Politecnico di Milano

School of Industrial and Information Engineering

Master of Science in Computer Science and Engineering

Department of Electronics, Information and Bioengineering

Integration of Genome-Wide Association Studies into the GeCo Repository

Supervisor: prof. Stefano CERI
Co-supervisor: Anna BERNASCONI, PhD
Arif CANAKOGLU, PhD

Master thesis by: Federico COMOLLI
Student Id n. 920258

Academic Year 2020-2021

Abstract

English version:

The first human genome has been sequenced at the turn of the year 2000. From this first project the modern biology has made great strides, thank to the introduction of Next-generation sequencing in the mid-2000s. The growing availability of genomic data has bring to the birth of the “tertiary analysis”, the one concerning sense-making of huge amount of data and useful biological information extraction. Many projects around the world have been brought forward in the last decade, obtaining a big amount of genomic data. Starting from the mid of '10s in the context of the GeCo project, some researchers of Polimi have introduced many tools to achieve genomic data integration to help biologists to perform tertiary analysis using multiple sources.

The Genomic Data Model, the Genomic Conceptual Model, the META-BASE architecture and the GMQL query language are some of the facilities proposed by the GeCo project to obtain genomic data integration. The META-BASE architecture is the core tool for the consolidation and it allows to transform raw data and to map them using a common conceptual schema. Integrated data can be queried or surfed using appropriate tools like GMQL or GenoSurf. All this works are meant to improve the quality of health care and to facilitate biologists to make new progresses in treat of diseases.

This thesis presents the efforts spent to integrate two more sources into the META-BASE architecture: `GWAS Catalog`, curated by the institutes NHGRI and EBI and `FinnGen`, curated by the University of Helsinki. It's the first time that are hosted Genome-Wide Association Study sources so the integration has required some extensions in the data schema of the GCM and the implementation of the new corresponding modules of the architecture.

The potentiality of the integration between multiple “omic” sources (e.g. ENCODE, Roadmap Epigenomics and TCGA) and GWA studies is then exploited running some GMQL queries to give a hint for future works and biological discoveries. Multi-omics studies are very important to deeply understand biological associations between genes, proteins, RNA and other omic data with the ultimate goal to improve human life.

Italian version:

Il primo genoma umano è stato sequenziato a cavallo degli anni 2000. Da questo primo progetto la biologia moderna ha fatto grandi passi avanti, grazie all'introduzione della tecnologia Next Generation Sequencing (NGS) a metà degli anni 2000. La crescente disponibilità di dati genomici prodotti ha portato alla nascita della "analisi terziaria", che riguarda la reinterpretazione e l'estrazione di informazioni biologiche utili da enormi quantità di dati. Molti progetti in tutto il mondo sono stati portati avanti nell'ultimo decennio, ottenendo una grande quantità di dati genomici. A partire da metà degli anni '10 di questo secolo nel contesto del progetto GeCo, alcuni ricercatori del Politecnico di Milano hanno introdotto molti strumenti per raggiungere l'integrazione in modo da aiutare i biologi a portare avanti l'analisi terziaria usando molteplici sorgenti di dati.

Il Genomic Data Model, il Genomic Conceptual Model, l'architettura META-BASE e il linguaggio di interrogazione GMQL sono alcuni degli strumenti che sono stati introdotti all'interno del progetto GeCo per ottenere l'integrazione di molteplici sorgenti genomiche. L'architettura META-BASE è lo strumento cardine per la consolidazione dei dati e permette di trasformare i dati grezzi e mapparli usando uno schema concettuale condiviso (il GCM). I dati integrati possono essere interrogati o resi accessibili grazie al linguaggio GMQL oppure tramite il servizio GenoSurf. Tutti questi sforzi hanno come scopo ultimo quello di migliorare la qualità della assistenza sanitaria e di facilitare la strada ai biologi per fare nuovi progressi nella cura delle malattie.

Questa tesi presenta gli sforzi compiuti per integrare due ulteriori sorgenti nell'architettura META-BASE: `GWAS Catalog`, curata dagli instituti NHGRI e EBI e `FinnGen`, curata dall'Università di Helsinki. E' la prima volta che vengono ospitate sorgenti GWAS, di conseguenza l'integrazione ha richiesto alcuni interventi al Genomic Conceptual Model e l'implementazione dei nuovi moduli corrispondenti dell'architettura.

Le potenzialità dell'integrazione tra molteplici sorgenti "omiche" (ad esempio ENCODE, Roadmap Epigenomics e TCGA) e le sorgenti GWAS sono sfruttate eseguendo alcune query GMQL per dare un suggerimento su possibili lavori futuri e su nuove scoperte biologiche. Gli studi multi-omici sono molto importanti per comprendere in profondità le associazioni tra i geni, le proteine, l'RNA e altri dati omici con lo scopo ultimo di migliorare la vita umana.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Deoxyribonucleic acid (DNA) is the most important molecule of the life. It is composed by two polynucleotide chains arranged in a double helix structure. The building blocks of the DNA are the five nucleotides cytosine [C], guanine [G], adenine [A] and thymine [T]. All the genetic information needed for all human being to develop, growth and reproduce is stored into the sequence of the four nucleotides C, G, A and T. The whole information is contained into all the cells of a human being. Cells from every tissue have all the 3,2 billion of base pairs. All humans share more than 99% of the DNA sequence; what justifies the diversity of our traits or diseases are small variations in the sequence of nucleotides.

The first attempt to sequence the base pairs of the human DNA was performed by The Human Genome Project [5], an international collaborative biological project launched in 1990. The rise of modern technologies has allowed the discoveries of modern genetics. The project was concluded in 2003 by sequencing about the 92% of human DNA and the result is stored into a public database made available through the World Wide Web.

After this initial project, many other followed it with the aim to understand how the sequence of nucleotides influence our diseases or phenotypes. In the mid-2000s, thanks to the introduction of Next-generation sequencing [25], a whole human DNA sequence can be known in short time and in a cheaper way.

The first ambitious project enabled by these new technologies was the 100,000 Genomes Project [7], followed by many others. It is a project started in the late 2012 conducted by the UK National Health Service. By the end of 2018, it achieved the goal to sequence the DNA of 100,000 patients of the NHS.

After being sequenced, is the turn of the so called tertiary analysis [24] which deals with sense-making of the huge amount of data produced by the previous analysis. The tertiary analysis is encouraged by data integration. Big amount of data produced by different studies need to be integrated to allow scientists to extract information useful to understand how life is orchestrated by the DNA and how the sequence of nucleotides affects diseases or phenotypes.

Data produced in the context of different projects have different formats, an obstacle for data interoperability required by the tertiary analysis.

A big effort to cope with genomic data heterogeneity is done by the GeCo project of Politecnico di Milano developing a data model (Genomic Data Model [19]), a query language (GenoMetric Query Language [18]) and a pipeline to integrate genomic data from multiple sources (META-BASE architecture [1]).

The GeCo project started in the mid of 10's of this century and its purpose is to create an integrated genomic repository which collects data from the major consortia around the world (like 1000 Genomes, Cistrome, ENCODE, GENCODE, RefSeq, Roadmap Epigenomics, TADs and TCGA). The ultimate goal of the GeCo efforts is to lay the basis to let biologists extract useful information from the sequenced genomes and to improve human life. This is also the objective of the "data science" which encloses the methods, processes and algorithms to extract knowledge and insights from raw data.

The goal of this thesis is to integrate into the META-BASE architecture a new class of studies called Genome Wide Association Studies (GWAS). GWA studies focus on variations of single nucleotides in the sequence of the DNA, in a case-control setup. By comparing groups of people with a disease or trait (cases) and without it (controls), the outcome of these studies are the more frequent nucleotides in the controls group against the cases. The difference of GWAS from other studies is the focus of the analysis: single nucleotide polymorphisms (SNPs) for GWAS, whole portion of genome or DNA features for other "omic" studies. In details, this thesis is focused on two GWAS repositories: GWAS Catalog and FinnGen.

The purpose of this thesis goes further GWA studies, aiming at integrating them with other "traditional" genomic data. Integrating into a shared data schema (the Genomic Conceptual Model [18]) multiple "omic" repositories (like genomic, proteomic and transcriptomic) serves to improve the knowledge about the molecular function and disease etiology. Multi-omic studies combines different biological entities to find novel associations between them, paving the road for disease treatments and prevention.

The goal has been achieved by mapping the conceptual schemes of the two GWAS sources into the Genomic Conceptual Model. The GCM has been extended in order to fit their schemes, so this thesis paves the road to integrate other GWAS sources into the GeCo repository.

This modelling step is followed by the implementation of the Scala classes and methods needed to achieve the integration into the META-BASE repository. The newly introduced modules of the META-BASE architecture easy the integration of other GWAS sources.

In Chapter 7 are presented some applications which leverage the integration efforts. The GenoMetric Query Language, developed in the context of the GeCo project, is a publicly available query language for genomic data. The user can exploit the public datasets, as well as private ones. All the available datasets are mapped into the GCM so that a single query can be built over multiple genomic datasets, since they share many metadata. The GMQL is an interoperable query language which can be exploit to run queries over multi-omic datasets. As result of the integration of GWAS Catalog and FinnGen into the

META-BASE repository, the two datasets are made publicly available into the GMQL web interface.

Genomic data can be considered "big data", since they are continuously growing and they come from heterogeneous sources. Between them, the four properties that characterize "big data" are:

- Volume: genomic data are continuously growing in size. Genomes from new individuals are sequenced every days, since the process is now cheaper and quite quickly.
- Velocity: genomic data are produced always faster and they have to be accessed quickly by experts or in a programmatic manner.
- Variety: the META-BASE architecture encloses genomic data from many consortia, each one with its own data schema.
- Veracity: data in the GeCo repository need to be consistent with the data in the original sources and they have to meet the integrity constraints of the GCM.

The two analyzed GWAS sources reflect all this properties, in fact are available periodically updated releases of their original repositories. Moreover, the integration tasks carried on during this thesis contribute to make the META-BASE architecture *interoperable*, *scalable* and *modular*.

My work is divided into two macro sections, the first describing how the integration is achieved and then are presented a few GMQL queries that demonstrate the benefits of genomic data integration.

The thesis is structured into 10 chapters, preceded by the Abstract.

Chapter 1 hosts this introduction and the goals of this thesis.

Chapter 2 illustrates the existing tools to achieve the integration between GWAS sources and explains their limitations.

Chapter 3 describes the background information upon which the work of this thesis is based on. It illustrates the main tools created in the context of the GeCo project to face the integration of multiple genomic sources.

Chapter 5 illustrates the conceptual efforts spent for the integration. It describes how the GCM has been extended and how the source-specific attributes are mapped over the ones of the new GCM.

Chapter 6 refers to the implementation of the META-BASE modules to achieve the integration. It contains the detailed description of how the steps of the pipeline are implemented, including some flow diagrams and data examples.

Chapter 7 introduces some GMQL queries over multiple genomic sources, included GWAS Catalog and FinnGen. This is a hint for future exploitation of integrated sources.

Chapter 8 contains the conclusions of this thesis and some possible future prospects.

Chapter 9 is made of a list of genomic-specific terms used in this thesis, accompanied by a short explanation. Finally, Chapter 10 shows a list of publications upon which this thesis is based on and from which valuable information are extracted.

Chapter 2

State-of-the-art

In this chapter is given an overview of the major current solutions adopted to handle large and heterogeneous genomic data, including GWA studies. For each proposed solution are introduced the pros and the cons. For the shortcomings that they present, it is explained how they are faced for reaching the goal of this thesis. The presented solutions are not meant to be an exhaustive list of the major genomic browsers, but they include the repositories and browsers that contain the data from the two GWAS sources considered in this thesis.

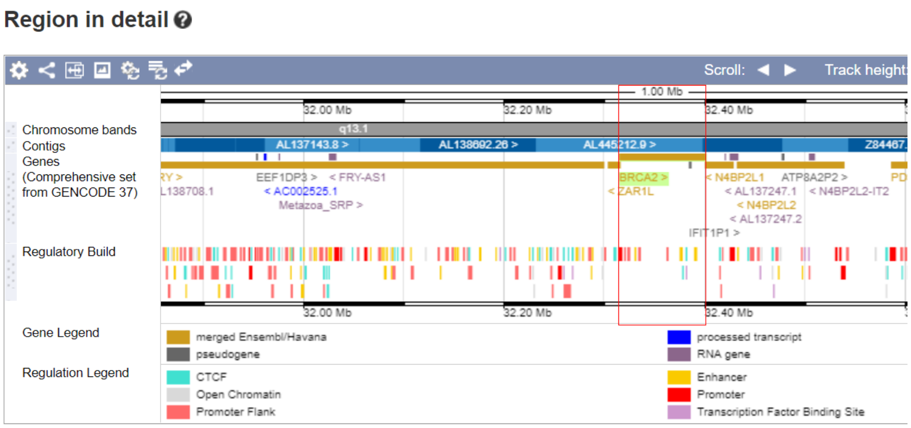
2.1 Ensembl

Ensembl is a long-term project, launched in 1999 by the European Molecular Biology Laboratory's European Bioinformatics Institute [13][11]. The goal of Ensembl was therefore to automatically annotate the genome, integrate this annotation with other available biological data and make all this publicly available via the web. In 2009, the Ensembl Genomes project was launched with specific web portals for plant, fungal, invertebrate metazoan, bacterial and protist genomes. By 2020, Ensembl supported over 50,0000 genomes.

Sequenced data are fed into the gene annotation system (a collection of software "pipelines" written in Perl) which creates a set of predicted gene locations and saves them in a MySQL database for subsequent analysis and display. The graphical visualization is the strength of the Ensembl project, displaying to the user the alignment of genes and other genomic data against a reference genome, allowing him to customize the display to suit his research interests.

