called LAMP stack, a highly popular open source bundle comprised of the Linux operating system, the Apache web server, the MySQL database and the Perl (or PHP or Python) software language [http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29]. The LAMP stack provides a complete software distribution framework and opens the possibility of OSS benefiting from cooperative advantage.

Cooperative integration creates, or should create, barriers for competing products, such as

alternative operating systems. The open source *Linux* operating system is compatible with the IBM PC platform but serves only a niche market of PC customers, typically programmers or advanced computer users who are attracted to this alternative offering due to low cost, an affinity for the OSS application itself or to make use of Linux-specific programs. Switching costs prevent the majority of users from considering an alternative operating system such as Linux. Not only would the Linux interface be unfamiliar, but the user would likely need to adopt replacement applications that may require retraining. Certain critical applications may not have Linux equivalents and file structures may not be compatible between Windows and Linux versions. Due to entry barriers and switching costs, the power positions of cooperative partners can become biased. Microsoft, due to success as a customer facing enterprise became highly powerful relative to its collaborators [35].

Although displacing Windows from the PC market has proven difficult, back office hardware (*server*) systems were more welcoming of alternative operating systems [23 35]. Back office users are typically computing professionals who are expected to be more amenable to adopting new technologies with many interested in the features of Unix-based operating systems such as Linux. IBM invested heavily in Linux, this, as well as investments in open source programs, such as the java-based open source Integrative Development Environment *Eclipse* (a software development productivity tool set competing with Microsoft Visual Studio, http://eclipse.org/), presumably, to undermine Microsoft's strategic positions in operating systems and programming languages [35 31].

Alternatively, Bessen notes that many firms participating in OSS projects appear not to expect competitive advantage as a result [10]. Additionally, firms contributing to OSS business software often have commercial options offered by non-rivals. Assuming that competing proprietary software can be provided at a cost less than the OSS investment, given that marginal software costs tend to be small, there seems to be little economic sense to justify a firm's OSS participation [10].

9.2.6 Complexity of OSS

Bessen proposes that OSS software is a *complex* public good. Specifically, OSS products often provide a wide range of capabilities that promote adoption by diverse user populations for a variety of purposes [10]. OSS provides a framework for increasing functionality in ways that are not necessarily envisioned by the original contributors. Bessen suggests that in addition to the OSS product itself, any OSS extensions to the product are in fact, themselves, innovations by nature. The evolutionary branching of OSS functionality, in Bessen's opinion, would be greatly curtailed in an environment constrained by intellectual property contracts [10]. Furthermore, Bessen hypothesizes that OSS software, and the specific features associated with enhancements would supplement, rather than compete, with proprietary software [10]. As such, the innovative nature of software development leads commercial entities to leverage OSS alternatives for *tailored* functionality even though the low marginal costs that are generally associated with commercial software would appear to make such activities inefficient [10]. Moreover, Bessen

proposes that OSS provides opportunities beyond typical mechanisms for decreasing transaction costs, such as bundling software applications or capabilities (i.e. the Microsoft Office Suite) or the creation of Application Programming Interfaces (APIs) that expose application functionality for use by other, external, programs [10]. With APIs, consumers having the requisite skills can create new software capabilities by writing their own programs that leverage existing features in the embedded software without changing the embedded software itself [10]. Due to complexity, post-purchase maintenance activities such as defect correction and business process changes to accommodate often rigid business work-flows imposed by the software, become a dominant factor of total cost of ownership. The propensity of firms to develop custom software reflects the difficulty in fitting commercial off the shelf (COTS) software to firm-centric specialty business processes. However, modification of OSS code allows companies to directly add features of interest or to integrate OSS with other value-added products to provide the requisite functionality [10]. As Bessen notes, by 2002, the Apache web server hosted over 60% of active web sites and approximately 50% of commercial firms using Apache had either modified the code (19%) or integrated third party products (33%) [10]. These customizations, many of which were provided back to the Apache community, led to a startling rate of capability advancement relative to commercial software alternatives [10]. However, as most were customizations, Bessen argues that these activities, by and large, increased the *white space* of software product opportunity as opposed to limiting commercial software markets [10].

The author wishes to note potential contrary opinion, for example, substantial use of Linux in corporate environments is undermining, not supplementing, investment in commercial server operating systems [35] [23]. Moreover, Bessen's own data regarding corporate adoption of Apache appears indicative of OSS/commercial competition [10]. Regardless of motivation, it is clear that commercial entities have more than a passing interest in OSS.

9.2.7 Transaction Costs and OSS Free-Riders (the Benkler Proposition)

Yochai Benkler [9] proposed an economic explanation for the apparent success of OSS projects. Benkler was inspired by Ronald Coase's classic insight that assembling dependent market processes into the controlled environment of a firm reduces transaction costs relative to free markets [9]. Benkler reasoned that if OSS projects are able to reduce the feature development costs (i.e. the software equivalent of transaction costs) below that required by corresponding commercial implementation then OSS development would be more efficient relative to commercial development. Software development transaction costs are associated with the Software Development Life Cycle (SDLC) and, as noted above, it is difficult to envision how the typical SDLC could progress in a minimally controlled environment [31].

Open projects such as the Mars Public Mapping Project, in which surface images of the planet Mars were annotated by public (presumed non-scientific) volunteers, were facilitated by careful decomposition of the surface into finite image areas that could be annotated by the

typical volunteer in approximately ten minutes [9]. Individual participants could generate as many annotations as they wanted. However, the low cost, in contributor time, required to produce a single unit of project value (i.e. the annotations associated with a single image region) encouraged widespread individual contribution. Essentially, the Mars Public Mapping Project decreased the transaction costs of annotation through modularization of the image surface into distinct regions that could be mapped in isolation. Each distinct planetary area was annotated many times with the results of the multiple reviews harmonized, or integrated, simply and cheaply via software. Benkler realized that modular software architectures that permit distinct software features to be delivered in isolation would promote individual contribution and facilitate software integration. If the software framework was appropriately modularized, the transaction costs could be reduced enough to compete with commercial SDLCs [9].

Given a population of OSS developers sized comparably to a discrete set of isolated modules, that are small enough to be economically developed by a single individual and that implement the desired software capabilities, the OSS developers could create the software project with limited central control and also tolerate free riders who would use the software but not contribute to its development [9]. In this context, OSS free riders do not include end users who lack software development skills (although such people may contribute by testing OSS software builds or by providing other valuable services such as user documentation).

Benkler considered the distinction between *"click worker"* efforts such as the Mars Mapping Project and OSS development. Specifically, there will be an expected diversity of effort, in terms of difficulty/time, required to complete the coding of individual modules. Benkler conjectured that diversity in the "option value" (difficulty or amount of effort relative to value) of modules would have an impact on the OSS project participation [9].

9.2.8 Game Theory and Open Source Participation

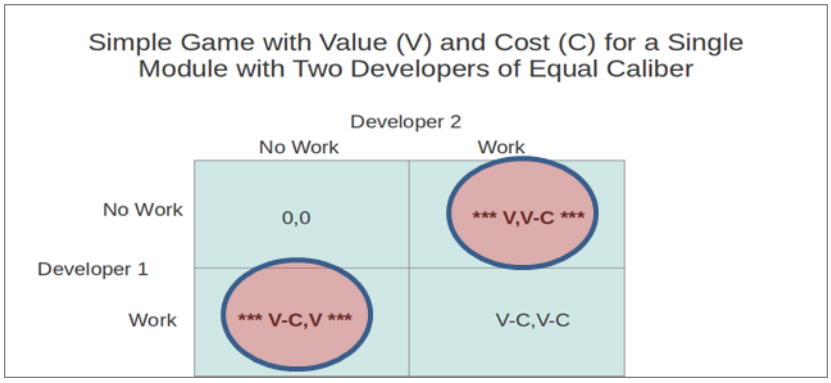
Although interesting speculation, Benkler's hypothesis lacked quantitative support. Carliss Baldwin and Kim Clark, who have extensively researched concepts of modularity, expanded Benkler's supposition through the use of involuntary altruistic game theory [6]. To model an OSS scenario, the game participants are given to be OSS developers having some measure of capability, or caliber of skills. These developers can either elect to participate in (i.e. "*work"*) or abstain from (i.e. "no work") developing a given OSS module having some *"value"* (can be modeled as the relative software value of the module relative to the cost required to develop the module). OSS developers have full knowledge of the existence of each other as well as the existence and value of a particular module. However, OSS developers cannot interfere or restrict each other's efforts. Additionally, an individual developer cannot determine whether other developers have decided to work on or abstain from the development of individual modules [5 6].

The following basic game describes economic impediments to OSS development. Two developers of equal caliber contemplate developing the same module. The module value (v) is assumed to be greater than the developer cost to produce the module. The cost to produce any module is always assumed to be greater than zero.

Constraints:

1. Two developers want the same OSS capability (or value *v*)
2. Either developer can create v at cost *c* (equal caliber). Development (or *work*) = v-c assuming that v > c and c>0
3. Each developer has full knowledge of the potential value v
4. Neither can restrict the other's effort

Figure 1: Simple game with two developers of equal caliber and Nash Equilibria circled. The highest value is obtained by the free rider when one developer works, the inference being that developers are incentivized to free ride.

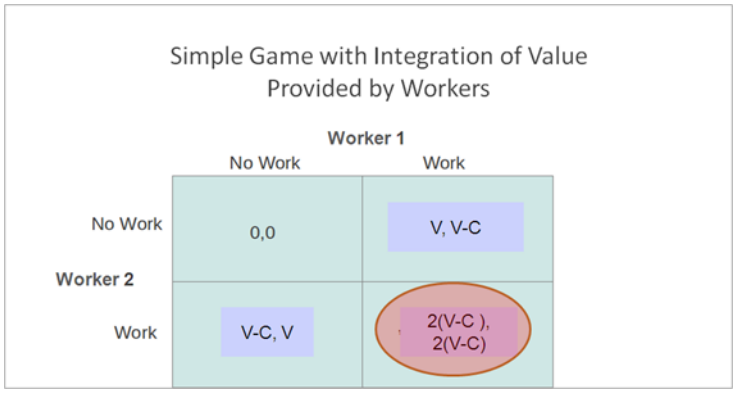


In the basic game, it is obvious that no value is generated if neither developer decides to work. However, as both developers work in isolation, their contribution is redundant should both elect to work. It follows that there are two *Nash Equilibria* (game states in which neither participant can unilaterally enhance their position) that represent the most efficient solutions in which one developer works and the corresponding developer elects not to work. In these cases, the non-working developer, as a free rider, gains the most value. Theoretically, free riders will benefit most in the basic OSS scenario, which is indicative of the general problem facing economists trying to explain the phenomenon of OSS participation [6].

The basic game also highlights an important distinguishing characteristic between click workers and OSS developers. Software is fundamentally non redundant as a single set of code, such as the module described in the basic game, when loaded by an operating system, can be executed any number of times by client programs (other dependent software modules). Therefore, two developers producing modules having the same functionality will be redundant and one of the modules will be selected for integration and one will be discarded. This competition drives the benefit enjoyed by the free riders.