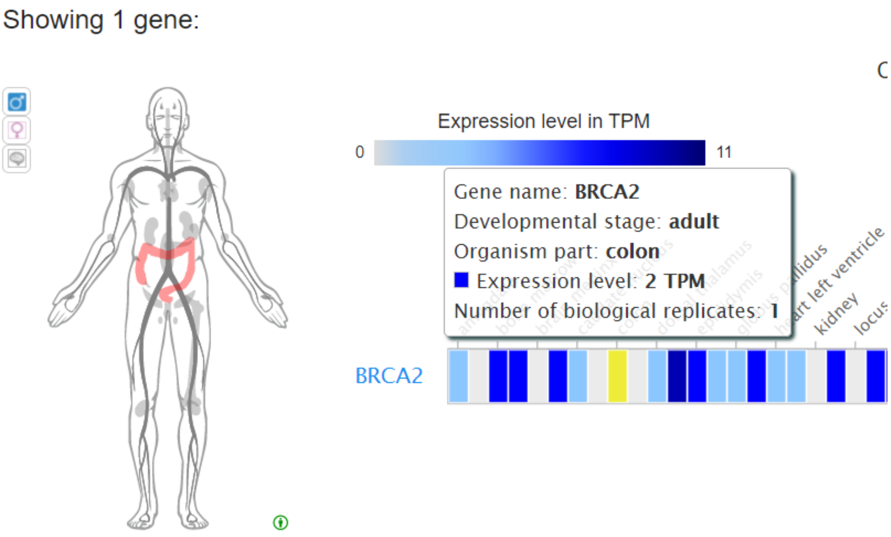
All the data are public accessible and downloadable by graphical tools, dedicated APIs, through a FTP server or by querying them into the BioMart data-mining tool.

Following are reported the screenshots of three graphical tools (Figures 2.1, 2.2 and 2.3) when studying the gene "ENSG00000139618", aka "BRCA2" using the HUGO ontology. The proposed tools are selected from more than forty diagrams available.



Chromosome bands Contigs Genes (Comprehensive set from GENCODE 37)		
Regulatory Build		
Gene Legend	merged Ensembl/Havana	processed transcript
	pseudogene	RNA gene
Regulation Legend	CTCF	Enhancer
	Open Chromatin	Promoter
	Promoter Flank	Transcription Factor Binding Site

Figure 2.1: In this figure is reported the position of the gene "BRCA2" over chromosome 13. The diagram is enriched with information about the adjacent genes and their derivations. For each gene in the diagram are highlighted, using different colors, the regulatory loci and their functional roles.



Gene name: **BRCA2**
Developmental stage: **adult**
Organism part: **colon**
Expression level: **2 TPM**
Number of biological replicates: **1**

Figure 2.2: This graphical tool shows an heatmap of the expression level of specific tissues for gene "BRCA2". In particular, is highlighted the colon of men (corresponding to 2 TPM). The expression level is indicated over a colored scale between 0 and 11 TPM (transcripts per million). The expression level indicates, for the tissue under study, the amount of molecules of RNA that are synthesized using the information encoded in the selected gene.

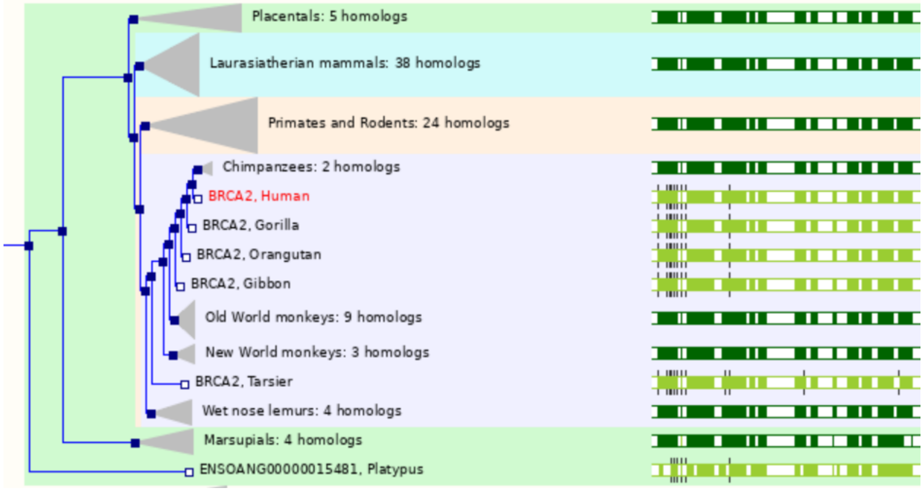

Placentals: 5 homologs
Laurasiatherian mammals: 38 homologs
Primates and Rodents: 24 homologs
Chimpanzees: 2 homologs
BRCA2, Human
BRCA2, Gorilla
BRCA2, Orangutan
BRCA2, Gibbon
Old World monkeys: 9 homologs
New World monkeys: 3 homologs
BRCA2, Tarsier
Wet nose lemurs: 4 homologs
Marsupials: 4 homologs
ENSOANG00000015481, Platypus

Figure 2.3: The Ensembl repository encloses also genomic data from many other species than human. In this tree diagram are shown homologues genes to “BRCA2” for multiple species. The species are grouped by their levels of similarity, considering the gene under study.

2.2 UCSC Genome Browser

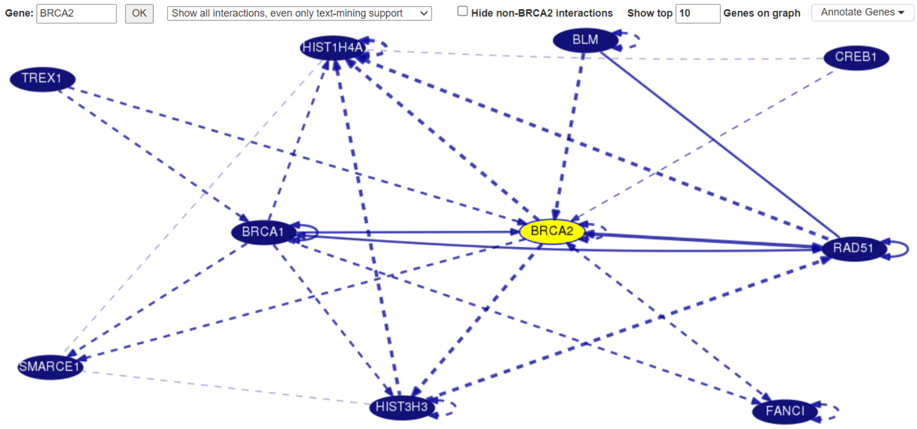
The UCSC Genome Browser was born at the University of California Santa Cruz in 2000, a few weeks after the first assembled genome was released on the web by the Human Genome Project [15]. The browser began as a resource for the distribution of the initial fruits of the HGP and it offered a graphical display of the first full-chromosome draft assembly of human genome sequence. From its born to nowadays, the browser has expanded to accommodate genome sequences of all vertebrate species and selected invertebrates for which high-coverage genomic sequences is available, now including 46 species.

The Browser is a graphical viewer optimized to support fast interactive performance and is an open-source, web-based tool suite built on top of a MySQL database for rapid visualization, examination and querying of the data at many levels.

Today the browser is used by geneticists, molecular biologists and physicians as well as students and teachers for access to genomic information.

The UCSC Genome Browser presents a diverse collection of annotation datasets (known as ”tracks” and presented graphically); the basic paradigm of display is to show the genome sequence in the horizontal dimension and show graphical representations of the locations of the mRNAs, gene predictions, gene-expression data and many other tracks in the vertical dimension.

Following are reported three screenshots taken from graphical tools of the UCSC Genome Browser (Figures 2.4, 2.5 and 2.6), when looking for gene “BRCA2” (the same gene of the examples of section 2.1 to make comparisons).



Gene: BRCA2 OK

Show all interactions, even only text-mining support

□ Hide non-BRCA2 interactions

Show top 10 Genes on graph

Annotate Genes ▾

Figure 2.4: The diagram shows the genes which interact with “BRCA2”. Only the 10 genes with the strongest associations are displayed, for graphical purpose. By clicking on an arrow, the information about the association and its product is shown to the user. The color and the line type indicate if the current association comes from text-mining tools or from database evidences.

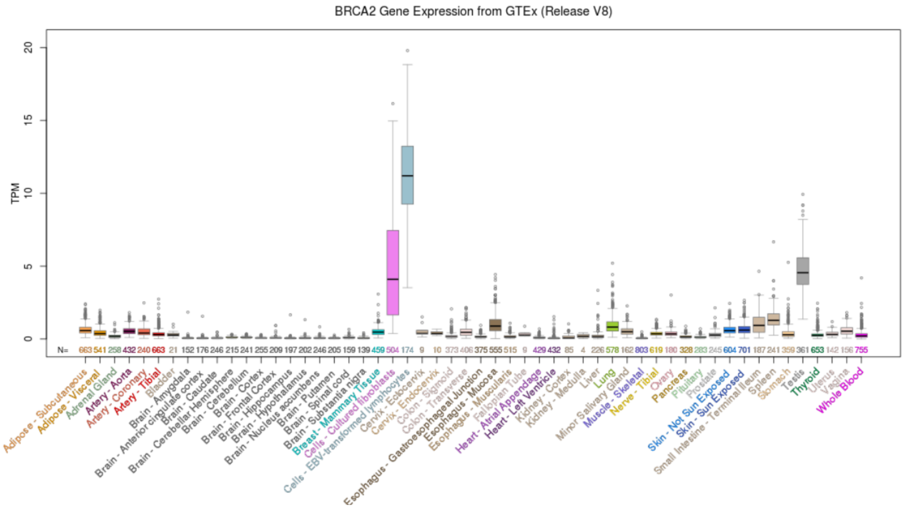


Figure 2.5: The box-plot compares the expression levels of gene “BRCA2” across 52 different tissues and 2 cell lines. This release is based on data from 17,382 tissue samples obtained from 948 adult post-mortem individuals. The highest median is for the cell-line `EBV-transformed lymphocytes`, corresponding to 11.20 TPM. The expression level of colon is in line with the TPM level expressed in figure 2.2.

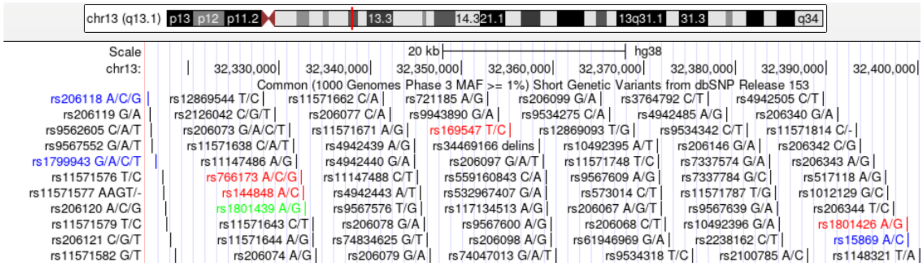


Figure 2.6: This schema represents some of the common (having MAF >= 1 %) SNPs occurring inside gene "BRCA2". The SNPs are shown according their position over the chromosome and they are colored depending on their functional effects. By clicking on a variant more information are displayed to the user such as its clinical consequences, some information about the studies and other details about the selected variant.

2.3 PheGenI

This project is more recent with respect to Ensembl and UCSC Genome Browser; it is born in the mid '10 of the current century by the National Center for Biotechnology Information. The PheGenI resource integrates content from several NIH resources such as dbGaP, GWAS catalog, dbSNP, NCBI Gene and eQTL data from the GTEx program [23].

The amount of available GWA studies is becoming bigger and bigger and each of them contains a list of associations between SNPs and phenotypic traits. The issue about GWA studies is that rarely are the true functional consequences of these variants understood. Thus replication, functional and follow-up studies are the crucial next steps. Integration of GWAS results with existing complementary databases can facilitate prioritization of variants for the follow-up, study design considerations and generation of biological hypotheses.

The user can surf the PheGenI browser following two main approaches: phenotype-oriented and genotype-oriented queries. The first approach consists of specifying the name of a trait while the second allows to specify the name of a SNP, of a gene or the coordinates of a location over chromosomes.

The browser page is divided into eight different sections. The first two concern the `Search Criteria` (phenotypic or genotypic search) and the `Search Summary` (statistics about the outcome of the search). Section `Gene` shows the resulting genes, section `Associations Results` lists the trait-SNP associations corresponding to the search criteria, section `SNPs` returns a table containing the SNPs matching the search criteria and section `Genome View` contains a graphical representation of the search outcome over the chromosomes. This last section is the only one having graphical support which consists of the schematic representation of all the chromosomes, with the possibility to interact with the schema. The last two sections `eQTL Data` and `dbGaP Studies` concerns the data coming from their respective sources.

Following are reported a couple of graphical outcomes from the PheGenI browser, one using the genotypic search (Figure 2.8) and the other using the phenotypic search (Figure 2.7).

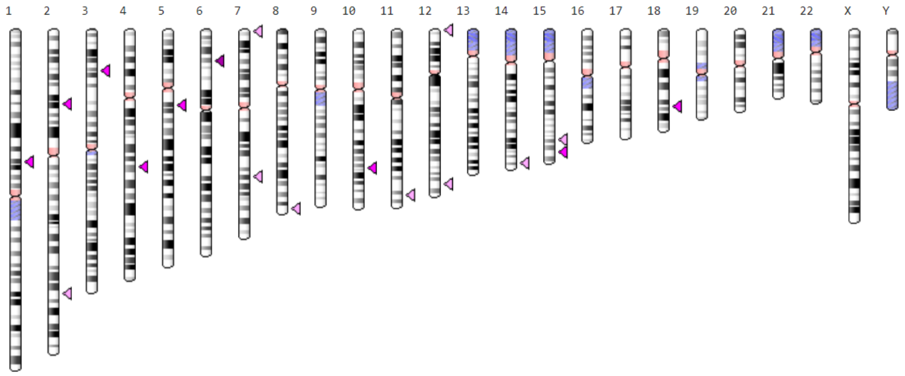


Figure 2.7: The picture is the `Genome View` of the PheGenI browser, showing results for "schizophrenia" (phenotypic search). In the picture are shown all the chromosomes, over which are mapped the genes that match the search criteria. Each gene, represented by a marker, can be clicked to open a more informative diagram such as the one in figure 2.8.

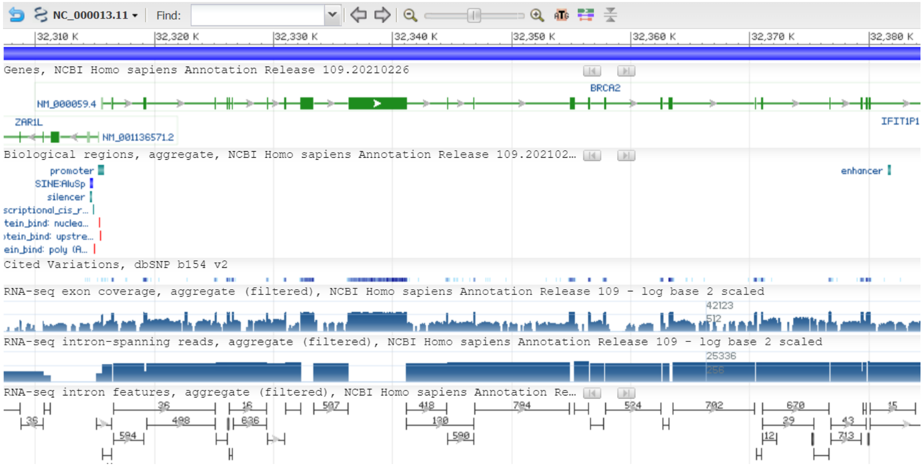

NC_000013.11
Find:
32,310 K
32,320 K
32,330 K
32,340 K
32,350 K
32,360 K
32,370 K
32,380 K
Genes, NCBI Homo sapiens Annotation Release 109.20210226
BRCA2
NM_000059.4
ZAR1L
NM_001136571.2
IFIT1P1
Biological regions, aggregate, NCBI Homo sapiens Annotation Release 109.202102...
promoter
SINE/AluSp
silencer
scriptional_cis_r...
tein_bind: nuclea...
tein_bind: upstre...
ein_bind: poly (A...
enhancer
Cited Variations, dbSNP b154 v2
RNA-seq exon coverage, aggregate (filtered), NCBI Homo sapiens Annotation Release 109 - log base 2 scaled
42123
512
RNA-seq intron-spanning reads, aggregate (filtered), NCBI Homo sapiens Annotation Release 109 - log base 2 scaled
25336
RNA-seq intron features, aggregate (filtered), NCBI Homo sapiens Annotation Re...
36
36
488
16
636
594
507
418
130
590
704
524
702
670
29
43
12
713
15

Figure 2.8: This multi-tracks diagram is obtained with the genotypic search for gene BRCA2 using the PheGenI browser. Each track is developed along the horizontal dimension and it is aligned over the reference chromosome. The tracks are stacked vertically and each of them contains information from its respective database. Examples of available tracks are intron and exon coverage as well as the variants in the current gene.

The genome browsers that are introduced and briefly descried in sections 2.1, 2.2 and 2.3 of this chapter are all valid tools for exploring the genome. They all have their strengths and weaknesses and they come from different contexts and periods. Their shared goal is to integrate GWA studies with other genomic data from multiple available sources in order to enrich the information that can be extracted using the genomic browsers.

GWAS data contains information about the associations between a phenotype and its causal variant. Unfortunately, mainly because of linkage disequilibrium, it's difficult to distinguish the truly causal variants from the ones linked to them. The prioritization of the variants is the next step to allow biologists to understand the truly functional role of each of them. The integration of GWA studies with other genomic sources is the key point to reach this goal.

GWA studies provide a list of associations and each variant is identified with its coordinates over chromosomes. Each "portion" of chromosome (enhancer, promoter, gene, ...) has its functional role in the life cycle of organisms, so it is necessary to enrich GWAS data with functional information about the positions over which the variations occur. This information improvement help biologists to understand the truly causal variants for the phenotype under study and to recognize the linkage between portions of chromosome.

We decided to undertake our own integration path since the META-BASE repository (presented in Chapter 3), previously developed in the context of the GeCo project of Politecnico di Milano, contains many genomic data from multiple sources (1000 Genomes, Cistrome, ENCODE, GENCODE, RefSeq, Roadmap Epigenomics, TADs and TCGA). Its strength is the shared conceptual schema (the Genomic Conceptual Model) of all the integrated data, allowing to create queries over multiple integrated genomic sources using the GenoSurf browser or the GMQL query tool.

The META-BASE architecture is an integration environment based on metadata, the most challenging attributes. Region attributes are made of coordinates of the regions under consideration and many additional information. Based on the four coordinates (chr, left, right, strand) it is possible to align all the DNA regions coming from different sources; the same is not possible also with their metadata. Mapping the source-specific metadata into the GCM, the META-BASE architecture overcomes this obstacle allowing the full integration of multiple genomic sources, both the regions and their metadata.

Including GWAS data into the GeCo repository allows to reproduce the GMQL queries presented in section 7, used as examples of the potentiality of the integration.

Chapter 3

GeCo Background

3.1 Genomic Data Model

The Genomic Data Model [19] is a data format which links genomic feature data to their associated experimental, biological and clinical metadata. The GDM is able to homogeneously describe semantically heterogeneous data and paves the way for providing data interoperability. The need for the GDM arouse from the tertiary-analysis of genomic data. After the primary analysis (production of sequences of base pairs called ”reads”) and the secondary analysis (alignment of reads to a reference genome and search for specific features), the tertiary analysis is the most interesting one. It concerns with knowledge extraction from heterogeneous genomic data: its goal is to understand how different regions interact and cooperate with each other.

In the genomic field, a dataset consists of collections of samples. Each sample contains two parts: the region data and the metadata. The former describes the features of the DNA while the latter are information about the sample. A sample s is formally modeled as a triple <id; R; M >where:

- id is the sample identifier
- R is the set of regions of the sample. It is composed by pairs <c; f >of *coordinates* c and *features* f. The *coordinates* are (chr, left, right, strand) and they identify a region on a reference genome; the *features* indicate properties of the identified region, such as p-value.
- M is the collection of metadata, composed by pairs <a; v >of *attributes* a and *values* v.

According to the GDM, a genomic dataset can be stored using two separate data structures, one for region data and one for metadata. The GDM allowed to create an integrated data repository from open sources like ENCODE, Roadmap Epigenomics and TCGA.

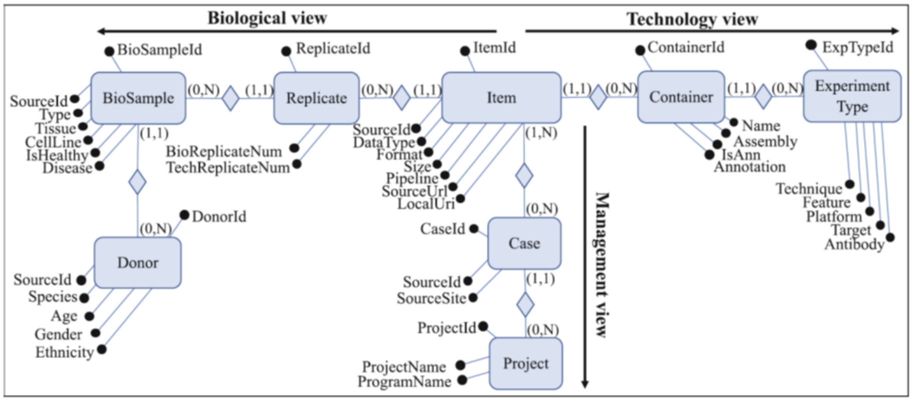


Figure 3.1: The Genomic Conceptual Model proposed by the GeCo project [2]. It illustrates the three views through which the central *experiment item* is analyzed.

3.2 Genomic Conceptual Model

The Genomic Conceptual Model [2] is born out of the need of sharing genomic data. It was built starting from the most common public data repositories and then was validated through the repositories TCGA, ENCODE and Gene Expression Omnibus. The GCM helps the tertiary analysis of genomic data since it offers a unified conceptual schema for many repositories. It is an ER model so it is quickly understandable. The central entity of the GCM is the *experiment item* and it represents an atomic information. It is analyzed by three different views:

- **Biological View:** describes the provenience of the data. It includes information about the tissue from which data are produced, the disease of the donor and its personal data.
- **Technology View:** offers details about how the experiment is carried on. It is related to technical aspects of the analysis.
- **Management View:** contains information related to the context in which data are produced. It shows the project or organization which has conducted the experiment.

In the Figure 3.1 are shown all the entities with their attributes and the three views in which the schema is organized.

The **Biological View** describes the biological process leading to the production of the `Item`, the central entity. It is composed by the entities `Donor`, `BioSample` and `Replicate`. Each `Item` can be made only of a single `Replicate` and a `Replicate` can contribute to many `Items`. The `Donor` represents the individual of a specific organism from which the biological material is derived. The biological sample taken from a `Donor` is represented by the entity `BioSample`. Its attributes describe the material sample taken from a biological entity. The entity `Replicate` is used when multiple material samples are generated from the same `BioSample`. For detailed description of the cardinalities, please refers to the Figure 3.1.

The **Management View** describes the organizational process carried on for the production of each `Item`. It includes the entities `Case` and `Project`. This last entity describes the project responsible for the production of the `Item`. The entity `Case` gathers different `Items` which participate to the same research objective.

The **Technology View** describes the technology through which an `Item` is produced. It is made of the entities `Container` and `ExperimentType`. The former is used to describe common properties of homogeneous items, the latter refers to the specific methods used for producing each `Item`.

3.3 GMQL

The GenoMetric Query Language has been proposed in the context of the GeCo project [18] in 2015. It supports queries over multiple heterogeneous genomic data sources. The GMQL is based on the Genomic Data Model presented above and it is similar to the well-known Structured Query Language (SQL). A GMQL query is expressed as a sequence of operations with the following structure:
< variable >= operator(<parameters >)< variables >;
where each variable stands for a GDM dataset. The operators can be on metadata (SELECT, EXTEND, ORDER), on regions (PROJECT, COVER, GROUP), as well as on multiple datasets (UNION, DIFFERENCE, JOIN, MAP).
A typical GMQL query starts with a SELECT operation, which loads a dataset with only the data samples that it filters out from an input dataset. Then, the query proceeds by processing the samples with the specified operations. Finally it ends with a MATERIALIZE operation, which stores the results as a GDM dataset. The GMQL has proved to meet all the three main challenges in data-intensive genomic analysis: declarativeness, portability and scalability.

The queries can be posted manually on the dedicated web interface or programmatically using the apposite REST Web services. The GenoMetric Query Language is made of twelve basic operators [22] and it is useful to answer to many interesting biological questions [21]. Following is reported the full list of the available operators, enriched with a brief explanation and a few examples.

SELECT

It selects all samples in input dataset and copies them in the output. The samples can be filtered through some values of metadata or though some region attribute values. As example, consider the operation:

RES = SELECT(patient_age <70; region: chr == chr1) input it takes all the samples from the dataset “input”, filters only the ones coming from patients younger than 70 years old and copies in “RES” dataset only regions on chromosome 1, which result from the previous filter.

MATERIALIZE

This statement is always necessary in order to compile and execute a query. Only in this way a result of the computation becomes visible and available for download. The typical statement is:

MATERIALIZE RES INTO materialize;

it saves the content of the temporary “RES” dataset into a file named [queryname]_[timestamp]_materialize.

PROJECT

The basic behaviour of this operation is to remove all the region attributes which are not coordinates (only *chr*, *start*, *stop* and *strand* are kept). If other metadata or region attributes are specified, it keeps as output only the region coordinates and the specified attributes. This operator allows also to create new region or metadata attributes, specifying how to compute it from existing attribute values. As example, let’s consider the operation:

RES = PROJECT(region_update: length AS right - left) D;

it creates a new dataset called “RES” by preserving all region attributes and creating a new region attribute called *new_right* which contains a copy of the value of the coordinate attribute *right*.

EXTEND

It allows to add some metadata to the input dataset, extracting their values from the regions. Let’s consider the operation:

RES = EXTEND(region_count AS COUNT(), min_pvalue AS MIN(pvalue))
D;

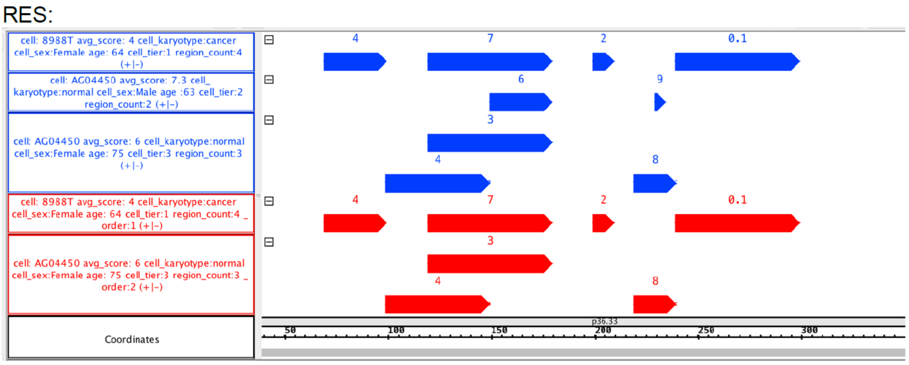
the new metadata attributes *region_count* and *min_pvalue* are introduced, computing them respectively by counting the number of regions in the input dataset and selecting the minimum *pvalue* of the input regions.

ORDER

It orders the samples of the input dataset according to the values of a specified metadata or region attribute. After the sorting it extracts only some of the samples, if specified in the query. For example, the operation:

RES = ORDER(Region_count DESC; meta_top: 2) D1;

orders the samples of the input dataset “D1” according to descending order of the *Region_count* metadata attribute, and it selects only the first two samples. The graphical outcome of this statement is shown in Figure 3.2.