Alternatively, in the click worker scenario (Mars Mapping), contributions are summarized such that each submission by a click worker contributes to the final outcome. For the Martian annotations, multiple independent annotations for the same image region are all integrated by software processes to produce a final annotation that represents the collective effort. For the click worker scenario, the Nash Equilibrium corresponds to the situation in which both participants elect to work as their respective contributions (values) will be summed in a manner that creates more value than each participant offers independently. There is no competition in the click worker scenario as opposed to the OSS scenario where wasted effort can result from the lack of centralized control over assignment of software modules.

Figure 2: The simple game with integrated value (click worker scenario with additive value/cost). The Nash Equilibrium is circled. In this case, both workers are incentivized to work as each contributes to a greater total value than either can provide alone.

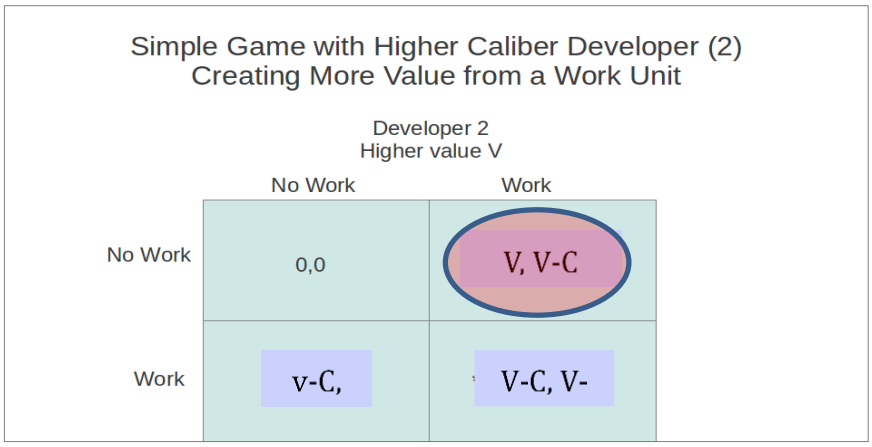


The basic OSS case is somewhat contrived as code submissions would generally be differentiated by various factors in addition to implemented features such as reliability, performance, code quality and developer productivity (*the productivity of software developers has been determined to have the greatest range of all major professions, including physicians!*) [38]. The basic OSS scenario can be extended by introducing heterogeneity of talent between the two potential OSS developers. In this case, given the same cost "C" (time for example) there is differential value created (small value "v" and high value "V") by each developer.

Constraints:

1. Two developers want the same OSS capability (or value V)
2. Developer 1 (higher caliber) can create value "V" at cost "c"
3. Developer 2 (lower caliber) can create a smaller value "v" at an equivalent cost "c"
4. Development (or *work*) = V-c or v-c assuming that V>v and v > c and c>0
5. Each developer has full knowledge of the potential value V
6. Neither can restrict the other's effort

Figure 3: A scenario of unequal developer caliber and unequal module value. In this case, the maximum value is generated by matching the caliber of developer to module value.



Regardless of the factors that differentiate the superior value provided by Developer 2, Developer 2's module will be chosen for integration while Developer 1's alternative lower value module will be discarded. In general, when multiple developers compete in the delivery of OSS modules, the contribution from the higher caliber developer will be expected to become part of the OSS project. If the developers are aware of their relative calibers with respect to the developer pool the lower caliber developers are likely to elect to be free riders and assume the value attributed by the higher caliber developers and incur no cost. If the differential of developer caliber is unknown at the onset, it would likely be discovered over time by each developer depending on whether their contributions are accepted for integration [6].

Assuming heterogeneity of talent, contributions are expected from the more talented developers while less talented developers would be expected to become free riders. Free riders retain the highest value in this model. Should there be information regarding the caliber of participating developers, lower caliber developers would be expected to refrain from participation. In the case of a surplus of potential developers the free riders are likely not to detract from the value of an OSS project. However, in a smaller developer pool in which all modules are of equal value the loss of the lower caliber developers to the frustration of potential competition would be expected to slow OSS development. Again, the inability to centrally assign resources would be expected to have a deleterious effect on OSS development.

Baldwin and Clark introduced a probabilistic element to their models [6].

$$\text{Probability of working } (p) = 1\text{-cost}(c)/\text{value}(v)$$

In this case, higher costs and lower values decrease the likelihood of working. This is consistent with the expectation that high caliber developers (lower cost, higher value) will be more likely to work relative to lower caliber OSS participants. However, in the case of click workers, such as the Mars annotators, caliber differences might be associated with the time required to produces a quality annotation (e.g. the fastest annotators can complete in 5 minutes vs. the slowest in 15 minutes). The outcome associated with discrepancies in Mars click worker

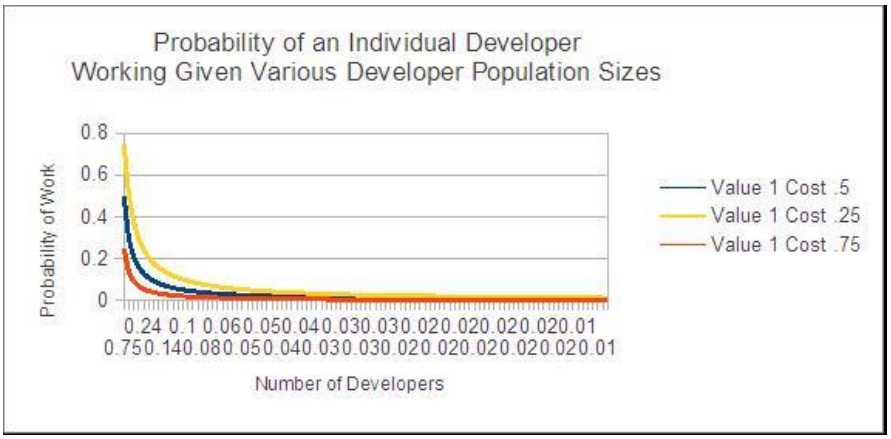
productivity might become manifest as a positive correlation between increasing click worker speed and increasing number of regions annotated per click worker. However, the probability of *participation* by any single Mars click worker may not be well represented by this probabilistic model.

Baldwin and Clark extended the probabilistic model to multiple developers as follows [6].

Probability of working (p) given a pool of "N" developers = 1-cost(c)/value(v) 1- (c/v)^(1/N-1)

This extension leads to a negative impact of increasing pool size on OSS contribution. This relationship is in keeping with the expectation that competition leads to free ridership (i.e. more developers, increasing competition for modules, increasing the amount of "wasted" work, decreasing participation) [6].

Figure 4: The chart demonstrates the impact of Baldwin and Clark's equation for probability of working at increasing number of developers at various probabilities.



As is expected from Benkler's proposition, for a given developer pool (N), only one developer need work for all N developers to attain v. However, when extending the probabilistic model to a larger pool (N) of developers using N = 1- (c/v)^(1/N-1) [from 1], plotted in Figure 4 for 2-100 developers, the probability of individual work decreases with larger N.

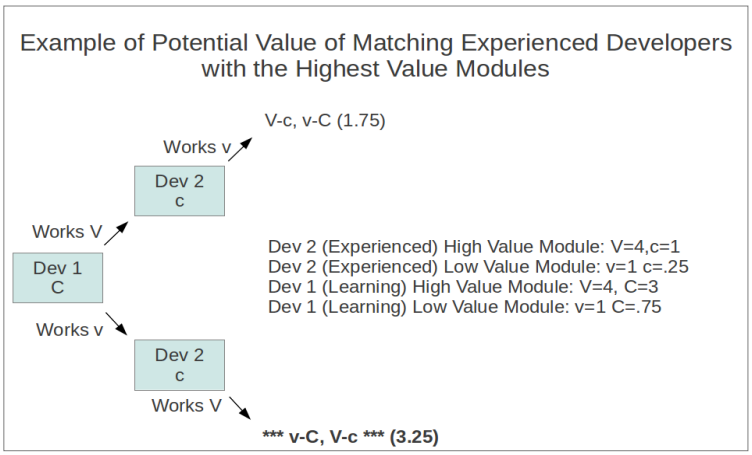
Baldwin and Clark next simulated the probabilistic model with various numbers of developers and modules over successive time periods. The simulations demonstrated that matching the number of developers with the number of modules maximizes participation. Furthermore, there is a negative relationship between free riders and the number of modules with free riders decreasing with increasing numbers of modules assuming the number of developers is fixed. These results are not unexpected given the impact of competition in the two-player games above. A perfect sub-game in which each developer is assigned to a distinct module is a logical

extension of the basic games although the outcome is unlikely in the absence of centralized planning or, as we shall see, variations in developer caliber and module value [6].

The final element which Baldwin and Clark added to the simulation is the assignment of values to modules according to a normal distribution. The variance of the distribution across the modules is referred to as the overall *option value* of the system [5 6]. The simulation now includes multiple developers of heterogeneous caliber and multiple modules having heterogeneous values. In a one time period game under these conditions, the authors demonstrated a decrease of free riders with *either* an increase of modules (consistent with the prior experiment) or an increase in option value [5].

From these results, the authors propose that variance in option value could lead to a more efficient distribution of developers to modules. Assuming some level of self-awareness on the part of the developer, each developer will be likely to align themselves with a module that is commensurate, in value, with their capabilities. The highest caliber developers will be likely to select the highest valued modules and vice versa in the range of module value [6]. Figure 5 shows a hypothetical example (module value $V >$ module value v, developer Cost $C >$ developer cost c). The sub game optimum is achieved when the lower skilled developer (C) works on the lower value module (v) and vice versa.

Figure 5: An example scenario that demonstrates the advantage of matching the relative caliber of developers with the relative value of modules.



It should be noted that the element of time does not factor into the model [6]. This obvious simplification could detract from the model given that software quality may be associated with time of effort in actual implementations (this author's note). Of course, potential interdependency between modules would force a natural scheduling of module contribution that is not included in the model (this author's note).

The combination of option value and developer caliber is a potential force to promote OSS contribution in the absence of formal centralized planning. If indeed relevant, a basic requirement for an OSS collective would be the creation of a modular framework during the early iterations of OSS product development. Moreover, a product roadmap (whether evolutionary or preplanned) that includes implementation diversity/difficulty with regards to future modules would appear conducive to engaging an active and productive community of developers having a broad range of capabilities. Fundamentally, Baldwin and Clark have provided quantitative theoretical support to Benkler's intuition regarding the potential of the combination of modularity and option value to lower transaction costs of OSS.

The key game outcomes that arise from Baldwin's and Clark's research are summarized [6].