RES:

cell: 8988T avg_score: 4 cell_karyotype:cancer cell_sex:Female age: 64 cell_tier:1 region_count:4 (+|-)

cell: AG04450 avg_score: 7.3 cell_karyotype:normal cell_sex:Male age :63 cell_tier:2 region_count:2 (+|-)

cell: AG04450 avg_score: 6 cell_karyotype:normal cell_sex:Female age: 75 cell_tier:3 region_count:3 (+|-)

cell: 8988T avg_score: 4 cell_karyotype:cancer cell_sex:Female age: 64 cell_tier:1 region_count:4 _ order:1 (+|-)

cell: AG04450 avg_score: 6 cell_karyotype:normal cell_sex:Female age: 75 cell_tier:3 region_count:3 _ order:2 (+|-)

Coordinates

Figure 3.2: This figure shows how the operation "ORDER" of the GMQL language works. Given the input dataset D1 (blue), the resulting dataset (red) is made of the two top samples with the highest number of regions, ordered according to descending number of regions.

GROUP

This operation groups together regions belonging to the same sample, which have the same coordinates (*chr*, *start*, *stop* and *strand*). Additional regional attributes can be specified in addition to the four coordinates, used by default. If some metadata are specified, they are used to group together different samples which share the values of the metadata specified. As basic example, let's consider the query:

RES = GROUP(cell_karyotype; region_aggregates: min_pvalue AS MIN(pvalue)) D;

the samples of the input dataset "D" are grouped according to the value of the metadata *cell_karyotype*. Then, inside each sample, the regions which share the same coordinates are grouped together and the attribute *min_pvalue* is introduced for each resulting region.

MERGE

This operation collapses all the samples of the input dataset into a single one. Specifying the parameter *groupby*, the samples are grouped together according to the value of the metadata specified. As simple example, consider the query:

RES = MERGE(groupby: sex) D;

this statement creates the dataset "RES", which contains one sample for each *sex* value found within the metadata of the input dataset "D". Inside each resulting sample there are the regions of the input dataset with the same specific value for the *sex* metadata.

UNION

This is a very basic operator. It creates a new dataset containing all the samples from the two specified input datasets. Let's consider the query:

RES = UNION() D1 D2;

it creates the dataset "RES" containing all the samples from dataset "D1" and "D2".

DIFFERENCE

Specifying two input datasets, it performs the intersection of the two sets. Consider the query:

RES = DIFFERENCE(exact: true) D1 D2;

a new dataset "RES" is created and it contains all the regions of the dataset "D1" that do not coincide (exactly from the *start* to the *end* coordinates) with at least a region in the dataset "D2". The graphical outcome of this statement is shown in Figure 3.3.

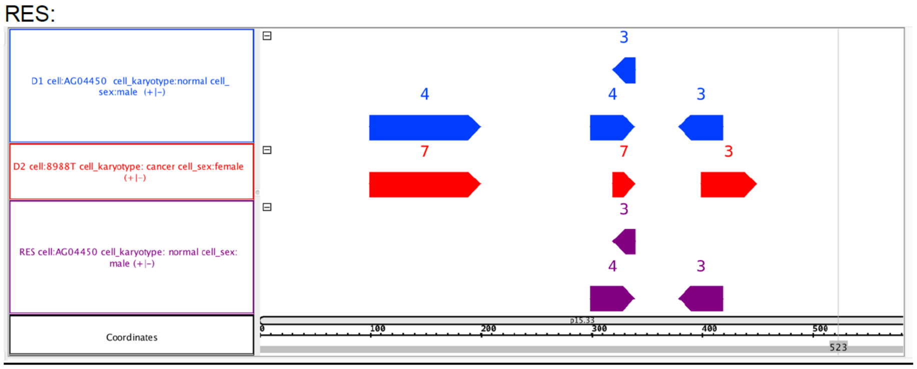

RES:
D1 cell:AG04450 cell_karyotype:normal cell_sex:male (+|-)
D2 cell:8988T cell_karyotype: cancer cell_sex:female (+|-)
RES cell:AG04450 cell_karyotype: normal cell_sex: male (+|-)
Coordinates
3
4
4
3
7
7
3
3
4
3
p15.33
0
100
200
300
400
500
523

Figure 3.3: This figure shows how the operation "DIFFERENCE" of the GMQL language works. Given the two input datasets D1 (blue) and D2 (red), the resulting dataset (purple) is made of the regions from D1 which do not coincide with any regions from D2.

MAP

This operation compares each sample of the two input datasets, counting the number of regions that overlap. Let's consider the following statement:

RES = MAP(avg_score AS AVG(score)) D1 D2;

the dataset "D1" is made of one sample, while "D2" is made of three samples. The output dataset "RES" contains three samples (multiplication of the cardinalities of the input datasets) counting the number of regions in each sample from the "D2" dataset which overlap with a region in the "D1" dataset sample.

This statement also computes the average score value across such regions, saving the results in the output "RES" dataset as a region attribute called *avg_score*.

JOIN

It is the most computationally intensive of all GMQL operations. It is an operation between two input datasets and, for each region of the first, it returns all the regions of the second dataset which fulfill certain conditions. It basis on the *distal condition*, which can be of the five types: DL (or DLE), DG (or DGE), MD, UPSTREAM and DOWNSTREAM. The matching of the two input datasets can be driven by a metadata value, if specified. Let's consider the following statement:

RES = JOIN(MD(1), DGE(150)) D1 D2;

for each region of the dataset "D1" extracts the closest (because of MD(1)) region from the dataset "D2", and the statement excludes from the output the regions of "D2" which are at a distance smaller than 150 base pairs from the corresponding region of "D1". Let's consider this other statement:

RES = JOIN(MD(1), DGE(150); output: CAT; joinby: cell_karyotype) D1 D2;

the behaviour is the same of the previous one, but are joined only the samples that share the same value of the metadata *cell_karyotype*. The option "CAT" indicates that the output is produced as the concatenation of regions resulting from the statement.

COVER

This statement is an unary operation and it returns the areas of an input dataset, which fulfill the input condition. Two parameters *minAcc* and *maxAcc* need to be specified. Let's consider the following GMQL statement:

RES = COVER(2, ANY) D1;

it creates the dataset "RES" containing the areas defined by a minimum of two overlapping regions up to any amount of overlapping regions in the input dataset samples. The graphical outcome of this statement is shown in Figure 3.4.

In the documentation of the GMQL language [21] are reported some examples of queries useful to answer to interesting biological questions. As example, a biological question is <<Consider all public somatic mutation data samples of TCGA Kidney Renal Clear Cell Carcinoma patients. For each sample, count the mutations occurring in each exon and select the exons with at least one mutation. Return such samples together with the number of such exons and the maximum number of mutations in a single exon.>>

The GMQL query used to answer is reported below:

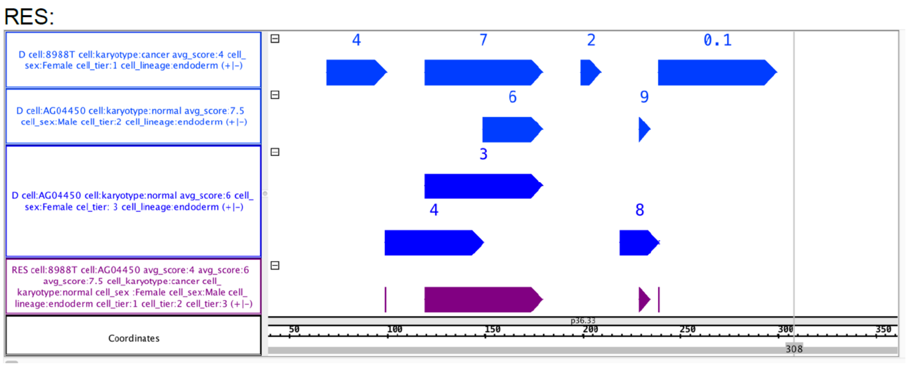

RES:
D cell:8988T cell:karyotype:cancer avg_score:4 cell_sex:Female cell_tier:1 cell_lineage:endoderm (+|-)
D cell:AG04450 cell:karyotype:normal avg_score:7.5 cell_sex:Male cell_tier:2 cell_lineage:endoderm (+|-)
D cell:AG04450 cell:karyotype:normal avg_score:6 cell_sex:Female cel_tier: 3 cell_lineage:endoderm (+|-)
RES cell:8988T cell:AG04450 avg_score:4 avg_score:6 avg_score:7.5 cell_karyotype:cancer cell_karyotype:normal cell_sex :Female cell_sex:Male cell_lineage:endoderm cell_tier:1 cell_tier:2 cell_tier:3 (+|-)
Coordinates
4
7
2
0.1
6
9
3
4
8
p36.33
50
100
150
200
250
300
350
308

Figure 3.4: This figure shows how the operation "COVER" of the GMQL language works. Given the input dataset D1 (blue), the resulting dataset (purple) is made of the areas from D1 which have at least two overlapping regions.

```
MUT = SELECT(manually_curated__cases__disease_type ==
      "Kidney␣Renal␣Clear␣Cell␣Carcinoma")
      GRCh38_TCGA_somatic_mutation_masked;
EXON = SELECT(annotation_type == "exon" AND release_version
       == "22") GRCh38_ANNOTATION_GENCODE;
EXON1 = MAP() EXON MUT;
EXON2 = SELECT(region: count_EXON_MUT >= 1) EXON1;
EXON_RES = EXTEND(exon_count AS COUNT(), max_mut AS
           MAX(count_EXON_MUT)) EXON2;
MATERIALIZE EXON_RES INTO EXON_RES;
```

From the dataset "TCGA", are selected the samples from patients with the disease "Kidney Renal Clear Cell Carcinoma". The exon regions are extracted from the dataset "GENCODE", release 22. The MAP operation maps the mutation to the exon regions. Finally, are extracted only the mapped exons which contain at least one mutation.

3.4 META-BASE repository

The integration of genomic metadata is a very important goal achieved in the context of the GeCo project [1]. The META-BASE architecture is a pipeline which aims to generate the GCM content. The task to integrate many genomic sources is challenging since domain is complex and there is no agreement among the vocabularies and ontologies used as metadata. The project is written using the Scala programming language and the integrated repository is managed using the Apache Spark engine.

The proposed pipeline is composed by six steps, illustrated in Figure 3.5.

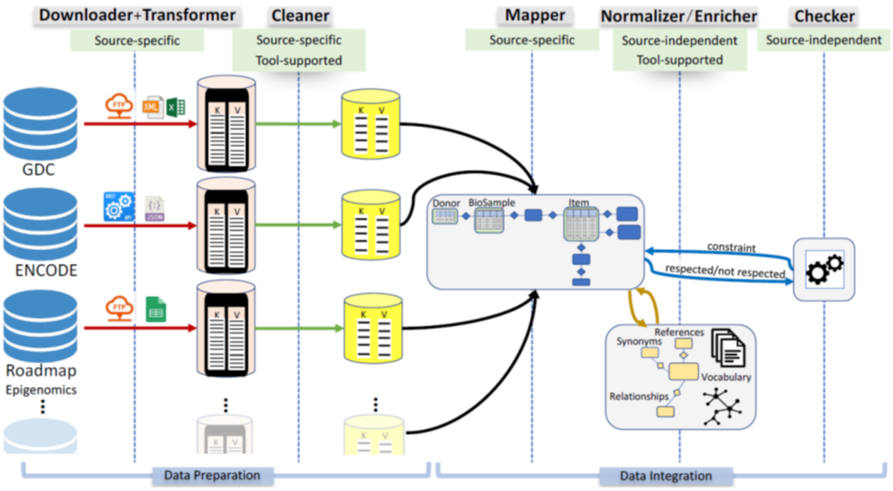


Figure 3.5: The pipeline of the META-BASE architecture [1]. The six sequential steps are needed to integrate a data repository into the META-BASE one.

They are splitted between the *Data Preparation* phase and the *Data Integration* one. Here are presented the details of each step:

1. **Data Download:** data are downloaded through available APIs or protocols from different genomic sources. The sources don't share a standard for region attributes and metadata, so the challenging part is to map each attribute into one of the two classes. Partitioning is important to selectively update a single partition, when it is modified on the source database. This process avoids to re-download the entire database even when a single file is changed.

2. **Data Transformation:** referring to the *experiment item* of the Genomic Conceptual Model, two files for each item are created. The metadata are divided from region data and following, only metadata files are considered. The metadata could be provided into three different structures: hierarchical format (JSON, xml), comma/tab-delimited formats or unstructured metadata. They are transformed into a file containing <key><value>pairs.

3. **Data Cleaning:** after being flattened, many <key> have very long names, not suitable for the next steps. Some rules source-specific are defined to simplify the attributes name. A rule is composed by the left and the right parts; the former recognizes complex patterns and the latter simplifies them into short strings. The rules are organized into an ordered list called *Rule Base*.

4. **Data Mapping:** also this step is based on source-specif rules. The goal is to populate the tables of the GCM with the pairs obtained by the previous

steps. A rule is a couple of attribute names; the first one is the source-specific name, the second is the name of the attribute in the Genomic Conceptual Model. At the end of this step, most of the metadata are integrated into the META-BASE database.

5. **Data Normalization and Enrichment:** it is a source-independent and supervised procedure. It aims to normalize some attributes values against a known ontology. This step is supported by the *Local Knowledge Base*, a structure that integrates different ontologies available. An examples of two synonyms are *breast carcinoma* and *breast neoplasm*; they are merged into the disease *breast cancer*.

6. **Integrity Checker:** some dependencies between values of the GCM are manually defined and they must be checked. An example of dependency is: if DONOR is “Homo sapiens” then the Assembly of the dataset must be one of “hg19”, “hg38” or “GRCh38”.

The implementation of the META-BASE architecture is executed using the Apache Spark engine. Its content can be queried through the GenoMetric Query Language [18] or through a user-friendly interface called GenoSurf [4].

3.5 GenoSurf

The integration of genomic data repositories has highlighted the importance of a user-friendly interface to query the integrated metadata. The GenoSurf interface is a web user interface public available through the website `http://geco.deib.polimi.it/genosurf/` and it allows to query the META-BASE repository, to analyze metadata and to retrieve the corresponding raw data from their original source [4]. A query is composed by selecting search values from the integrated attributes among predefined normalized term values enriched with the Local Knowledge Base ontologies. Another way to retrieve metadata is to specify pairs <key ><value >referred to the original metadata. The web interface allows the user to further analyze the retrieved data using the GMQL engine. All queries in GenoSurf are translated into a JSON format in order to support re-use of them. User can download a query structure and upload it when needed.

3.6 GWAS

Genome-Wide Associations Study is a way to find the associations between a genetic variant and a trait or disease. It is typically conducted in a case-control setup. Two groups of people are selected merely according to the phenotype, one affected by a trait or a disease named cases group and the other is an healthy controls group. A DNA sample is taken from all individuals (through blood collection or cheek swab) of the two groups and they are genotyped against a reference genome. If one nucleotide in a certain position of the chromosomes is more frequent in people of the cases group over the controls group, the variant is said to be associated with the trait or disease. As result, for the most common single-nucleotide polymorphisms are reported some statistics about the alleles that appear in the controls and cases groups.

GWAS is a non-candidate-driven approach, in contrast with the previous studies which focused on small number of pre-specified genetic regions. The whole DNA of the two groups is scanned to find out the associations between the trait under study and some SNPs.

These studies aim at improving the knowledge in the health field, so to allow researchers to develop better strategies to detect, treat and prevent the disease. The future of healthcare includes personalized medicine, by which a tailored strategies is developed ad hoc for each patients and its necessities.

GWAS were born in the early 2000s thanks to the availability of new technologies that give the chance to quickly and accurately analyze whole-genome samples for genetic variations that contribute to the onset of a disease. The DNA samples are quickly analyzed by a machine that strategically selects markers of genetic variation (SNPs). The first GWA studies were conducted in the early 2000 and nowadays there are thousands of studies available. While many studies were carried on, some public repositories were born to let them access in a simple way.

Between them, GWAS Catalog (`https://www.ebi.ac.uk/gwas/`) is one of the most important collection of different unstructured literature sources [3]. It was created by the National Human Genome Research Institute (NHGRI) in 2008 and it has become a collaborative project between the NHGRI and the European Bioinformatics Institute (EBI) since 2010.

Their data are public available into three different files: associations, studies and ancestries. Those files are updated monthly and their terms are mapped to the Experimental Factor Ontology(EFO). In the release of May 2021, the repository counts more than 16 thousands of studies. GWAS data can also be queried from the GWAS Catalog website specifying the value of one attribute such as the trait or the SNP and so on. The website offers also many diagrams, useful to graphically visualize the SNPs over the chromosomes and their p-values.

Chapter 4

Data sources

This thesis focuses on two GWAS data sources: GWAS Catalog [17] and FinnGen [9]. In this chapter are described the features of these public repositories, from the history of the projects to the technical aspects of the data. Then in Chapter 5 are illustrated the challenges faced to integrate the data model of the new sources with the Genomic Conceptual Model [2]. In Chapter 6 are explained the technical details of the implementation of the integration into the META-BASE architecture.

4.1 GWAS Catalog

The development of the Catalog has been a collaborative project between EMBL-EBI (European Bioinformatics Institute) and NHGRI (National Human Genome Research Institute) since 2015 [6]. It is an open-access database of GWA studies. It gathers unstructured studies and, thanks to manual curation, they are stored into a publicly-accessible structured data repository. New studies are found through weekly PubMed searches. New data are manually extracted from the literature by expert scientists and some information about SNPs is added by an automatic pipeline. The traits are mapped to the Experimental Factor Ontology (EFO), so to encourage data sharing and interoperability.

The repository can be accessed through the search engine on the website or can be downloaded by means of three files through the website of the Catalog, through the dedicated FTP server or though an API. The user can exploit the search engine specifying the name of a trait (e.g. “breast carcinoma” in Figure 4.1)), the identifier for a SNP (e.g. “rs7329174” in Figure 4.3)), an author or other values for attributes in the Catalog.

Moreover data can be accessed through an user-friendly diagram showing the SNPs on chromosomes, filtering them by trait names (see Figure 4.2 for an example).

The files names are self-explicating: Ancestry.tsv, Studies.tsv and Associations.tsv. New versions of the repository are released monthly. Release of May

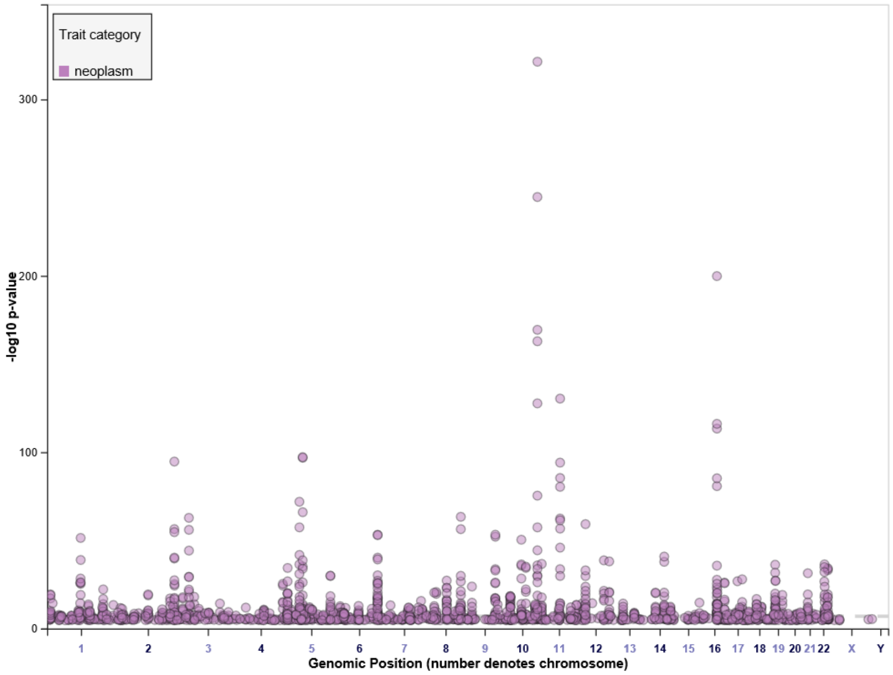


Figure 4.1: The figure is shown by the GWAS Catalog search engine by specifying the trait name "breast carcinoma". The engine shows to the user many results, such as the studies related to the specified traits or the associations between SNPs and the trait under consideration. The image is created by a tool named "LocusZoom" and shows all the SNPs associated with the searched trait, specifying their position and information about the study in which they have been discovered.

6th 2021 includes 16,854 studies, corresponding to 257,352 associations between SNPs and related traits.

The Studies file contains one entry for each trait of a PubMed study, so multi-traits PubMed studies are splitted according the traits. The Ancestry file contains information about the cohort of people used for the studies (e.g. the number of people that participated in a study and their provenience). The biggest file is the Associations one, containing a row for each association (relation between a SNP and the study trait). This latter file contains also statistical properties about the correlations found (e.g. "p-value"). The three files share some attributes, so they can be merged into a structured data frame as shown in the next chapters. Following is reported the complete list of the attributes of the Catalog with a brief explanation of their meaning, each one linked to the files in which it appears ([AS] stands for Associations file, [S] for studies and

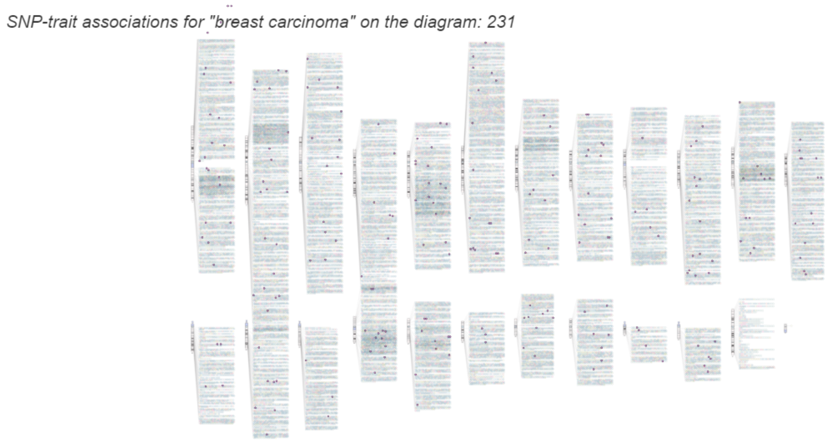

SNP-trait associations for "breast carcinoma" on the diagram: 231

Figure 4.2: In the figure are reported the SNPs available in the GWAS Catalog mapped to trait “breast carcinoma”. The diagram shows the SNPs according to their position on chromosomes. Each SNP can be selected in an interactive way and information about them are shown to the user.

[AN] for Ancestry).

- DATE ADDED TO CATALOG [AS][S]: the date in which a study is published in the Catalog.
- PUBMEDID [AS][S][AN]: PubMed identification number.
- FIRST AUTHOR [AS][S][AN]: last name and initials of first author.
- DATE [AS][S][AN]: publication date of the study online.
- JOURNAL [AS][S]: abbreviated journal name in which the study is published.
- LINK [AS][S]: PubMed URL of the study.
- STUDY [AS][S]: title of paper.
- DISEASE/TRAIT [AS][S]: disease or trait examined in study.
- INITIAL SAMPLE DESCRIPTION [AS][S][AN]: sample size and ancestry description for initial stage of the study.
- REPLICATION SAMPLE DESCRIPTION [AS][S][AN]: sample size and ancestry description for subsequent replication(s) of the study.
- REGION [AS]: cytogenetic region associated with the SNP.
- CHR_ID [AS]: chromosome number associated with the SNP.

- CHR_POS [AS]: position in the chromosome of the SNP.
- REPORTED GENE(S) [AS]: gene(s) reported by author.
- MAPPED GENE(S) [AS]: gene(s) mapped to the strongest SNP. If the SNP is located within a gene, that gene is reported. If the SNP is located within multiple genes, these genes are listed separated by commas. If the SNP is intergenic, the upstream and downstream genes are listed, separated by a hyphen.
- UPSTREAM_GENE_ID [AS]: entrez Gene ID for nearest upstream gene to rs number, if not within gene.
- DOWNSTREAM_GENE_ID [AS]: entrez Gene ID for nearest downstream gene to rs number, if not within gene.
- SNP_GENE_IDS [AS]: entrez Gene ID, if rs number within gene; multiple genes denote overlapping transcripts.
- UPSTREAM_GENE_DISTANCE [AS]: distance in kb for nearest upstream gene to rs number, if not within gene.
- DOWNSTREAM_GENE_DISTANCE [AS]: distance in kb for nearest downstream gene to rs number, if not within gene.
- STRONGEST_SNP_RISK_ALLELE [AS]: SNP(s) most strongly associated with trait + risk allele (? for unknown risk allele). May also refer to a haplotype.
- SNPS [AS]: strongest SNP; if a haplotype it may include more than one rs number.
- MERGED [AS]: denotes whether the SNP has been merged into a subsequent rs record (0 = no, 1 = yes).
- SNP_ID_CURRENT [AS]: current rs number (will differ from strongest SNP when merged = 1).
- CONTEXT [AS]: SNP functional class.
- INTERGENIC [AS]: denotes whether SNP is in intergenic region (0 = no, 1 = yes).
- RISK ALLELE FREQUENCY [AS]: reported risk allele frequency associated with strongest SNP in controls.
- P-VALUE [AS]: reported p-value for strongest SNP risk allele.
- PVALUE_MLOG [AS]: -log(p-value).
- P-VALUE (TEXT) [AS]: information describing context of p-value.

- OR or BETA [AS]: reported odds ratio or beta-coefficient associated with strongest SNP risk allele.
- 95% CI (TEXT) [AS]: reported 95% confidence interval associated with strongest SNP risk allele, along with unit in the case of beta-coefficients.
- PLATFORM (SNPS PASSING QC) [AS][S]: genotyping platform manufacturer used in initial stage.
- CNV [AS]: study of copy number variation (yes/no).
- ASSOCIATION COUNT [S]: number of associations identified for this study.
- MAPPED_TRAIT [AS][S]: trait mapped over the Experimental Factor Ontology.
- MAPPED_TRAIT_URI [AS][S]: URI of the EFO trait.
- STUDY ACCESSION [AS][S][AN]: accession ID allocated to a GWAS Catalog study.
- GENOTYPING_TECHNOLOGY [AS][S]: genotyping technology used in this study, with additional array information in brackets.
- STAGE [A]: stage of the GWAS to which the sample description is referred (initial, replication).
- NUMBER OF INDIVIDUALS [AN]: number of individuals in this sample.
- BROAD ANCESTRAL CATEGORY [AN]: broad ancestral category to which the individuals in the sample belong.
- COUNTRY OF ORIGIN [AN]: country of origin of the individuals in the sample.
- COUNTRY OF RECRUITMENT [AN]: country of recruitment of the individuals in the sample.
- ADDITIONAL ANCESTRY DESCRIPTION [AN]: any additional ancestry descriptors relevant to the sample description.