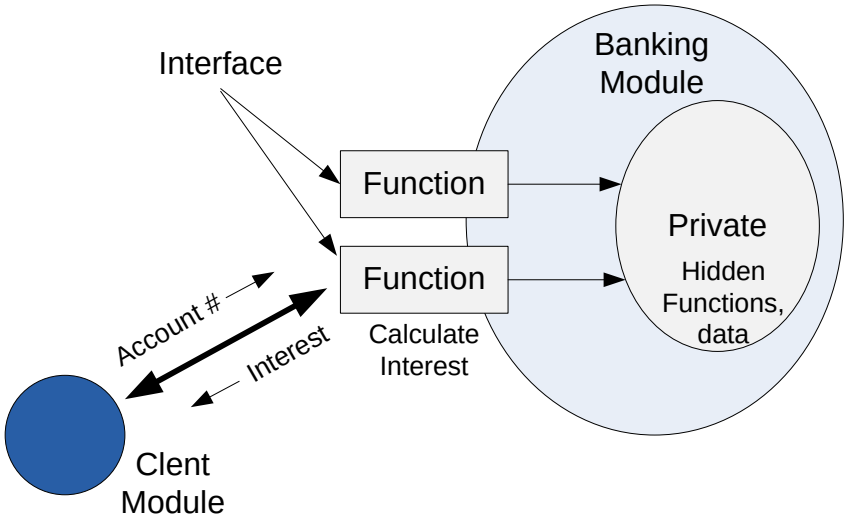
1. The probability of free-riders increases with increasing number of developers
2. Increasing modularity leads to increased participation
3. Increasing option value leads to increased participation
4. Matching the relative to the caliber of the developer to the option value of the module promotes participation

9.2.9 Conceptual Foundations of Modularity

Although modularity is recognized as a central characteristic of well-developed software, it is important to reaffirm the definition and the generally accepted value associated with modularity as a programming best practice. The benefits of modularity became research topics in the late 1960's and early 1970s [29]. A module can be defined as a grouping of related interdependent tasks having well established inputs and outputs. The inputs and outputs are designed to eliminate ambiguity associated with interactions between multiple modules. In principle, modules should be stand-alone sources of key capabilities, and as such, can be implemented and evaluated in a disjoint manner. Well designed modular systems are expected to be more resilient to system-wide errors, with such errors more easily identified, traced to source and corrected[29]. Additionally, modules should be replaceable without perturbing the system provided that strict adherence to input and output specifications are maintained [29].

Parnas noted that modularization is associated with design decisions *"which must be made before the work on independent modules can begin".* Modularity results from a process of functional decomposition, in which the needs of a system are deconstructed into components that are distinct and constrained enough to be unambiguously understood [29]. Parnas recommended that module design should be guided by the most difficult or malleable aspects of the systems with modules incorporating and "hiding" the implementation of these key aspect from other modules. The concepts of *encapsulation* and *information hidin*g are now standard practices for modular-based software development methodologies such as *Object Oriented Programming* [17 20]. Of course, OSS (by *constitution*) guarantees access to all parts of the source code such that there are no restrictions from reading and reviewing any component of the code [25]. However, the software operating framework can apply access restrictions during program execution.

Figure 6: A simple example of a software module



As a simple illustration, consider a possible scheme for calculating bank account interest. A client module needs to know the interest on a bank account. The client *"calls"* the *"Calculate Interest"* function (or method) on the *"Banking Module's" public interface*, passing the banking module an account number. The *public interface* is the only mechanism for the client to access the capabilities of the banking module. As a result of the client's call to calculate interest, the banking module returns the actual amount to the client. In this scheme, the client does not know the *how* the banking module actually calculated the interest. The client must trust that the banking module has accessed the correct account information and has provided the correct interest amount based on the bank's declared services to the account holder. The client can neither intervene in the interest calculation (such as by dictating a compounding model or interest rate) nor can the client determine the method the banking module has used to fulfill the request. The public interface function essentially provides a *"contract"* between the client module and banking module. The preconditions for determining interest must include the provision of an account number (presumably that conforms to the bank's specific account format, exists and is active). Post conditions would presumably include the format of the amount returned (for example, the interest will be in US dollars, to the hundredth decimal place rounded down to the nearest penny) and provide a mechanism to return an error (account number does not exist, incorrect account number format, etc.).

This simple example can be aligned with modularity benefits put forward by Langlois and Garazelli [25].

1. Limits the need for widespread communication among the modules (and their creators!)

   The public interface defines and simplifies communication between the client and banking modules

2. Economies of substitution: Replacing modules without impacting the system as a whole

The banking module could be replaced (e.g. with a new module having different interest schedules) without any impact to the client provided that the new banking module exposes an identical public interface

3. Specialization and the effective use of local knowledge for comparative advantage

   The client and banking modules could be developed by separate teams having specialized skills. For example, the client could be created by the bank's customer service team to be used as part of a customer web portal while the banking module could be created by the bank's operations team responsible for managing and monitoring account transactions.

4. The more the system benefits from rapid trial-and-error learning

   With defined interfaces, simulated modules can be created for initial development. Based on the public interface of the banking module, the client team could create a *simulated* banking module that returns default interest values and error messages. This approach would allow the client team to develop and test software independently without needing immediate access to the resources associated with the banking module.

5. Coordination is imbedded or institutionalized in the structure of the system

   The public interface of the banking module enforces conformance to bank policy by ensuring that the banking module retains complete control over transaction processing

Although trivial, the simple example of modular interaction described above is indicative of the potential benefits associated with modular design. Moreover, it becomes elementary to juxtapose the described benefits of modularity with Benkler's proposition that modularity fosters OSS participation [9]. Enabling independent action, specialization, trial and error approaches, facilitated substitution would *appear* to promote the type of productivity observed in successful open source development communities. However, the aspect of coordination and enforcement of institutionalized structures as a benefit of modularity is difficult to reconcile with Benkler's model, and subsequently Baldwin and Clark's quantitative assessment. Clearly, Benkler considered the integration of contributions a critical component to the success of volunteer communities [9]. The economics of coordination is easily demonstrable in the click worker scenario in which integration processes can be manifest in computer programs written by a central authority and in which click worker contributions are congruous. Per Langlois and Garazelli, the advantage gained from such communities is an unmistakable modern twist of Adam Smith's division of labor evident in the top down coordination and division of repeatable activities [457]. Although the author in no way intends to disparage the great ingenuity, efficiency and value associated with click worker projects, the basic economic premise has long been well understood and utilized [per Adam Smith, Wealth of Nations]. However, the complications associated with organizing the development of interrelated collections of unique system capabilities, derived from coordinated interactions of purpose-built subcomponents created by

independent volunteers possessing distinct specialties (OSS communities) seem daunting and worthy of exploration.

9.2.10 Implications of Modular Design

Regardless of the theoretical and practical advantages of modular software design, there are many ramifications in electing to architect systems based directly upon the outcomes of analysis by functional decomposition. The suitability of modular software design is not universally accepted. Brooks' Law [11] infers a general inverse relationship between the number of software developers and the speed of development. This relationship, demonstrated during the development of the IBM 360 Operating System, was inferred to be due to the exponential costs of coordinating personnel efforts on a highly interconnected project that, consequently, nullifies any advantage from division of labor [25]. Brooks' system was non decomposable and required developers to understand the breadth of its design and operations. Brooks and Parnas debated the merits of encapsulation and information hiding for decades as non-decomposable (integral) and decomposable approaches to software architecture carry reciprocal trade-offs.

Establishing modular systems requires an initial fixed cost as the design rules (interfaces and modular structure) must be pre-developed and accepted. These design costs can be substantial, as can be the process of communicating the resulting design to enable participation [25]. Modular systems are more likely to be prone to performance issues than corresponding integral systems should various modules require coordinated optimization in order to maximize performance [25]. Although modular design imparts substitution efficiency associated with the facility in replacing modules, *systemic* design changes require recreation of the modular design rules. Changes to modular design rules lead to re-incurring the initial fixed costs of modular systems. Moreover, if the modular system supports many clients, each client is also likely to require a corresponding fixed-price design phase to rationalize the changes to the design rules of the core system [25].

Langlois and Garazelli introduce the concepts of *autonomous* and *systemic* innovation. Autonomous innovation describes system modifications having local impact, such as changes to the "hidden" code of software modules that do not violate established design rules. Conversely, systemic innovation refers to the need for simultaneous dependent modifications across the system which, for modular systems, are expected to mandate new design rules. Modular systems foster autonomous innovation while integral systems foster systemic innovation [25].

Moreover, *speed* of implementation may favor integral systems as the specification and socialization of design rules may lead to unacceptable delays. Design rules that are generic and, thus, potentially reusable would remediate the initial fixed costs associated with modular designs [25].

Although only a brief examination of the essence of modular design concepts, the necessity for

substantial up-front fixed-price coordination associated with modular systems challenges the Benkler proposition that modularity coupled with the efficient self-assembly of suitable matches between modules and developers is the core driver (*secret sauce*) of OSS. Moreover, if modularity is indeed prerequisite for successful OSS projects, there should be types of software development initiatives, such as those requiring systemic innovation to support emerging performance requirements that would be unsuited to an OSS approach. (It would be interesting to determine whether the differentiation of autonomous and systemic innovation is at all correlated with Bessen's proposed *white space* of OSS solutions).

9.2.11 Expanding the Model of the Volunteer Community

Langlois' and Garazelli's summarization of the coordination challenges associated with modular design resolved to an expanded characterization of the OSS community model. Models for exchanging *products* and *effort* with respect to whether providers and consumers self-identify their intentions were contrasted. *Products* are considered those market contributions that are easily measured or priced while *effort* refers to contributions that are provided by individuals and generate value via cooperation. As such, measuring and pricing effort is nontrivial [25].

The classic market serves people who *self-identify* (in the absence of coordination) their products to consumers. The classic (Coasian) firm identifies people having requisite skills and coordinates their *efforts* to meet cooperative value objectives. Contract or outsourcing firms coordinate personnel to deliver discrete products or services. Call services and other forms of division of labor are categorized as outsourcing and, it follows that, click worker initiatives would conform to the outsourcing model [25].

The model of *Voluntary Production* was proposed as a combination of self-identified contributions and effort and is inclusive of open science and OSS projects. However, Langlois and Garazelli noted that voluntary production invariably occurs within the context of some preexisting underlying element of structure or coordination, whether technological, research driven or managerial [25]. The result is a *"hybrid"* model of output in which people self-identify their contributions based on their skills and interest while agreeing to conform to, at least some, element of structured cooperation. An additional value associated with self-selecting groups of contributors is the potential emergence of *"collective intelligence"*. That the emergence of creative ideas is associated with intellectual diversity is well documented [22]. If, indeed, voluntary production initiatives lead to cognitive diversity as a result of self-assembly, then such projects would enjoy advantages in innovation and problem solving.

The Benkler proposition has now been expanded to include a necessary element of coordination. Although the spontaneous self-assembly of OSS project teams seems implausible, modular architecture coupled with comparative advantage through self-identification *could* reduce the need for project structure enough to foster voluntary production. The potential collective intelligence that emerges *might* explain the perception that open source products are of higher

quality [25] [22].

9.2.12 Design Patterns and Modularity

Inspired by the work of architect Christopher Alexander [3], the software community has established design frameworks to guide the development of common recurrent software system features and capabilities[17] [20]. Alexander and his collaborators dissected and cataloged the salient features leading to successful architectural implementations, ranging from the application of basic architectural components, such as design and placement of windows, through large scale city plans. This architectural *"pattern language"* could then be used as a basis for future architectural design [3]. As a result, patterns-based construction products, such as buildings, courtyards and neighborhoods, could be creative and novel in form and execution while still benefitting from conformance to pre-established product-specific success criteria [3].

Originally explored by Kent Beck and Ward Cunningham in 1987 [20], the creation of software development pattern languages was carried forward to groundbreaking effect in the 1990's [17]. Patterns based-solutions now exist for standard software design dilemmas relevant to applications ranging from single-user workstation programs to multi-tiered distributed software environments [20]. Software patterns are typically presented with a context that describes the pattern's implementation and explains the pattern's value in solving a specific design challenge or set of design challenges. The pattern's implementation is defined using modeling constructs, such as diagrams, and a coded example is usually provided. Pattern documentation typically concludes by listing supplemental related patterns such as those that are commonly used in concert with, or are derivatives of, the pattern described [3] [17] [20].

Software design patterns, by their nature as established professional competencies, are intended to promote reusable code structures, provide elements of direct organization and coordination that embody, at least partly, the set of design rules required by modular projects [17] [20]. As such, the use of design patterns would serve to reduce the initial fixed cost associated with functional decomposition. Furthermore, patterns that address certain performance challenges could be leveraged proactively by teams planning modular software systems. The use of software design patterns, now well entrenched in the discipline of software engineering, could facilitate the development of the modular designs that, as detailed prior, are implicated as OSS success factors by a variety of investigators [9] [6] [31].

9.2.13 Empirical Evidence Associated with OSS

Software design patterns, modularity and comparative advantage have been proposed as elements that promote the economic feasibility of voluntary production of software relative to corresponding closed source/commercial initiatives. Collective intelligence could, potentially, provide the component of added value that accounts for the perception that open source projects exceed the quality of similar commercial offerings. Given the substantial research put forth to create a cogent and representative theoretical basis for software voluntary production, it is

important to review relevant empirical support for these theories.

At time that Benkler was developing the concept of modularity as a foundation for OSS, Stamelos et. al. were analyzing OSS code quality to quantitatively determine whether quality is, in general, distinguishable between OSS and commercial solutions [34]. Stamelos et. al. describe the OSS process as a rapid evolutionary approach based on the initial work of an individual, or small local core of developers, followed by the release and participation of a cascade of open source developers. Although there were many clear examples of large-scale OSS success (Apache, Linux etc.), however, the lack of an OSS "process" and metrics to quantify productivity and quality concerned many in the research community [34 10 23].

This case study [in[34]] employed a quality assessment software suite, Logiscope, (Telelogic, 2000) that automatically generated comprehensive code quality metrics, comparing these with user-defined programming standards. Moreover, Logiscope's programming standards were based on conclusions of an empirical analysis of millions of lines of industrial source code [34]. One hundred individual applications built upon the GNU/Linux open source operating system were evaluated in the case study [34] with the following metrics assessed.

1. **Number of statements (N_STMTS)**
    a. A count of the average number of executable statements per component [1–50].
2. **Cyclomatic complexity (VG): as defined by McCabe (1976)**
    a. A metric based on graph theory that represents the number of linearly independent paths in a connected graph. For the Linux assessment, this is a metric that represents the number of alternate flows of control for a tested component and is considered an indicator of the effort needed to understand and test the component [1–15].
3. **Maximum levels (MAX_LVLS)**
    a. Measures the maximum number of nestings in the control structure of a component. Excessive nesting reduces readability and testability of a component [1–5].
4. **Number of paths (N_PATHS)**
    a. The Counts of the mean number of non-cyclic paths per component. This is another indicator of the number of tests necessary to test a component [1–80].
5. **Number of unconditional jumps (UNCOND_J)**
    a. A count of the number of occurrences of "GOTO"-like statements that contradict the principles of structural programming for sequential control flow [0].
6. **Comment frequency (COM_R)**
    a. The proportion of comment lines **to** executable statements [0.2–1].
7. **Vocabulary frequency (VOC_F)**
    a. Defined by Halstead (1975) as the sum of the number of the unique operands, *n1, and operators, n2, necessary for the definition of the program.* This metric provides an alternative measurement of component size [1–4].
8. **Program length (PR_LGTH)**
    a. Measures the program length as the sum of the number of occurrences of the unique operands and operators. This metric provides also another view of

component size [3–350].

9. **Average size (AVG_S)**
    a. Measures the average statement size per component and is equal to PR_LGTH/N-STMTS [3–7].
10. **Number of inputs/outputs (N_IO)**
    a. A count of the number of input and exit nodes of a component with the equality of inputs and outputs an accepted practice associated with program quality.

Logiscope evaluated "*testability, simplicity, readability and self descriptiveness*" producing recommendations of "*accept, comment, inspect, test and rewrite"* and the authors developed a method to aggregate the results across components and applications that is normalized for specific programming languages [34]. In considering the results, two major interpretations are made; OSS code quality is better than would be expected given the limited control over development, however, the quality is lower relative to the commercial software evaluated a priori using Logiscope. The analysis also included expert opinion regarding the usability (user satisfaction) of the applications and the case study authors purport an inverse relationship between module size and usability. User satisfaction decreases with increasing module size, this is an expected outcome. However, there appears to be a maximum level of user dissatisfaction such that dissatisfaction is no longer sensitive to further increases in module size beyond some threshold length [34] (*this author's naïve interpretation is that possibly, at some point, the software simply becomes unusable*).

Stamelos et. al. recommends, as viable options, that the central coordinator enforce code standards and plan for strategic re-factoring efforts (i.e. deliberate quality re-engineering of existing code). That coding style can be emergent and self-regulated by individual programmers outside of coordination is considered, by the authors, likely not to be feasible [34].

Conversely, *Reasoning LLC* published findings comparing open source and corresponding commercial software based on Apache, Linux TCP/IP and MySQL (a popular open source database system) [32]. Reasoning tested for memory leaks, null pointer de-references, bad de-allocations, out of bounds array access and uninitiated variables using a proprietary code quality assessment tool. Reasoning's approach focused on coding issues that lead to explicit software execution problems rather than coding principles and best practices associated with the work of Stamelos and colleagues. As the assessment was proprietary, there is little to discuss in terms of methods other than the number of lines of code assessed between the open source and corresponding industry applications and whether the assessment was done on code early or late in the development life cycle [32].

Apache v2.1 Development release
Apache .53 defects/KSLoC* (~58 KSLoC)
Industry .535 defects/KSLoC (30000 KSLoC)

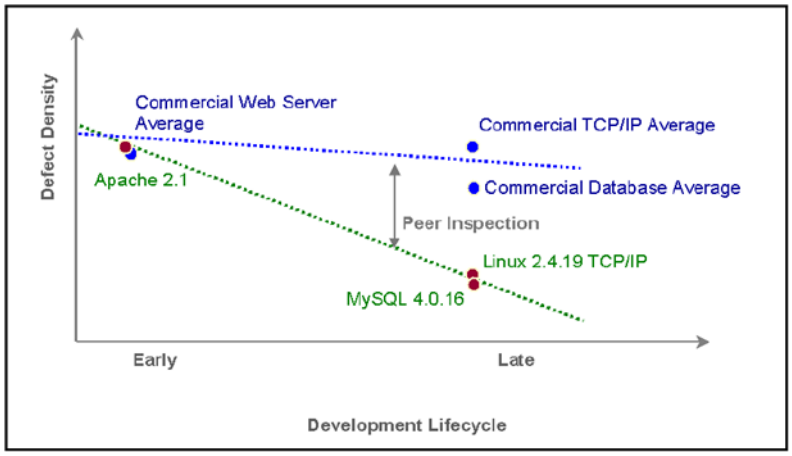
LINUX TCP/IP v2.4.19 Production
Linux 0.10 defects/KSLoC (~82 KSLoC)

Industry 0.25 defects/KSLoC (~22000 KSLoC)

MySQL v4.0.16
MySQL .09 defects/KSLoC (~235,667 KSLoC)
Industry defects .535 (35000 KSLoC)

****Tomcat**
297 defects/KSLoC (~127 KSLoC)

*KSLOC Kilo (1000s) Source Lines of Code
**Tomcat was assessed alone without comparison to a corresponding industry standard



Reasoning found that the defect density was smaller for MySQL and Linux TCP/IP, while Apache was similar in defect density, relative to corresponding commercial alternatives. Moreover, Reasoning notes that MySQL and Linux TCP/IP were assessed later in the development life cycle vs. Apache and that the outcomes could be indicative of quality enhancements relative to product maturity [32].

Although these OSS quality case studies are informative, the author notes the apparent dearth of definitive quantitative quality comparisons between OSS and commercial software packages. Both cases presented lack comprehensive representations of OSS products and the methods used by Reasoning are not available for scrutiny.

Additionally, perplexing is the aspect of modularity with Linux being an often-used case example for OSS. Linux is based on the Unix operating system developed in the 1970's by Bell Laboratories [23 28]. Unix is based on a "kernel" that provides core operational capabilities and interacts directly with hardware systems (file systems, registers, peripheral, security, connectivity, etc.). The capabilities of the kernel can be extended by programs written (through "Unix shell language") in a manner that leverages the Unix kernel's capabilities for value added operations. These program extensions could be easily shared and themselves modified or reused as the basis for further extensions. Bell Labs' potential claim for intellectual property rights over the Unix extensions prompted the creation of the open source extension libraries GNU and Berkeley BSD. These open source efforts focused on UNIX extensions rather than

creating an OSS Unix Kernel. With the Linux Operating System, Linus Torvalds contributed an open source Unix-like operating system kernel [23 28].

As noted by Narduzzo and Rossi, Linux is based on hierarchical design and fully modular designs were evaluated and dismissed as too risky and other modular designed contemporary operating system projects, such as Windows NT and HURD, were anecdotally understood to be poor performing and progressing more slowly than anticipated [28]. For Linux, Torvalds developed the hierarchical design philosophy and, after the initial open version was released to additional developers, Torvalds became responsible for code reviews as well as testing and, thus, provided the centralized locus of quality control for development [28 31].

If Linux is to be deemed weak as a case example for OSS in terms of modularity and decentralized control the Linux case may posit *extensibility* as another potential driver for OSS success. Extensibility is defined as the addition of features/capabilities to a software product without needing to modify, or otherwise impact, the existing software [1 4]. Shell script additions to the Unix kernel as well as the plug-in frameworks associated with many software packages, such as internet browsers and office productivity suites (such as Microsoft Office), are examples of extensibility. Henttonen et. al. described the importance of extensibility and, as noted in other contexts prior, integration potential to OSS. Extensible systems demonstrate ease of feature development and an element of loose coupling between feature additions that should, in principle, well support distributed development. However, extensibility as a driver of OSS success may be highly logical (and likely correct) [453] but the empirical support of this hypothesis, in the opinion of this author, is not readily apparent.

Although the apparent lack of empirical differentiation between OSS and closed source/commercial packages may be at first surprising, it is important to appreciate the challenges associated with quantitative evaluation. Wong, Kim and Dalton recently published a method, called "*CLIO"*, for detecting software modularity violations and tested the product on OSS projects Hadoop (a large-scale high-performance database application) and the Eclipse software development environment [36]. CLIO attempts to determine whether separate modules within a system must change together during development and, as a consequence, erode the benefits obtained by isolating code into distinct operating units [36]. CLIO resolves the system modularization, in that the system's modules are identified structurally, using a scoring technique developed by Baldwin and Clark [36]. As a separate action, the software revision histories are mined to determine legacy concurrent code modifications. CLIO then reconciles the modular architecture with the concurrent change log to determine whether multiple structural modules must change in concert during code modification. There is an inverse relationship between change dependency and modularity, the greater the change dependency between modules the less modular the system [36]. Assuming that CLIO can import the structural and legacy revisions and analyze the associations in real time (the analytical time and need for multiple attempts is not clear), the resulting violations are manually confirmed in the code by CLIO users. Additionally, revisions were checked manually to determine whether exposed violations were fixed or were identified to be fixed in a future software release. Moreover, it

seems unlikely that the revision and architecture information would be formatted for use by CLIO without manipulation [36]. As the architecture of a system may also change over time as modifications are realized, the development of a structural representation of the software to use for comparisons could be problematic. Additionally, use of CLIO assumes that architecture and change revisions have been documented in detail over the course of the software project and are readily available. Software projects having such comprehensive documentation are likely to be highly centrally coordinated with enforced disciplined management practices [author's opinion]. In a sense, the types of projects that are evaluable by CLIO may not represent the breadth of software projects that are of interest.