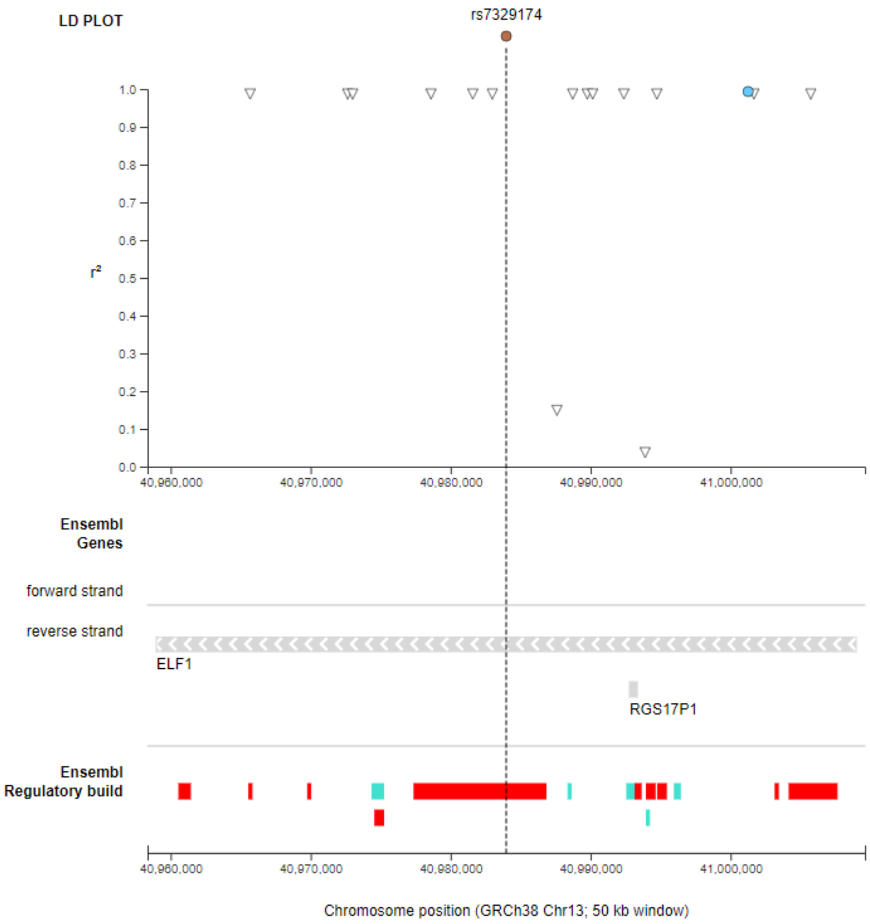


Figure 4.3: The figure is one of the results showed by the GWAS Catalog search engine by specifying the variant identifier "rs7329174". This variant has been found associated to the Crohn's disease. The picture shows how this variant and the variant "rs57141708", associated with the height, are in Linkage Disequilibrium. This last is a condition for which the presence of a variant influence the presence of the other, so to make difficult the comprehension of what are the truly causal variants for a phenotype.

The three files can be merged by means of the reported shared attributes, among which the most relevant is the STUDY ACCESSION. As a basic example, in Table 4.1 are reported the entries for the study accession "GCST005097" taken from the Studies file. For greater emphasis on the eyes, in the tables are reported only few attributes of the three files Ancestry, Studies and Associations. In Table 4.2 are reported the entries from the Ancestry file, referred to the same study accession. To conclude, in Table 4.3 are reported the three associations found in that study.

Table 4.1: In this table are reported some relevant attributes from the Studies file for the study accession "GCST005097".

PUBMEDID	FIRST AUTHOR	JOURNAL	STUDY	DISEASE-TRAIT	ASS. COUNT
29170203	Alonso N	Ann Rheum Dis	Identification of a novel locus on chromosome 2q13, which ...	Fractures (vertebral)	3

Table 4.2: In this example are reported some relevant attributes from the Ancestry file for the study accession "GCST005097".

STUDY ACCESSION	INITIAL SAMPLE DESCRIPTION	REPLICATION SAMPLE DESCRIPTION	STAGE	NUMBER OF INDIVIDUALS	COUNTRY OF RECRUITMENT
GCST005097	1,553 cases; 4,340 controls	1,028 cases; 3,762 controls	replication	2799	U.K., Italy, Spain
GCST005097	1,553 cases; 4,340 controls	1,028 cases; 3,762 controls	replication	1991	U.K.
GCST005097	1,553 cases; 4,340 controls	1,028 cases; 3,762 controls	initial	5893	Australia, Denmark, U.K., Slovenia, Spain

Table 4.3: In this last example are reported some of the relevant attributes of the three associations found in study with study_accession “GCST005097”. The entries reported come from the Associations file.

REGION	CHR_ID	CHR_POS	MAPPED GENE(S)	SNPS	P-VALUE
15q26.1	15	92464744	ST8SIA2	rs2290492	3*10 -7
2q13	2	112192944	AC092645.1 - ZC3H8	rs10190845	1*10 -9
11q12.1	11	57980425	OR5BD1P - CYCSP26	rs7121756	4*10 -7

4.2 FinnGen

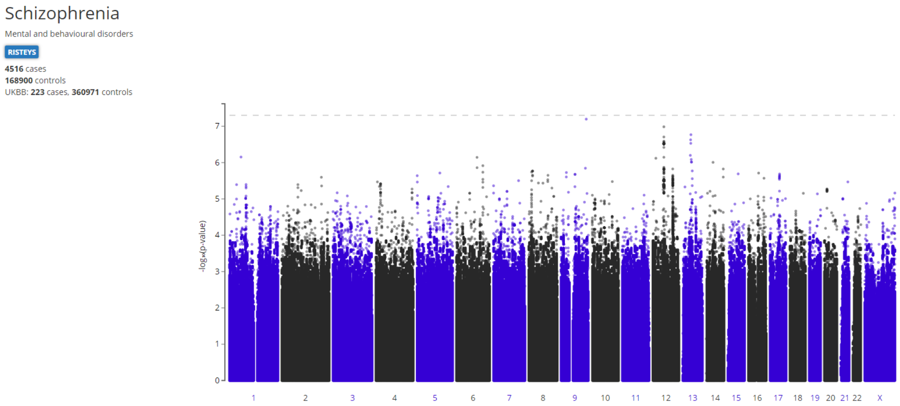
The project is a big collaboration between private and public Finnish institutes, born in Autumn 2017 [9]. It aims to improve human health through genetic research. Collaboration between universities, hospitals, biobanks and pharmaceutical companies is the key to achieve disease prevention, diagnosis and treatment. The project wants to pave the road to personalized medicine with ad-hoc treatments, besides to produce medical innovation with an ever seen private-public collaboration.

Every Finnish person can take part in the project by giving the consent to be part of the study cohort. The project aims to reach a cohort of 500,000 Finnish people by 2023 and they are already close to the goal (441,000 people in March 2021). All the individuals that take part to this giant study are genotyped using GWAS. Of course there is a special consideration for data protection and privacy for people who participate in the cohort.

The FinnGen project is composed by many genome-wide association studies. The outcome of these studies are the SNPs which are found relevant for the phenotypes under consideration, called endpoints in FinnGen context. Data can be accessed through a search engine by specifying an endpoint (the name of a trait or phenotype) as in Figure 4.4 or can be downloaded through different channels, both programmatic access or web browser-based access.

The repository is updated twice a year and it is publicly available the year after it is produced. The most recent available release now (May 2021) is the fifth (Release 5). It contains the SNPs associated to 2,804 endpoints.

The repository is composed by two mainly modules: summary statistics and fine-mapping. The summary statistics are composed by a Manifest (containing the name of the endpoints) and many files as the number of endpoints in the release. Each endpoint file of the summary statistics contains all the SNPs associated to that phenotype and some statistical properties of the SNPs like p-value. Also the fine-mapping module contains one file for each endpoint, each



Schizophrenia

Mental and behavioural disorders

4516 cases
168900 controls
UKBB: **223** cases, **360971** controls

Figure 4.4: In the figure is reported a Manhattan plot displaying the SNPs associated to the phenotype "schizophrenia", obtained through the search engine "PheWeb". The plot shows the SNPs according to their position on chromosomes. Each SNP can be selected in an interactive way to zoom on the nearby of a locus.

one including the outcomes of the fine-mapping process.

FinnGen data are fine-mapped with the softwares "SuSiE" and "FINEMAP". The corresponding files contain information about the importance of the associations found in the studies, taking into consideration the linkage disequilibrium.

Following is reported the complete list of the attributes of the FinnGen summary statistics with a brief explanation of their meaning, each one linked to the files in which it appears ([M] if the attribute is in the Manifest file, [E] if it is contained into the endpoint files).

- phenocode [M]: alphanumeric code given to a phenotype.
- name [M]: complete name of the phenotype.
- n_cases [M]: cardinality of the cases group for the associations of the current trait.
- n_controls [M]: cardinality of the controls group for the associations of the current trait.
- path_bucket [M]: path of the current file for Google cloud-based access.
- path_https [M]: path of the current file for command-line access.
- #chrom [E]: chromosome on build GRCh38.
- pos [E]: position in base pairs on build GRCh38.
- ref [E]: reference allele.

- alt [E]: alternative allele (effect allele).
- rsids [E]: variant identifier.
- nearest_genes [E]: nearest gene name from variant.
- pval [E]: p-value from SAIGE.
- beta [E]: effect size estimated with SAIGE for the alternative allele.
- sebeta [E]: standard deviation of effect size estimated with SAIGE.
- maf [E]: alternative (effect) allele frequency.
- maf_cases [E]: alternative (effect) allele frequency among cases.
- maf_controls [E]: alternative (effect) allele frequency among controls.

For clarity purpose in Table 4.4 are reported a few entries contained in the Manifest of the fifth release. In Table 4.5 and Table 4.6 are reported some entries from files "TUBERCULOSIS.gz" and "F5_SCHIZO.gz" respectively.

Table 4.4: In this table are reported a few entries of the Manifest of the Release 5 of FinnGen repository.

phenocode	name	n_cases	n_controls	path_bucket	path_https
F5_SCHIZO	Schizophrenia, schizotypal and delusional disorders	7999	168900	gs:// ... F5_SCHIZO.gz	https:// ... F5_SCHIZO.gz
G6_PARKINSON	Parkinson's disease	1587	175312	gs:// ... G6_PARKINSON.gz	https:// ... G6_PARKINSON.gz
TUBERCULOSIS	Tuberculosis	801	176098	gs:// ... TUBERCULOSIS.gz	https:// ... TUBERCULOSIS.gz

Table 4.5: In this table are reported a few entries of the file "TUBERCULOSIS.gz". In the table appear only some relevant columns of the file.

#chrom	pos	ref	alt	rsid	pval
1	115637	G	A	rs74337086	0.8316
1	216439193	T	C	rs571377638	0.7225

Table 4.6: In this table are reported a few entries of the file "F5_SCHIZO.gz". In the table appear only some relevant columns of the file.

#chrom	pos	ref	alt	rsid	pval
1	133855	C	T	rs528106901	0.5142
1	195798153	A	C	rs2942912	0.655

Chapter 5

Methods

In this chapter are introduced the challenges to fit the two data sources GWAS Catalog and FinnGen as instances of the GDM [19]. The GCM [2] has been extended to accomplish also the attributes meaningful for GWA studies. The analysis starts with the GWAS Catalog repository followed by the FinnGen one.

5.1 Data Design

5.1.1 GWAS Catalog region data and metadata

In this first part of Chapter 5 are described the modelling steps that we have performed as support of the implementation phase of the integration, described in Chapter 6.

The Genomic Data Model [19] described in Chapter 3 is the basis of the META-BASE architecture [1]. Since the goal of this thesis is to integrate the new data sources into the already implemented META-BASE architecture, the very first step is to map the GWAS Catalog as an instance of the GDM.

The entries of the GDM are triples <id; R; M >, where:

- **id** is the sample identifier
- **R** is the Region part of a sample
- **M** is the Metadata part of a sample

It is necessary to split also the GWAS Catalog into this binary partition "Region" and "Metadata". The Region part contains attributes strictly referred to genomic features of the sample. It includes the coordinates of the genomic region under consideration and the properties of that region. All the region attributes are taken from the file *Associations.tsv*.

The Metadata instead are all the attributes needed to describe the study in which the Region is analyzed. The Genomic Conceptual Model [2] described in Chapter 3 is made only of metadata. Metadata include information about the people that provided biological samples, as well as information about who conduced the study or the technologies used to analyze the samples. Region data refer to DNA features, metadata describe how region data are produced.

All the attributes of the Catalog are described in Chapter 4; following is provided a binary partition of them taking into account the structure of the GCM for metadata.

Presenting the metadata attributes, they are grouped according to the entity of the GCM they belong.

Entity CaseStudy

- PUBMEDID
- STUDY
- LINK

Entity Item

- STUDY ACCESSION
- PLATFORM (SNPS PASSING QC)

Entity ExperimentType

- GENOTYPING_TECHNOLOGY

The entities of the GCM `Donor`, `Biosample` and `Replicate` do not fit the Metadata of GWA studies. A couple of entities need to be added to describe the cohorts and their provenience. Two main differences between GWA studies and the "traditional" ones are the basis of this changes in the conceptual model. Traditional studies are based on single person, not cohorts of people. The second relevant difference is the target of the study: single mutations in the whole chromosome for GWAS, regions of chromosomes for traditional annotation studies.

Entity Cohort gathers all the attributes related to the composition of the cohort from which the study is conducted. Between the Catalog attributes, the ones belonging to this new entity are:

- MAPPED_TRAIT
- INITIAL SAMPLE DESCRIPTION
- REPLICATION SAMPLE DESCRIPTION

Entity Ancestry contains the attributes that describe the origin or provenience of the people that participate in the cohort of the study:

- NUMBER OF INDIVIDUALS

- BROAD ANCESTRAL CATEGORY
- COUNTRY OF ORIGIN
- COUNTRY OF RECRUITMENT
- ADDITIONAL ANCESTRY DESCRIPTION

The attributes DATE ADDED TO CATALOG, FIRST AUTHOR, DATE, JOURNAL, DISEASE/TRAIT, ASSOCIATION COUNT, MAPPED_TRAIT_URI and STAGE are discarded since the GCM does not contain them and they don't provide a conceptual contribution to the model to justify the creation of new entities.

In the previous lists are reported a rough division of the attributes of the GWAS Catalog according to the entities of the GCM, both already present or freshly introduced. Many of them will not appear as they are in the GCM, but they need some manipulation or manually curation. This process is described in the subsection 5.1.3 and in Chapter 6.

The attributes in the below list are the ones that describe the coordinates of the SNPs in the chromosomes or features of the identified mutations, therefore they belong to region data.