Quite simply, software quality evaluation is time consuming and open to interpretation. Although the source code is available for OSS projects, supplemental information pertinent to such evaluations, such architecture and change history, may not be complete or be very difficult to construct. New methods of quantifying code quality continue to emerge. There appears to be limited incentive, for commercial software vendors, to contribute their source code for such evaluation (at least evaluation leading to academic publication), as the interpretable outcomes could subsequently require product defense.

Moreover, those prosecuting comparison studies that definitively contrast and distinguish OSS processes with those of commercial software practices are likely to be, at least partly, impeded by the universal application of software best practices. Code standards, modularity, extensibility and design patterns are all techniques that are applied, and beneficial to, any software project. Applying measures of significance to comparisons between software initiatives that employ similar best practices may be further problematic.

Given the pragmatic issues associated with comparing software quality, it is not surprising that studies that definitively compare OSS with commercial (or otherwise closed source) software solutions, across a wide range of product types and implementations, appear elusive.

9.2.14 Alternative Points of OSS Comparison: Business Requirements for OSS

As noted above, Bessen hypothesized that OSS represents a *complex* public good in that open software can be readily tailored to meet the variable needs of individual developers [10]. With OSS, commercial software developers are able to refine software implementations to meet the specialized needs of the business processes within the firm for which these developers are employed. Bessen proposes that leveraging application programming interfaces (APIs) as a form of customization is less empowering than modifying the underlying source code. With APIs, capabilities are limited to those exposed by the developers of the API, which may not provide the capabilities required by API users, and proposed API modifications (and corresponding schedules) may be out of the control of the API's users [10].

However, circumventing programmatic interfaces (as Bessen purports to be valuable for customization) may be contrary to the notion of modularity expanded upon earlier [25]. If customizations require *systemic* [25] changes to the core software code the costs of altering the

design rules and integrating the changes would appear daunting and may be expected to fracture open source code bases. Therefore, the need for repeated *systemic* changes would be inconsistent with the notion of modularity as a key enabler of voluntary production in OSS. Customizations implemented by repeated *autonomous* [25] changes to the core code may be tolerable if these can be enabled by, for example, modular substitution.

The challenges are similar for extensible frameworks. Extensibility can be cast as an architectural enabler of both modularity and open customization. It has been noted prior that extensible frameworks provide such value for operating systems, internet browsers, spreadsheets and other applications that are built upon a core set of generic functionalities (e.g. hardware components, browser and worksheet models, mathematic algorithms) that are highly robust to change and can be perpetually *"decorated"* through a progression of new combinations of the existing base-level capabilities [21] [24]. Extensible frameworks are not applicable to all process scenarios and, even when applicable, their up-front specification, like that of any modular system, may be non-obvious. However, where extensible software frameworks exist, it could be argued that such a framework eliminates, to a large extent, the value of having the underlying core code provided as open source as the extensions are [ideally] completely decoupled from the core code base. The need to modify the underlying code of a purported extensible framework, to enable further extensions, is, in a strict sense, a contraindication to the definition of the framework as extensible [24]. Any *systemic* [25] modifications to the core code of an extensible OSS framework *to enable new extensions* could risk code base fracturing.

Overall, that code base customization is hypothesized to provide a driver for OSS adoption, more so than API development, is at best uncomfortable as such customization carries the risk of branching software versions that could be prove difficult to integrate in the absence of centralized management. However, Bessen's observations of the power of customization [in [10]] can lead to alternative insight into the nature of voluntary production of software.

An interesting observation regarding OSS applications that are prominent in the academic literature (e.g. *Apache* web server, *Linux* operating system, *MySQL* database, *Eclipse* integrated development environment, *Tomcat* enterprise software environment, etc.) is that these applications are, themselves, enablers of software development and deployment (http://projects.apache.org/indexes/alpha.html for the listing of >150 Apache applications pertinent to software development). Many of the most outstanding OSS successes are of primary value to software developers and, as such, those in need of, and those providing, OSS solutions are often one in the same. Bessen notes that by 2002 Apache hosted over 60% of the active web sites and roughly 50% of commercial firms using Apache had either modified the code (19%) or integrated third party Apache products (33%) [10]. The apparent ease in which software developers, using OSS, can specify their own software requirements that they, themselves, subsequently implement is worthy of attention. As a result, an investigation into the relevance of business analysis within the context of OSS is warranted.

The challenge of attaining a successful return on investment for software projects remains a serious and prominent issue for the software industry [4 13 19]. There are many studies that monitor software success incorporating a variety of metrics pertinent to the Software Development Life Cycle (SDLC). Post-SDLC maintenance and adoption challenges that include distribution, training and change management are critically important barriers to software success as end users must often transition from well understood and entrenched legacy processes and systems to new processes that are typically embodied in new software systems. Although software success has generally improved over the past two decades the level of success remains disturbingly low as reported by the 2009 revision of the infamous Standish Report [19].

Figure: Success of software projects taken from the 2009 Standish Report

Year	1994	1996	1998	2000	2002	2004	2009
Succeeded	16	27	26	28	34	29	32
Failed	31	40	28	23	15	18	24
Challenged	53	33	46	49	51	53	44

A 2003 Oxford University report (Saur and Cuthbertson) [in [19]] provided more dismal view noting only 16% of software projects as successful with 74% challenged and 10% abandoned. Jaques (2004) [in [19]] noted that software failure costs tens of billions of pounds sterling with a 22.6B pound spend and a 16% success rate. The National Institute of Standards and Technology estimate that defects cost $60B annually with 80% of development costs put towards identifying and correcting defects.

Tata consultancy (2007) [19] noted that:

- 62% of organizations experienced IT projects that failed to meet their schedules
- 49% suffered from budget overruns
- 47% had higher-than-expected maintenance costs
- 41% failed to deliver the expected business value and ROI
- 33% file to perform against expectations

In the same year, Sauer, Gemino and Reich published software product abandonment rates at 9% and over delivery at 7% (*Communications of the ACM* 2007). Moreover, three out of five Information Technology projects fail to deliver for the expected costs with 49% exceeding budget, 47% require higher than expected maintenance costs and 41% fail to deliver expected business value [19].

Echoing industry consensus, Robert Lawhorn has highlighted the following aspects of software failure related to business analysis [19].

1. *Poorly defined applications (miscommunication between business and IT) contribute to a 66% project failure rate, costing U.S. businesses at least $30 billion every year (Forrester Research)*
2. *60% – 80% of project failures can be attributed directly to poor requirements gathering, analysis, and management (Meta Group)*
3. *50% are rolled back out of production (Gartner)*
4. *40% of problems are found by end users (Gartner)*
5. *25% – 40% of all spending on projects is wasted as a result of re-work (Carnegie Mellon)*
6. *Up to 80% of budgets are consumed fixing self-inflicted problems (Dynamic Markets Limited 2007 Study)*

Further, per IAG consulting [19].

1. *Companies with poor business analysis capability will have three times as many project failures as successes.*
2. *68% of companies are more likely to have a marginal project or outright failure than a success due to the way they approach business analysis. In fact, 50% of this groups' projects were runaway which had any 2 of:*
    1. *taking over 180% of target time to deliver;*
    2. *consuming in excess of 160% of estimated budget;*
    3. *or delivering under 70% of the target required functionality.*
3. *Companies pay a premium of as much as 60% on time and budget when they use poor requirements practices on their projects.*
4. *Over 41% of the IT development budget for software, staff and external professional services will be consumed by poor requirements at the average company using average analysts versus the optimal organization.*
5. *The vast majority of projects surveyed did not utilize sufficient business analysis skill to consistently bring projects in on time and budget. The level of competency required is higher than that employed within projects for 70% of the companies surveyed.*

Some of the preeminent literature referenced in this paper, such as Langlois and Garazarelli [25], infers the difficulty in specifying software requirements.

"For large complex software artifacts it may be almost impossible to separate ex-ante all interdependencies, so unforeseen coupling between components at later stages (like for instance, integrating new and existing modules), may strongly affect the final outcome"

"The most common reason for failure is the emergence of some interdependencies which were left out at the beginning, at the time of architecture and interfaces definition."

Bessen's use of the term "complex" to describe OSS as a public good [10] is interesting as self-emergence is a property associated with complex systems [http://en.wikipedia.org/wiki/Complex_system]. It could be inferred that, for large-scale software products, requirements beyond those initially envisioned by the original developers are apt to naturally emerge and give rise to novel functional trajectories for the software. In cases of such emergent functional needs for software, predefinition of requirements would not be so much a challenge, it would be practically impossible.

Requirements elicitation is generally considered part of the first "scoping" step of the SDLC. In classic *waterfall* project management models [23] [31], in which projects progressed linearly through scoping, design, build and evaluation phases, large projects were hindered by "changing" requirements (i.e. new or modified requirements) that manifested throughout the design, build and evaluation phases of the SDLC and forced regressive work that lengthened project timelines and led to substantial uncertainty in project planning.

Given the software industry's acknowledgement that requirements elicitation is a core issue that plagues purposeful software delivery [19], it is surprising that the OSS literature appears to focus solely on the SLDC design and build phases when trying to reconcile the behavior of OSS communities and their individual participants. The aspect of scoping will be explored in the context of OSS.

9.2.15 The Nature of Requirements: Pattern Languages

In order to better appreciate the difficulty of requirements scoping for large-scale projects, the dialogue will more closely examine a proposed theoretical basis regarding the apparent malleability of requirements. The discussion will expand on the inspiration of Christopher Alexander, introduced prior, and Alexander's application of design patterns to problems of architecture.

Alexander received the first doctorate in Architecture conferred by Harvard University and his dissertation, copyright 1964, is available as a book entitled "*Notes on the Synthesis of Form*" [1]. The dissertation used the planning of a rural town in India as a case study for the development and prioritization of project requirements [1]. Alexander constructed a process that combined a simple requirements scoring system with a probabilistic algorithm to automatically generate a hierarchical representation of the project's requirements based on their relative association and priority. The intent of the computational process that Alexander developed was to deterministically limit the many alternatives inherent in approaching large-scale architecture projects by ensuring that high priority requirements were identified and that related, and presumably dependent, requirements were explicitly associated such that these could be addressed in a concerted manner [1].

Alexander published an influential article in 1966 ("A City is Not a Tree" [2]) in which, by the author's interpretation, the notion that requirements can be explicitly described a priori for complicated domains, and more explicitly, domains that are evolutionary, is called into question. Reviewing examples of the "great" cities, Alexander noted the apparent hierarchical structure in the way cities tended to develop districts to support specific interests such as those related to commercial, residential and recreational uses. However, on closer inspection of more granular derivations of such purpose-based districts, the healthy examples appeared to support a diversity of activities (residences within the business and recreational districts, business

populating residential areas, etc.). Although city plans appeared to be *rooted acyclic directed graphs* (i.e. a *hierarchy* or *tree*) from a high level (district) viewpoint, evaluation of the granular structure of districts yielded a *lattice* of diverse interactions. Essentially, healthy purpose-oriented districts are comprised of a myriad of granular city elements supporting a variety of activities that lead to a distinctive cultural experience. A paucity of granular interaction appeared associated with community failure [2].

Moreover, Alexander noted the influence of evolutionary development on successful communities. An interpretation being that top-down design is incapable of specifying the unique characteristics of successful communities, rather the unique cultural elements emerge as the community evolves over time [2].

Expanded to any large-scale project, such as software development, the interpretation is that a priori scoping will be challenged to specify the full breadth of project needs as new requirements will naturally (appear to) *"emerge"* as interactions become evident as the project proceeds and tangible products begin to materialize [2]. Predicting such interactions, while in principle not impossible, would require extraordinary foresight in predicting all of the ways in which individuals will engage, use and value a complicated product or environment.

Alexander and colleagues produced their seminal work, described above, in 1977, that detailed design patterns to address a multitude of architectural problems [3]. Design patterns provide foundational purpose-driven implementations based on prior successful realizations. Based on such prior productive results, well developed patterns naturally address the granular interactions that are difficult to predict and specify but are inherent to the problem under consideration. The application of patterns applies best practice to address scoping uncertainty.

As noted above, architectural patterns inspired software design patterns as a way of providing formal elements of coordination within the context of technical software development [17 20]. However, of great interest to this author is the exceedingly few, of Alexander's, patterns that are technical in nature [3]. Alexander's design patterns were fundamentally detailed to enhance the experience of the human user of the resulting architecture. Circumferences of columns and the widths of moldings promote a perception of strength and well-being for the human inhabitants rather than simply serving as structural support or protection from moisture damage. Designing public spaces toward the main entry of a home promotes human interaction while sustaining the necessary element of privacy for the human inhabitants. Architectural Design Patterns aim to prevent construction of structurally sufficient dwellings that are shunned by the human beings for which occupancy is intended [3]. It follows that the originators of software design patterns, in developing valuable technical methods for programmers, may have misinterpreted the nuance of Alexander's intent in directing the beneficial outcome of patterns to the end user, not the builder.

9.2.16 OSS and Evolutionary Development

Having developed a basis to explain the difficulties associated with project scoping activities in architecture, we can ascribe, as an analogy, the same impediments to software development when comparing voluntary production to commercial implementation. Essentially, interactions among requirements (i.e. the variety of ways end users might use, or want to use, a system) are difficult to specify up-front and, when surfaced, appear to result from poor requirements elicitation or poor understanding/articulation of requirements by the end users themselves. One aspect that is anticipated to correlate with success is the process of evolutionary development and a *bottom-up* approach characterized by the avoidance overdesign by creating working evaluable code early in production. This approach is directly embodied by *iterative* software development methodologies [8]. Iterative methodologies are an alternative to traditional *"waterfall"* project management methods [440] that execute the SLDC in a linear sequence with project requirements defined up front.

Iterative methodologies build software by introducing new features over time in distinct versions [23]. In principle, subsequent development efforts become driven by the experience end users have with the current software version. User feedback and new or modified requirements can be addressed in the subsequent version of the software [23]. These methodologies are highly advantageous for projects in which subsets of the fully envisioned capabilities are useful. For example, an application that intends to search a collection of, say ten, public databases may be valuable to a client as soon as one or two databases are made available (version 1). Value accrues with the additional databases such that each new database can be implemented incrementally and incorporated into a distinct version release of the software. Conversely, software supporting a complicated business process may require a comprehensive initial implementation to address the target process' end to end workflow before the software will be of value to any end user.

Additionally, there exist software development methodologies that incorporate iterative cycles of evolutionary development. *Agile* software development and project management methodologies focus on short term implementations of functionally incomplete, working code in order to constrain the SDLC into short sprints that repeat iteratively throughout the course of the project [18]. Agile development focuses on individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan and rapid iterative development [18]. Requirements that manifest during the execution of an *agile* project can be better accommodated within the sprint structure. Agile projects provide more flexibility in implementation at the cost of the perceived *certainty* in project scope and length [18] that is evident with project plans developed for waterfall methods that detail the project from start to finish.

The use of Agile methodologies for software development has been cited as a *major step forward* and could be linked to the relative increase in the number of successful software projects (2004 Standish Report).

It could be posited that OSS projects, by their very nature, progress in a more evolutionary manner. Certainly, modularity and extensibility would promote the evolution of new features. Although many OSS projects have a project roadmap of desired capabilities [31], the manner in which new participants join (and leave) the team over time, contributing those features that are of most personal interest, would further foster a development reality that is evolutionary in nature. However, as Agile approaches, and other iterative and prototyping methods, are well adopted across software industry, it is not apparent that evolutionary methods distinguish OSS from many large-scale commercial efforts. Clearly, commercial software providers are more incentivized to develop in a prescriptive manner in order to define and ascertain specific financial goals. However, empirical differentiation of OSS and commercial software based on evolutionary considerations may be a difficult matter to approach experimentally even if an investigator could obtain access to the methodologies of key commercial vendors.

As a final note on evolution, Raymond [31] notes the standard order project management tasks are designed for closed source projects:

1. To *define goals* and keep everybody pointed in the same direction
2. To *monitor* and make sure crucial details don't get skipped
3. To *motivate* people to do boring but necessary drudgework
4. To *organize* the deployment of people for best productivity
5. To *marshal resources* needed to sustain the project

Raymond notes that OSS functions operate in reverse order with goal definition being the last of the process. With overarching goals taking shape later, the project has had substantial time to incubate incremental modifications and improvements [31]. This reverse approach may model a viral tipping point where a certain finite level of achievement drives the project to substantial success [18]. Per Malcolm Gladwell [18], salespeople, mavens (technical experts) and connectors (highly connected persons) would be expected to populate the OSS community, driving the motivation to carry out the drudgework. Raymond describes Linus Torvalds (Linux originator) in terms that can be interpreted as Linus being an exceptional connector and maven [31].

Conversely, the high profile working group reviewing the failure of the ~$350M caBIG (Cancer Biomedical Information Grid), that was intended to provide OSS to integrate U.S. cancer centers, cited an overly constrained hierarchical chain of command type of organizational structure and limited engagement of key opinion leaders [12].

9.2.17 OSS and Software Use Patterns

The spreadsheet below lists the top ten OSS downloaded projects of all time (as of Oct 2012 [34]). All, with the possible exception of item eight, have similar predecessor applications (commercial or OSS).

Rank	Project Name	Downloads	Type
1	VLC media player	667,923,231	Plays DVDs
2	eMule	653,226,189	p2p file sharing
3	Azureus / Vuze	535,064,336	p2p bit torrent
4	Ares Galaxy	322,675,655	p2p client
5	7-Zip	283,850,645	archive utility
6	Smart package of Microsoft's core fonts	261,358,715	MS true type fonts for web development
7	FileZilla	199,327,335	FTP Client
8	PortableApps.com: Portable Software/USB	194,865,506	Launch apps from USB
9	MinGW - Minimalist GNU for Windows	145,923,636	Operating System
10	GTK+ and GIMP installers for Windows	138,037,263	Image Manipulation

Within the context of business use patterns, an interesting observation regarding OSS products is that many OSS deliveries mimic existing products that are already available. The table below includes examples of popular OSS products and their commercial equivalents. The Open Office spreadsheet application is practically indistinguishable from Microsoft Excel, GNU/Linux are strategic imitations of Unix, the same holds true for a great many popular applications (Author's aggregation).

Type	Open Source	Commercial predecessor
Office Suite	Open Office	Microsoft Office, Lotus 123, etc.
Operating System	GNU/Linux	Unix
Statistics	R	S-Plus
Web Browser	Firefox	Netscape, etc.
IDE	Eclipse	MS Visual Studio, JBuilder etc.
Database	MySQL	Oracle, Sybase, etc.
Database	posgreSQL	Oracle, Sybase, etc.
Web Server	Apache	MS IIS
App Server	JBoss	BEA WebLogic, IBM WebSphere
Music	LMMS	FL Studio
Music Studio	MusE/LASH	CakeWalk, etc.
Imaging	Open Microscopy	Aperio
Genetic Seq	GATK/SAMS	Cassava, Genomatics

The concept of repeated application of pre-existing successful use patterns as a driver of subsequent success for projects of similar scope has been detailed. As an example, a courtyard pattern that includes the requirement for multiple gates and a view beyond the courtyard area can be used for any future courtyard project [3]. Although the final designs of these patterns-based courtyards will be creatively distinct, all will incorporate expansive views and multiple gates in an effort to re-create the success of the courtyards that were used to derive the pattern.

Many open source products not only have pre-existing commercial products from which to derive patterns, the OSS projects often copy, to a high degree, the user interfaces and workflows.

Given the difficulties associated with adopting new software, faithful recreations of common interfaces would be a key enabler of user acceptance as the transition costs (in training for example) would be greatly reduced in replacing the commercial software with the corresponding open source software. Information Protection law generally does not provide rights for design elements such as graphical user interfaces [ref]. Therefore, OSS packages appear to have an enormous advantage in that the barrier to entry for developing productive user experience models, and training clients in the use of these implementations, can be practically nonexistent as the above OSS examples serve to illustrate.

9.2.18 Innovation, Evolution and the A Priori Existence of Imitable Products

Christensen [14] proposed the concept of disruptive innovation as a repeatable and predictable element associated with the demise of industry-dominant corporations as a consequence of the emergence of novel products from fringe competitors. Dominant industry players offering leading products and services excel in tailoring these products to meet the stated needs of their customers. Product development of this nature, or sustaining innovation, is predicated on customer requirements that arise during the product lifecycle. Leading suppliers are well positioned to receive and act on these requirements due to their close customer relationships and their willingness to invest in these opportunities, which typically lead to increased margins for differentiation and enhanced capabilities (i.e. up-market products). Sustaining innovations may be incremental or radical [14]. However, sustained innovation, by nature, is associated with progress along standard metrics of quality and performance. Therefore, sustaining improvements, regardless of implementation challenges, are based upon consumer desires that are well understood by the dominant suppliers and positioned as premium products and services. Moreover, standard financial and marketing methodologies are applicable for predicting return on investment in a sustaining scenario. Therefore, sustaining programs for established market leading products are highly likely to garner corporate commitment and resources [14].

Disruptive innovations, by contrast, address non-traditional elements of value [14]. The market for consuming disruptive advances is likely small, or undefined, upon product introduction. Standard financial and marketing techniques cannot be reliably applied to an undefined market. Furthermore, disruptive products are likely to be far inferior, at least initially, relative to the established products of major competitors. As such, disruptive ideas are unlikely to attract the attention and company resources of dominant market entities given the unpredictable, and initially feeble, expectations for performance relative to existing offerings [14].

The failure of incumbent players faced with disruptive innovation is explained by:

1. Superior product fit for a fringe market that, in time, overtakes and supersedes the prior established market.

2. Excessive development or performance (over specification) of incumbent products such that the disruptive entrant, with continued development, eventually becomes competitive (suitably up-market) for traditional customers while delivering fewer capabilities in comparison to the incumbent products. The entrant becomes a rival for a share in the established market, likely competing at a lower cost.

3. The inability of an incumbent to modify its existing value chain to efficiently produce a product or service capable of competing with the disruptive entrant[14].