- REGION
- CHR_ID
- CHR_POS
- REPORTED GENE(S)
- MAPPED GENE(S)
- UPSTREAM_GENE_ID
- DOWNSTREAM_GENE_ID
- SNP_GENE_IDS
- UPSTREAM_GENE_DISTANCE
- DOWNSTREAM_GENE_DISTANCE
- STRONGEST_SNP_RISK_ALLELE
- SNPS
- MERGED
- SNP_ID_CURRENT
- CONTEXT

- INTERGENIC
- RISK ALLELE FREQUENCY
- P-VALUE
- PVALUE_MLOG
- P-VALUE (TEXT)
- OR or BETA
- 95% CI (TEXT)
- CNV

5.1.2 FinnGen region data and metadata

In this section are retraced the steps to map the FinnGen repository as an instance of the GDM. As explained in Chapter 4, the repository is composed by a Manifest and one file for each endpoint. Compared to GWAS Catalog, FinnGen has less metadata and they are mainly contained in the Manifest; the endpoints files contain the region data.

The Genomic Conceptual model is filled only with metadata. As basis for the implementation phase described in Chapter 6, here is reported a binary partition of the attributes of FinnGen into region data and metadata. The new GCM proposed in section 5.1.3 contains only metadata, both rough or manually curated. The following FinnGen attributes are grouped according the entities of the GCM and according the freshly introduced entities “Ancestry” and “Cohort” in section 5.1.1.

Entity CaseStudy

- phenocode
- path_https

Entity Cohort

- name
- n_cases
- n_controls

The remaining attributes, taken from the endpoints files, belong to region data since they describe the coordinates of the mutations over chromosomes and contain some features of the SNPs found in the study.

- #chrom
- pos

- ref
- alt
- rsids
- nearest_genes
- pval
- beta
- sebeta
- maf
- maf_cases
- maf_controls

5.1.3 New Genomic Conceptual Model

The Genomic Conceptual Model in Figure 3.1 presented in Chapter 3 has been introduced by A. Bernasconi et al. [2] with the aim to standardize genomic metadata from different sources. This data modelling work has started with the integration of multiple sources like TCGA, ENCODE, Gene Expression Omnibus, 100k Genomes Project, Roadmap Epigenomics Project and other projects.

During the years the presented GCM has been slightly improved; for example the entities `Container` and `ExperimentType` of the **Technology View** has been splitted into the **Extraction View** and the **Technology View**.

The **Biological View** is not able to satisfy the needs of the GWA studies. They are based on cohorts of people, of which we don't know the personal anagraphical information. About the cohorts we know some aggregate information about their geographical provenience. Accordingly, the entity `Donor` looses its meaning. The entity `BioSample` also is meaningless in the context of GWA studies since are not provided information about the tissue or cell line from which the biological samples come from.

Taking into account these considerations, a new view is introduced: the **GWAS View**. It is made of the entities `Ancestry` and `Cohort`. They capture the information about the composition of the cohorts and their ancestral information. Each GWA study is based on multiple stages; each one can be "initial" or "replication". A single study can have more than one initial stages and zero or more replication stages. This information is aggregated in the `Cohort` entity. The initial proposal of the New GCM included two different entities for the initial and replication stages; but at the end we have tried to keep the model as simple as possible, aggregating all the information about the cohort into a single entity.

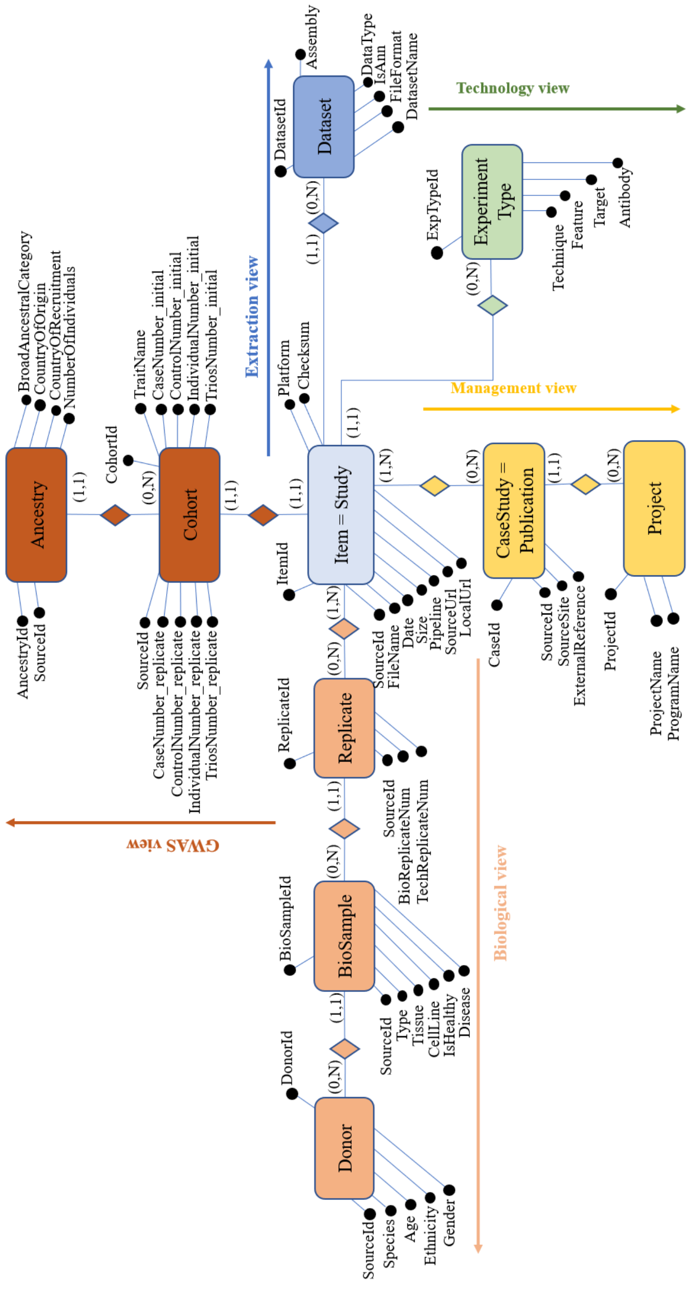


Figure 5.1: This is the new version of the Genomic Conceptual Model, modified to fit the features of GWA studies. The starting point is shown in Figure 3.1 in Chapter 3. The new view "GWAS view" is introduced, containing the new entities "Ancestry" and "Cohort".

The entity Cohort is made of the following attributes:

- CohortId: the identifier of the current cohort
- TraitName: it indicates the phenotype or disease under consideration for the current Item
- SourceId: stands for the study accession of the Item to which the Cohort refers
- CaseNumber_initial: the number of people in the case group(s) of the initial stage(s)
- CaseNumber_replicate: the number of people in the case group(s) of the replicate stage(s)
- ControlNumber_initial: the number of people in the control group(s) of the initial stage(s)
- ControlNumber_replicate: the number of people in the control group(s) of the replicate stage(s)

Some GWA studies are not based on the traditional cases and controls setup, but the Cohort can be made of individuals or trios. The Cohort is composed by individuals when the people that participate are not splitted into cases and controls but they are all grouped together, so they are all cases (they manifest the phenotype) or all controls (they don't manifest the phenotype). Moreover, the Cohort is made of trios when participate people with their parents.

- IndividualNumber_initial: the number of people in the individual group(s) of the initial stage(s)
- IndividualNumber_replicate: the number of people in the individual group(s) of the replicate stage(s)
- TriosNumber_initial: the number of people in the trios group(s) of the initial stage(s)
- TriosNumber_replicate: the number of people in the trios group(s) of the replicate stage(s)

An Item refers to a single Cohort and vice versa, while a Cohort can be linked to more than one Ancestries. On the contrary, an Ancestry refers to a single Cohort. For details about the cardinalities of the conceptual model, please refer to Figure 5.1.

The entity Ancestry includes the following attributes:

- AncestryId: the identifier of the current ancestry
- SourceId: stands for the study accession of the Item to which the Ancestry refers

- BroadAncestralCategory: the broad ancestral category to which the individuals in the sample belong
- CountryOfOrigin: the country from which the individuals in the linked `Cohort` come from
- CountryOfRecruitment: the country from which the individuals in the linked `Cohort` are picked up to participate in the GWA study
- NumberOfIndividuals: the sum of all the individuals to which the current `Ancestry` refers. It is obtained by summing the cases, controls, individuals or trios of both initial or replicate stages that participate in the current `Ancestry`

To clarify the cardinality between the entities `Cohort` and `Ancestry` following (see Table 5.1 and Table 5.2) is reported how the two entities are filled with data about the study accession "GCST005538".

Table 5.1: The entry of this table refers to the content of the `Cohort` linked to the `Item` with the study accession "GCST005538".

CohortId	ItemId	TraitName	CaseNumber_initial	CaseNumber_replicate
1309	1308	Sarcoidosis	1726	2693

ControlNumber_initial	ControlNumber_replicate	Ind.Number_initial	Ind.Number_replicate	TriosNum_initial	TriosNum_replicate
5482	6814	0	0	0	0

The attributes "cohortId" in the `Ancestry` table and "itemId" in the `Cohort` table are the foreign keys to reproduce the relations between the two tables. From the example tables is clear that each `Ancestry` entry differs from all the others for at least one attribute. As double check for the integrity of the tables, we can see that the sum of the NumberOfIndividuals fields of `Ancestry` that is 16,715 corresponds to the sum of the fields of the `Cohort` table. The item with study accession "GCST005538" refers to the cohort with Id "1309", which one refers to three different ancestries "3997", "3998" and "3999". From the tables we can see that 7,208 people have been picked up in Germany to participate to this GWA study; unfortunately we don't know how many of them are cases or controls neither their division into initial and replication stage. On the other hand we know the division of cases and controls into the two stages, but we don't know the ancestral composition of each of the previous partitions.

Table 5.2: The entries of this table refer to the content of the `Ancestries` linked to the `Cohort` in Table 5.1 with the sourceId attribute "GCST005538".

AncestryId	CohortId	BroadAncestralCategory	CountryOfOrigin	CountryOfRecruitment	NumberOfIndividuals
3997	1309	African American or Afro-Caribbean	NR	U.S.	1657
3998	1309	European	NR	Germany	7208
3999	1309	European	NR	Czech Republic, Germany, Sweden	7850

5.1.4 Mapping metadata

In section 5.1.3 is presented the new Genomic Conceptual Model, with the aim to fit also the metadata of the GWAS Catalog and FinnGen repositories. In Chapter 4 are listed all the attributes of the two genomic data sources, while in sections 5.1.1 and 5.1.2 of this chapter is provided a rough partition of the attributes taking into account the entities of the New GCM.

Obviously some of the original attributes cannot be used as they are to fill the GCM, but they need some extraction and elaboration steps. Moreover, some attributes are manually added since they are not contained in the downloaded data.

GWAS Catalog

Many Catalog attributes cannot be used without transforming them to fill the New GCM. Following is provided, one entity at a time, the list of the attributes that can be filled with the attributes of the Catalog, both using the raw ones or by transforming or extracting them. In this latter case is illustrated the transformation. Each item of the list is of type <attribute of GCM >: <source-specific attribute >

Next to the GCM attributes is annotated if they are filled with raw [R] or derived or manually added [M] attributes. Next to the source-specific attributes is annotated from which file(s) they come from: [S] for Studies, [AS] for Associations and [AN] for Ancestry.

Entity Ancestry

- broadAncestralCategory [R]: BROAD ANCESTRAL CATEGORY [AN]
- countryOfOrigin [R]: COUNTRY OF ORIGIN [AN]

- countryOfRecruitment [R]: COUNTRY OF RECRUITMENT [AN]
- numberOfIndividuals [R]: NUMBER OF INDIVIDUALS [AN]
- sourceId [R]: STUDY ACCESSION [AS][S][AN]

Entity Cohort

All the attributes marked with [M] of this entity are obtained by extracting some information from the reported attributes. For a practical example about the extraction please refer to Table 5.3.

- traitName [R]: MAPPED TRAIT [AS][S]
- caseNumber_initial [M]: INITIAL SAMPLE DESCRIPTION [AS][S][AN]
- controlNumber_initial [M]: INITIAL SAMPLE DESCRIPTION [AS][S][AN]
- individualNumber_initial [M]: INITIAL SAMPLE DESCRIPTION [AS][S][AN]
- triosNumber_initial [M]: INITIAL SAMPLE DESCRIPTION [AS][S][AN]
- caseNumber_replicate [M]: REPLICATION SAMPLE DESCRIPTION [AS][S][AN]
- controlNumber_replicate [M]: REPLICATION SAMPLE DESCRIPTION [AS][S][AN]
- individualNumber_replicate [M]: REPLICATION SAMPLE DESCRIPTION [AS][S][AN]
- triosNumber_replicate [M]: REPLICATION SAMPLE DESCRIPTION [AS][S][AN]
- sourceId [R]: STUDY ACCESSION [AS][S][AN]

Table 5.3: Many attributes of the entity `Cohort` are extracted from source-specific attributes. Here is reported a simple example about the `Item` with study accession "GCST005538". For more details about the implementation refer to Chapter 6.
INITIAL SAMPLE DESCRIPTION: 1,726 European ancestry cases, 5,482 European ancestry controls
REPLICATION SAMPLE DESCRIPTION: 1,912 European ancestry cases, 5,938 European ancestry controls, 781 African American cases, 876 African American controls

CaseNumber_ initial	ControlNumber_ initial	CaseNumber_ replicate	ControlNumber_ replicate
1726	5,482	1,912 + 781	5,938 + 876

Entity CaseStudy

- sourceId [R]: PUBMEDID [AS][S][AN]
- sourceSite [R]: STUDY [AS][S]
- externalRef [R]: LINK [AS][S]

Entity Project

- programName [M]: “GWAS Catalog”
- projectName [M]: “GWAS Catalog”

Entity ExperimentType

- technique [R]: GENOTYPING_TECHNOLOGY [AS][S]

Entity Datasets

- name [M]: “gwas”
- dataType [M]: “gwas”
- format [M]: “gdm”
- assembly [M]: “GRCh38”
- isAnn [M]: “false”

Entity Item

- sourceId [R]: STUDY ACCESSION [AS][S][AN]
- size [M]: <size of the file >
- date [M]: <download date of the file >
- checksum [M]: <checksum of the file (computed) >
- platform [R]: PLATFORM (SNPS PASSING QC) [AS][S]
- fileName[M]: STUDY ACCESSION [AS][S][AN] + “.gdm”

FinnGen

The FinnGen repository have less metadata than GWAS Catalog and they require less transformations. FinnGen data has no information about the `Ancestry` of people that participate in the studies. The only information available is that people are picked up from Finland. Following is provided, one entity at a time, the list of the pairs <GCM attribute >: <FinnGen-specific attribute >. All attributes source-specific are taken from the Manifest file, while next to the GCM attributes is marked if they are filled with raw data [R] ore derived and manually extracted [M].