OSS market entrants bear some similarity to disruptive products in that they are generally poor performing upon entry relative to the established commercial market players. As an example, the highly popular open source database MySQL (the most used relational database management system (RDBMS) in the world) [7, http://en.wikipedia.org/wiki/Mysql], which has been provided under a discretionary pricing model, free of charge for non-commercial users and by paid license for commercial, was originally released, under its name MySQL, in 1998. High-level MySQL releases have included:

- 2004 (R-trees and B-trees, subqueries, prepared statements)
- 2005 (cursors, stored procedures, triggers, views, XA transactions)
- 2008 (event scheduler, partitioning, plugin API, row-based replication, server log tables)
- 2010 InnoDB becomes default storage engine: referential integrity constraints. Semisynchronous replication, user-defined partitioning [http://en.wikipedia.org/wiki/Mysql]

These advanced features were generally available in established commercial relational database systems, such as Oracle, in 1998. MySQL itself is neither a sustaining nor disruptive innovative product per se; however, MySQL's progress towards being the most used RDBMS follows a trajectory similar to that of disruptively innovative products. There was an underserved market for a free, or cheaper, RDBMS product that did not require the full capabilities of the established commercial RDBMS products. Established RDBMS vendors were not financially incentivized to serve this "down market" customer group interested in cheap RDBMSs. This underserved market adopted MySQL in its primitive form. Adoption created demand for more advanced features that were added incrementally either by MySQL AG, third parties (such as Innobase Oy) and community members. By 2008, MySQL was enough of a threat to the established market that Sun Microsystems bought MySQL AG for $1B [7, http://en.wikipedia.org/wiki/Mysql]. A year later, Oracle acquired Sun. The traditional metrics of technical performance are applicable to MySQL and the established RDBMS competitors, such as Oracle. Moreover, MySQL provides a standard user presentation similar to existing relational database products. However, MySQL's discretionary pricing model (free for most users) and evolutionary imitation of the incumbent RDBMS products eventually secured the product extraordinary distribution in a highly expanded market.

The combination of user familiarity and favorable pricing set a foundation from which MySQL

could develop in an incremental/evolutionary manner and eventually challenge established incumbents. The MySQL case can be characterized by:

1. Superior fit for a fringe market (those in need of a cheap RDBMS having the most basic storage features)
2. Established RDBMS incumbents providing a product far superior to that needed by the fringe market
3. The inability of RDBMS incumbents to provide an equivalently featured RDBMS version to challenge MySQL. Offering a capability-limited RDBMS, as a defense strategy, would likely not have been deemed financially worthwhile for an incumbent such as Oracle. Existing Oracle customers would not accept such a capability-limited version.

Eventually, as MySQL acquired greater capabilities, the OSS database became a competing product for organizations in need of an RDBMS product that would have otherwise selected products from major vendors such as Oracle, IBM or Microsoft.

9.2.19 Discussion

It has been proposed that OSS products supplement commercial offerings in under-serviced portions of the software marketplace [10]. However, there is ample evidence that OSS products challenge some of the most dominant and recognizable commercial software products [9 10 37 23]. Operating systems (Linux), relational databases (MySQL and postgreSQL) and productivity suites (Open Office) are at the core of mammoth open source efforts attracting thousands of potential commercial users away from incumbent commercial products [10 31]. The motivations for OSS contributors have been studied broadly by economists with theories proposed that are applicable to individual and industrial contributors [9 30 31 10]. There are clear examples of OSS-driven high-level corporate strategies intended to weaken competitors, such as IBM's substantial investment in Linux [36], as well as examples of commercial entities acquiring open source product rights (Oracle's acquisition of Sun included rights to the MySQL OSS database system for which Oracle now provides fee-based high value additions) [7]. Moreover, the use of OSS by government entities, a large software customer segment, is rapidly rising with pro-OSS policies and legislation gaining traction [10 23]. The literature, by and large, does suggest that OSS entrants pose serious challenges to incumbent commercial software vendors [10 31 23]. Purported individual factors for contribution include socio-psychological factors such as status, autonomy, mastery and community participation as well as economic potential from supplemental services beyond paid licensing such as a business consulting and the provision of support and maintenance [30 31 9].

Technical facilitators for OSS participation have also been studied. Modular design coupled with heterogeneous module option values have been theorized to promote efficiency of OSS product development capable of competing with commercial software production models [9 6]. Alternatively, voluntary production efforts, such as OSS, have been proposed to arise from a combination of individual self-selection (to provide effort) coupled with the acceptance of

certain elements of overt control. As such, voluntary production represents a novel hybrid model of transactions that possess both elements of markets and firms. Given this hybrid state, the likelihood that OSS projects could deliver products in the absence of some level of central coordination seems implausible [25]. Although the typical OSS project appears to have some element of formal central coordination, it is unclear as to what level of authoritative control would disrupt the drive for developers to self-identify and participate [25]. The author proposes that the use of established software design patterns, as a professional best practice, could serve to provide a locus of technical control without the encumbrance of personnel oversight and, therefore, could serve to lower transaction costs associated with OSS production.

However, empirical evidence that distinguishes OSS production from closed-source commercial software production is not definitive [32 34]. Quality metrics are not necessarily consistent with claims of superiority associated with OSS [31 23 10]. The difficulties associated with executing statistically relevant comparisons of software quality were discussed and are substantial [35]. Moreover, the techniques that would be envisioned to promote OSS, such as componentized design and iterative/agile development methodologies [8], are applicable and widely used across the breadth of largescale software development ventures. Isolating distinguishing quality features between OSS vs. closed source initiatives would, in this author's opinion, be exceedingly difficult, even if the underlying information to perform the comparisons was readily available.

The author notes that all inquiries into OSS processes focus on the technical components of the software development lifecycle such as the manner in which the architecture is designed and the code is integrated. Given the software industry's recognition that requirements elicitation appears to be the most serious factor implicated with the failure of software projects [19], it seemed prudent to examine whether there are business analytical distinctions between OSS and closed source initiatives. The advantages of evolutionary development, for large-scale projects, were discussed in the context of architectural patterns theory. That OSS projects are evolutionary in practice is of interest but difficult to reconcile as a distinguishing OSS feature. However, that many of the most popular OSS projects are able to imitate existing client use practices established a-priori by closed-source initiatives is both obvious and compelling. It would seem that OSS projects are substantially advantaged by the existence of mature corresponding closed-source products having user experience features, such as workflows and graphical user interfaces, which have been difficult to protect by existing intellectual property law [http://www.patentlyo.com/patent/2013/03/guest-post-what-is-next-in-design-patents-for-on-screen-icons.html…*Apple vs. Samsung*]. It will be interesting to see whether the Apple vs. Samsung litigation, which recently protected an Apple user interface from infringement, will expand to provide commercial originators of a user interface from copy by OSS projects.

The ability for OSS products to imitate existing closed source solutions, while likely an important enabler that speeds the development and launch of a usable familiar product, is unlikely, by itself, to account for OSS success in the marketplace. OSS products may act

somewhat like disruptive innovations [14]. If OSS products are able to leverage an unsupported market which has been left un-serviced by closed-source incumbents, the OSS product can expand this whitespace into a compelling niche market. The niche market, if sizable, interesting and important enough, would provide motivation for participation by capable developers. Participation would be especially compelling if the developers themselves could reap the benefits of the OSS directly, as appears to be the case for many OSS applications that have computing professionals as a targeted user group. With initial developer participation, existing products to imitate and a fringe customer base, a well architected OSS system based on modular or extensible software patterns could be incrementally developed to serve an expanding market. Success could result in greater project visibility and financial opportunities for participants resulting in sustained product enhancement. Eventually, the OSS would become suitably feature rich to compete with the existing up-market vendors for, at least a portion, of their customer base. Incumbents could try to compete directly with the OSS product by distributing a functionally diluted product for free. However, as with disruptive innovation scenarios, it is not likely that the incumbent will anticipate return on investment in supporting a down-market free offering and may struggle to compete once the OSS challenger is established in the fringe market. Simply, successful OSS appears to follow a disruptive pattern of market emergence although the OSS products themselves are not necessarily innovative from the perspective of customers.

The author does not discount any prior hypothesis regarding success factors for OSS. However, the availability of existing business use patterns, instituted by closed source software, is suggested to allow OSS project teams to circumvent the process of business analysis, the premier cited source of software project failure [19]. By avoiding pitfalls associated with product scoping, the OSS team can be in a position to quickly release feature-limited products for a core of interested niche users for which existing closed-source vendors cannot service due to cost prohibitions. The OSS project establishes a follower strategy for market growth that, in cases such as MySQL, Linux, Apache, Eclipse and many others, can eventually challenge the corresponding products of premier vendors.

Chapter 9 References

[1] Alexander, Christopher. 1964. Notes of the Synthesis of Form. Harvard University Press.

[2] Alexander, Christopher. 1966. A city is not a tree. Design London, Council of Industrial Design.

[3] Alexander, Christopher et. al. 1977. A Pattern Language. Oxford University Press.

[4] Atwood, Jeff. 2006. http://www.codinghorror.com/blog/2006/05/the-long-dismal-history-of-software-project-failure.html.

[5] Baldwin, Carliss and Clark, Kim. 2002. The Option Value of Modularity in Design. Harvard Business School. Draft for Comments.

[6] Baldwin, Carliss Y., Clark, Kim B. 2005. The Architecture of Participation: Does Code Architecture Mitigate Free Riding in the Open Source Development Model. Harvard Business School.

[7] Barret, Victoria. 2009. Why Oracle Won't Kill MySQL. Forbes.

[8]Beck, Kent et. al. 2001. Manifesto for Agile Software Development.

[9] Benkler, Yochai. 2002. Coase's Penguin, or, Linux and The Nature of The Firm. The Yale Law Journal.

[10] Bessen, James. 2005. Open Source Software: Free Provision of Complex Public Goods. Boston University School of Law.

[11] Brooks, Frederick. 1995. The Mythical man Month. Addison-Wesley.

[12]Califano, Andrea et. al. 2011. An Assessment of the Impact of the NCI Cancer Biomedical Informatics GRID (caBIG). National Institutes of Health, National Cancer Institute.

[13] Charette, Robert N. Sep 2005. Why Software Fails. We waste billions of dollars each year on totally preventable mistakes. IEEE Spectrum.

[14]Christensen, Clayton. 1997. The Innovator’s Dilemma. Harper’s Business.

[15] Coase, Ronald. 1974. The Lighthouse in Economics. Journal of Law and Economics.

[16]Fernandez-Ramil J., Izquierdo-Cortazar D. and Mens T. 2008. How Much Does It Take to Achieve One MegaLOC in Open Source? BENEVOL.

[17] Gamma, Helm, Johnson and Vlissides. 1994. Design Patterns: Elements of Reusable

Object-Oriented Software. Addison-Wesley.

[18] Gladwell, Malcolm. 2000. The Tipping Point: How Little Things Can Make a Big Difference. Little Brown.

[19] Golorath, Dan. Seer (Website). http://www.galorath.com/wp/software-project-failure-costs-billions-better-estimation-planning-can-help.php

[20] Grand, Mark. 1998. Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML. Wiley.

[21] Henttonen K., M. Matinlassi, E. Niemelä and T. Kanstrén. 2007. Integrability and Extensibility Evaluation from Software Architectural. Models – A Case Study. The Open Software Engineering Journal.

[22] Johansson, Frans. 2004. The Medici Effect: Breakthrough Insights at the Intersection of Ideas. Harvard Business School Publishing.

[23] Kenwood, Carolyn A. 2001. A Business Case Study of Open Source Software. MITRE Product.

[24] Krishnamurthi and Felleisen. 1998. Toward a Formal Theory of Extensible Software. ACM Foundations of Software Engineering Symposium Nov 1998.

[25] Langlois and Garazarelli. 2005. Of Hackers and Hairdressers: Modularity and the Organizational Economics of Open-Source Collaboration.

[26] Lee, Jyh-ahn. 2004. New Perspectives on Public Goods Production: Policy Implications of Open Source Software. Vanderbilt Journal of Entertainment and Technology Law.

[27] Lerner, Josh and Tirole Jean. 2000. The Simple Economics of Open Source. National Bureau of Economic Research.

[28] Narduzzo A and Rossi A. 2003. Modularity in Action: GNU/Linux and Free/Open Source Software Development Model Unleashed. Quaderno DISA.

[29] Parnas, D.L. 1972. On the Criteria To Be Used in Decomposing Systems into Modules. Comms of the ACM.

[30]Pink, Dan. 2009. Drive:The Surprising Truth About What Motivates Us. Penguin Group.

[31] Raymond, Eric Steven. 2000. The Cathedral and the Bazaar. Print version published by O'Reilly.

[32] Reasoning LLC. 2006. How Open Source and Commercial SoftwAare Compare: MySQL 4.0.16. Reasoning Technical White Paper. http://www.reasoning.com/pdf/MySQL_White_Paper.pdf

1. Apache:http://www.reasoning.com/pdf/Apache_Metric_Report.pdf
2. Tomcat:http://www.reasoning.com/pdf/Tomcat_Defect_Report.pdf
3. Linux TCP/IP: http://beta.finance-on.net/upload/sanjose/sanjose3e4c76ae3f77c/Open_Source_White_Paper.pdf
4. MySQL:http://www.reasoning.com/pdf/MySQL_White_Paper.pdf

[33] Samuelson, Paul. 1954. The Pure Theory of Public Expenditure. The Review of Economics and Statistics.

[34]Stamelos I., L. Angelis, A. Oikonomou & Georgios L. Bleris. 2002. Code quality analysis in open source software development. Info Systems J.

[35] SourceForge Top Project Listing: http://sourceforge.net/top

[36] Wong S., Y. Cai, M. Kim and M. Dalton. 2011. Detecting Software Modularity Violations. 33rd International Conference on Software Engineering

[37]Yoffie, David and Kwak, Mary. 2006. With Friends Like These: The Art of Managing Complementors. Harvard Business Review.

[38] Zakaria, Fareed. 2011. The Post American World, Release 2.0. W.W Norton and Co. ISBN 0-393-08180-x

Economic References for public goods.

5. http://en.wikipedia.org/wiki/Public_good
6. http://www.ewp.rpi.edu/hartford/~stoddj/BE/PubGoods.htm,
7. are.berkeley.edu/courses/EEP101/spring05/Chapter07.pdf

Conclusion

Jay Bergeron and Yike Guo

The Innovative Medicines Initiative (IMI), UK Medical Research Council and other European -based institutions have funded, and continue to fund, many groundbreaking translational research public private partnerships. The IMI envisioned a common information management platform that could be used to collect, process and analyse the data generated by these projects to reduce the cost of bespoke systems development while greatly facilitating the distribution and reuse of these data. The IMI-eTRIKS project, an effort spanning 2012-2018, developed and released, under open license, this generalized translational research knowledge management platform. Although the platform's features will need to expand and change over time as new research techniques are developed, eTRIKS created a system that has served the data handling needs of a great many European projects to date and continues to serve new projects through commercial and academic organizations that further develop and apply the platform. This book summarized some of the key practices and considerations associated with the challenging, albeit productive and satisfying, journey of creating the eTRIKS platform. The content is provided to help inform the perspectives of clinicians and scientists that participate in the generation and interpretation of the highly variable and large volume datasets that are associated with translational research studies.

In addition to saving the high costs of per-project custom information system development and eliminating intellectual property issues via open licensing, the eTRIKS platform also conserves the legacy of the transformational datasets being generated by IMI projects and other scientific public private partnerships. The eTRIKS platform reduces wasteful, redundant translational research investments and promotes a cohesive IMI Translational Medicine informatics community. Thus, the eTRIKS common knowledge management platform fosters the full realization of IMI project value by diminishing the obstacles to drug and diagnostic development caused by heterogeneous data processing and handling.

It is difficult, and likely not desirable, to develop a uniform service to meet the various needs of different stakeholders involved in translational research. Rather, building a flexible system infrastructure to support a wide range of services capable of addressing the diverse expectations of a broad scientific community is the sensible approach. The following groups of stakeholders were key to eTRIKS' development philosophy.

Academic researchers: eTRIKS is a system for managing and sharing traditional medical measurements with corresponding molecular biomarker data between scientists who conduct clinical research. This collaborative platform enables cross-institutional research. Thus, the system supports academic activities such as research process management, interoperability

with related systems and innovative analytics approaches that enable biomarker discovery, drug response, patient stratification and disease mechanism understanding. The system facilitates the development of new analytic methods to mine translational data, disseminating and publishing the results of these investigations. The system is also a vehicle for training the scientific community in new analytic methods and data handling standards and techniques. Extensible "plug in" workflows allow the platform to evolve to accommodate emerging clinical research methods and technologies, such as wearable devices. The importance of usability with respect to clinical researchers was well understood and an essential consideration during system design. The Borderline interface was developed to optimize the user experience for clinical researchers.

Pharmaceutical industry: The eTRIKS platform is designed to foster collaborative research programs involving pharmaceutical companies and academic researchers. An open and capable knowledge management platform that can be used with minimal cost by each consortium partner speeds the delivery of data to each partner and ensures that all partners have equal access to the data during and following the term limits of the partnership. Moreover, such a platform can be integrated with the enterprise informatics environments of the commercial partners to better facilitate confidential research endeavors involving both collaborative and company-proprietary datasets. Data fidelity and security are essential to industry partners and the eTRIKS platform benefits from rigorous software engineering and quality control processes.

Bioinformatics developers: Bioinformaticians are driven to develop new analytical and processing methods for biomedical data. Consistent open data standards and community-agreed data formats are critical facilitators to the work of the bioinformatician. Therefore, eTRIKS promoted stable data standards, common programmatic interfaces and interoperable open technologies. eTRIKS introduced an open license high performance distributed compute engine to accelerate infrastructure-exhaustive bioinformatic research methods.

Data engineers for curation and content management: Scientific data engineers build stable, curated and annotated translational content repositories. However, there are few tools and systems that support curation and long-term data management. Scientific data curation has essentially been an art practiced manually using only general scripting applications. The eTRIKS Harmonization System, a major effort of the eTRIKS project, provides a specialized environment to enable efficient scientific curation and content management incorporating well established standards, ontologies and meta data management services.

eTRIKS aspired to become the European Translational Research Commons Framework to support and enable translational medicine initiatives. eTRIKS provided a generalized information environment for scientific knowledge to flourish and for new approaches for the prevention, diagnosis, and treatment of disease to evolve, ultimately redefining the way biomedical research is translated to better health. eTRIKS developed an infrastructure which

enables scientific communities to build, expand and share informatics solutions. It is hoped that the reader has gained an appreciation of the nature of translational research data generation, analysis and interpretation as well as the critical importance of information management infrastructure in enabling the conduct of translational research studies.

Although the eTRIKS project ended in late 2018, closing formally in September 2019, the project supported an exceptional number of scientific teams during its tenure. Given the many academic and commercial entities that have adopted and continue to apply eTRIKS' best practices, the products and services developed by the many colleagues who participated in the eTRIKS consortium will continue to provide value to future collaborative research efforts within and outside of the IMI. It is hoped that these information best practices will play a crucial role in the mission to alleviate the suffering of patients who are fighting complex and chronic diseases.

About the Authors

Jay Bergeron represented *Pfizer* in the eTRIKS consortium serving as the consortium's Scientific Coordinator and co-lead for the Software Development and Management/Sustainability work packages. Jay chaired the code committee of the tranSMART Foundation while eTRIKS was underway. Jay is currently responsible for the data lakes and downstream analytic systems supporting clinical development and translational medicine at *Pfizer*.

Yike Guo, PhD co-led the eTRIKS consortium representing the managing entity, *Imperial College London*. Yike co-led the Software Development work package and served as Chief Technology Officer of the tranSMART Foundation while eTRIKS was underway. Yike is the founder and director of the *Data Sciences Institute, Imperial College London* and serves as Chief Innovation Officer at *IDBS*.

Scott Wagers, MD is the founder and CEO of *Biosci Consulting*. Scott served as the eTRIKS Program Management Officer co-leading the Management/Sustainability work package where he developed the eTRIKS Data Sciences Network. Scott is the author of the book *"Assembled Chaos: Accelerate your medical research career while changing the future of medicine through highly interactive consortia"* (published April 2020) based on his decade and a half experience in focused collaboration management. *Biosci Consulting* continues to enable biomedical consortia under Scott's leadership.

Mansoor Saqi, PhD spearheaded the disease network efforts within eTRIKS' Analytics work package as a member of the *European Institute for Systems Biology in Medicine* based at the *Centre National de la Recherche Scientifique* (Lyons). Mansoor is currently a Research Fellow at the *Data Sciences Institute, Imperial College London*.

Xian Yang, PhD is a computational scientist and Research Fellow at the *Data Sciences Institute, Imperial College London*.

David Henderson, PhD represented *Bayer* in the eTRIKS collaboration and co-led the Ethics and Data Reuse work package. David is a Principal Scientist at *Bayer* currently involved in the IMI-FAIRplus and T2EVOLVE consortia.

Fabien Richard, PhD participated in the Ethics and Data Reuse, Data Standards and Analytics work packages as a member of the *European Institute for Systems Biology in Medicine* based at the *Centre National de la Recherche Scientifique* (Lyons). Fabien is currently a Knowledge Management Scientist at *Merck Serono*.

Neil Fitch served as project manager for multiple eTRIKS work packages including Data Standards and Ethics and Data Reuse representing *BioSci Consulting*. Neil is currently a Project Management and Business Development Consultant based in Belgium.

Ibrahim Emam, **PhD** created the eTRIKS Harmonization Service as a member of the Analytics work package representing *Imperial College London*. Ibrahim is currently a Research Associate in the *Data Sciences Institute, Imperial College London*.

Florian Guitton, PhD served as the technical lead for the eTRIKS Software Development work package representing *Imperial College London*. Florian was instrumental to the creation of many key capabilities of the eTRIKS platform including the fully open source tranSMART stack, advanced biomedical data visualizations designed for collaboration within immersive "data observatories" and the bioinformatic microservices delivered in the final release of the eTRIKS platform. Florian is currently the Data Centre Operations Manager at the *Data Sciences Institute, Imperial College London* and Head of User Experiences at *Secretarium Ltd*.

Axel Oehmichen, PhD created the eTRIKS Analytical Environment as a member of the Analytics work package representing *Imperial College London*. Axel is currently Chief Data Sciences Officer at *Secretarium Ltd*.