Entity Ancestry

- countryOfRecruitment [M]: "Finland"
- numberOfIndividuals [M]: n_cases + n_controls
- sourceId [R]: phenocode

Entity Cohort

The FinnGen repository does not provide information about how the cohorts are composed for the initial or replication stages. The only information available is the cardinality of the cases and controls groups. We treat the FinnGen studies as if they were composed only of the initial stage. For this reason all the GCM attributes that refer to the replication stage remain empty.

Moreover, the cohorts can be composed only of cases and controls and not of individuals neither trios.

- traitName [R]: name
- caseNumber_initial [R]: n_cases
- controlNumber_initial [R]: n_controls
- sourceId [R]: phenocode

Entity CaseStudy

- sourceId [R]: phenocode
- sourceSite [M]: "https://www.finngen.fi/en"
- externalRef [R]: path_https

Entity Project

- programName [M]: "FinnGen"
- projectName [M]: "FinnGen"

Entity ExperimentType

- technique [M]: “FinnGen_technique”

Entity Dataset

- name [M]: “FinnGen”
- dataType [M]: “gwas”
- format [M]: “gdm”
- assembly [M]: “GRCh38”
- isAnn [M]: “false”

Entity Item

- sourceId [R]: phenocode
- size [M]: <size of the file >
- date [M]: <download date of the file >
- checksum [M]: <checksum of the file (computed) >
- fileName[M]: phenocode + “.gdm”

5.1.5 The metadata ”traitName”

GWAS studies follow the phenotypes-first approach. The participants of these studies are classified according to their clinical manifestations. When looking at GWAS studies, the phenotype is one of the most interesting attribute candidate. During this thesis, the two GWAS sources GWAS Catalog and FinnGen have been integrated both into the META-BASE architecture, allowing to create queries upon both sources. The feature of GWAS studies is to search for SNPs given a phenotype. This is the reason why is interesting to understand the set of phenotypes present in both sources.

All traits in GWAS Catalog are mapped over the EFO ontology [6]. Traits in the GWAS Catalog are highly diverse and include diseases, e.g. Type II diabetes, disease markers, e.g. measurements of blood glucose concentration, and non-clinical phenotypes, e.g. hair color. The Experimental Factor Ontology was chosen as the ontology to represent GWAS Catalog traits as it is highly adaptable and extensible. It is freely available in OWL format from the EFO website and can be browsed in the Ontology Lookup Service. At the moment of writing (March 2021), the GWAS Catalog contains 2413 different traits from the EFO ontology. Each study is characterized by one or more traits contained into the source-specific attribute “MAPPED TRAIT”, comma separated.

FinnGen phenotypes are harmonized over the International Classification of Diseases (ICD) revisions 8, 9 and 10, cancer-specific ICD-O-3, (NOMESCO) procedure codes, Finnish-specific Social Insurance Institute (KELA) drug reimbursement codes and ATC-codes [9]. The latest release at the moment of writing (May 2021) is the fifth. In its manifest are listed all the files available, each with its corresponding phenotype. The fifth release contains 2804 different phenotypes.

To create integrated queries over the two genomic sources, is useful to understand which ones phenotypes are shared between the sources. It is not immediate since, unfortunately, the phenotypes are mapped over different ontologies. Applying exact-matching between the list of phenotypes of the two sources, only 94 traits are found to be shared between both of them. A simple graphical representation of the intersection of the sets of phenotypes is provided in Figure 5.2.

For many traits, like “schizophrenia” or “asthma”, both in GWAS Catalog and FinnGen are present many complex traits, but with exact matching only few of them are spotted. In Figure 5.3 are reported all the phenotypes related to “asthma” that are present in GWAS Catalog and only few of the ones from FinnGen (10 out of 37 total, for graphical reasons). There are some correspondences between the traits of the two sources, but using an algorithm that performs exact matching between strings, only one trait is spotted to be in common.

In Figure 5.4 is reported another example. In both tables are reported all the phenotypes resulting by searching for word “schizophrenia”. Also in this example, only one common phenotype is spotted using exact matching, but more correspondences can be found manually.

Mapping two sets of phenotypes coming from different ontologies is a very complex effort and requires expert in the field of biology and medicine. That effort is out of the goal of this thesis. The study of the common traits is a preliminary step for the creation of some integrated queries over both GWAS sources, presented in Chapter 7. To spot inexact common traits is enough to take one of the 94 shared phenotypes and to look into the two sources the entries that contain that string (string containment), as reported in Figures 5.3 and 5.4.

5.2 Data Integration

5.2.1 Transforming region data

The main effort of the META-BASE architecture is to provide metadata integration between multiple genomic sources. Each source has its own conceptual schema and, thanks to the sequential steps introduced in section 3.4 (`DOWNLOAD`, `TRANSFORM`, `CLEAN`, `MAP`, `NORMALIZE-ENRICH` and `INTEGRITY CHECK`), they are all mapped into the Genomic Conceptual Model [2] to achieve the integration.

The data that are processed through this pipeline are ready to be queried using the GenoMetric Query Language [18], introduced in section 3.3. Using

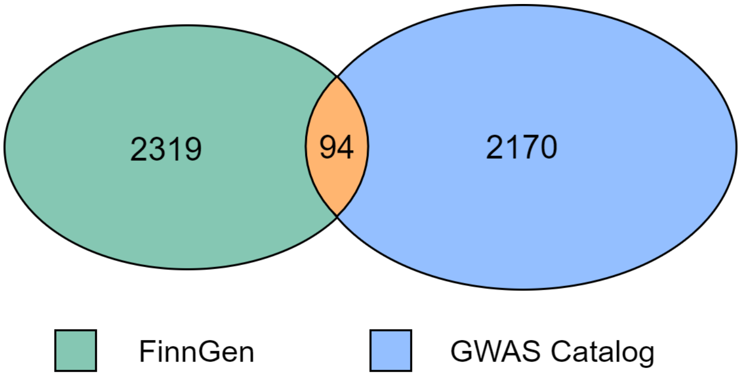
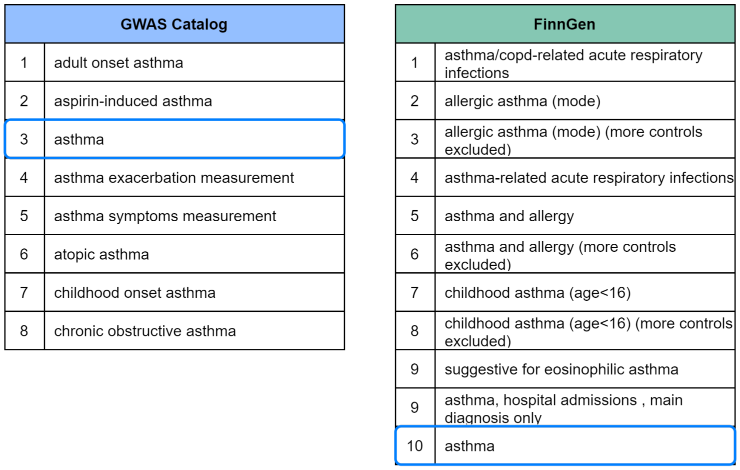


Figure 5.2: In this figure is reported the intersection of the sets of phenotypes of the two sources GWAS Catalog and FinnGen. The intersection is obtained through exact matching of the two sets and it represents a small portion of both of them. To spot more traits in common is required a complex integration effort, driven by biological and medicine experts.



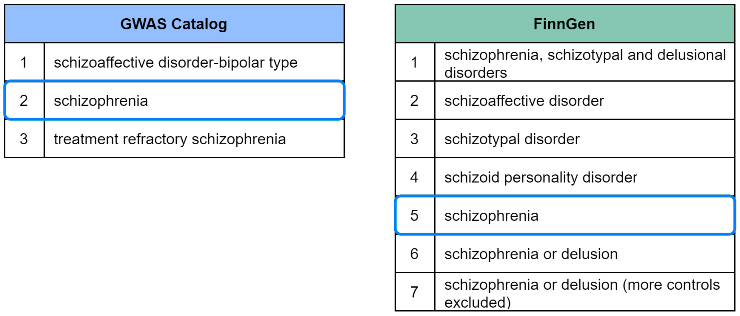
GWAS Catalog	
1	adult onset asthma
2	aspirin-induced asthma
3	asthma
4	asthma exacerbation measurement
5	asthma symptoms measurement
6	atopic asthma
7	childhood onset asthma
8	chronic obstructive asthma

FinnGen	
1	asthma/copd-related acute respiratory infections
2	allergic asthma (mode)
3	allergic asthma (mode) (more controls excluded)
4	asthma-related acute respiratory infections
5	asthma and allergy
6	asthma and allergy (more controls excluded)
7	childhood asthma (age<16)
8	childhood asthma (age<16) (more controls excluded)
9	suggestive for eosinophilic asthma
9	asthma, hospital admissions , main diagnosis only
10	asthma

Figure 5.3: In this figure are presented the troubles when trying to spot shared traits between the two GWAS sources related to "asthma". In the blue table are reported all the 8 phenotypes of GWAS Catalog while in the green one are reported only 10 of them out of 37 total, for graphical reasons. Only one trait is shared, all the others require domain experts to be mapped.

this query language, multiple genomic sources can be queried by specifying the values of both metadata and region attributes.

Unlike the metadata which share a common schema, the region attributes of multiple sources are not guaranteed to have a shared one. The only constraint about region data comes from the Genomic Data Model [19] presented in section



	GWAS Catalog
1	schizoaffective disorder-bipolar type
2	schizophrenia
3	treatment refractory schizophrenia

	FinnGen
1	schizophrenia, schizotypal and delusional disorders
2	schizoaffective disorder
3	schizotypal disorder
4	schizoid personality disorder
5	schizophrenia
6	schizophrenia or delusion
7	schizophrenia or delusion (more controls excluded)

Figure 5.4: In this figure are presented the difficulties when trying to spot shared traits related to "schizophrenia". In the blue table are reported the phenotypes of GWAS Catalog while the green table is dedicated to FinnGen. Only one trait is shared, all the others require domain experts to be mapped.

3.1. According the GDM, each region in the repository is uniquely identified by the four coordinates *chrom*, *start*, *end* and *strand*.

Genomic data from GWAS Catalog and FinnGen are all GWAS studies, so the schemes of their region data share some attributes. In fact the goal of a GWAS study is, for a given position, to identify the allele which is "causal" for a given phenotype. The schemes of both sources have of course information about the reference allele, the causal allele and about the "strength" of the association found. This information are contained into attributes with different names and often with different formats.

For this reasons, to prepare the region data of both sources for being mapped in section 5.2.2, some of the attributes require a transformation. Once the integration between the region schemes is provided, it is possible to create GMQL queries obtaining regions from both sources.

GWAS Catalog

In Figure 5.5 are reported some relevant attributes of GWAS Catalog region data. It includes all the attributes that are modified (identified by the *orange* color) and some of the unchanged attributes (identified using the *light blue* color).

The first attribute that has been modified is `Chr_id`. It contains the number of the chromosome in which the current region is located, but the corresponding attribute `chrom` of the GDM contains the number of the chromosome preceded by the prefix "chr". The modification, consequently, consists of the addition of the prefix.

The attribute `Chr_pos` contains the position in the specified chromosome, expressed in base pair, of the identified SNP. Since a SNP is a region made of a single nucleotide, it has length one. Thus the attribute `Chr_pos` is used to fill the attributes `start` and `end` of the GDM. The attribute `start` contains the same value of `Chr_pos`, while the attribute `end` contains the starting position of the region increased by one base pair.

The attribute strand of the GDM is not contained into GWAS Catalog. It identifies which one of the two strands of the DNA double helix hosts the current genomic region. Since a SNP regards both the two strands, the value of the corresponding attribute is filled with the wildcard character "*".

The last modification concerns the attribute Strongest snp-risk allele, which contains two informations: the identifier of the current SNP ("rs" ID) and the risk allele. Since the "rs" identifier of the SNP is contained also in the attribute Snps and in order to make this attribute matching with the attribute alt of FinnGen (for details about this association please refer to section 5.2.2), is kept only the risk allele. Thus the attribute Strongest snp-risk allele, after the transformation, holds only the allele which is causal for the phenotype under consideration.

FinnGen

The Figure 5.6 reports some relevant region attributes of FinnGen dataset. The attributes identified by the color *orange* are the ones that are changed while the *light blue* attributes remain unchanged.

The first attribute that has been modified is #chrom; it contains the number of the chromosome in which the current SNP is located. The corresponding attribute chrom of the GDM contains the same information but it's preceded by the prefix "chr". Therefore, the attribute #chrom is transformed into chrom by adding the prefix "chr" before the chromosome number.

Then is modified the attribute pos, using its value to fill the attributes start and end of the GDM. The attribute pos holds the position expressed in base pairs of the current SNP over the identified chromosome. Since a SNP is a region of length one base pair, the attribute start coincides with pos, while the attribute end is filled with the value of pos increased by one base pair.

The last modification concerns the attribute strand of the GDM. It is not present in FinnGen region attributes and it spots which one of the two nucleotide chains of the double helix contains the current region. Since a SNP concerns both strands, the corresponding attribute is filled with the wildcard character "*".

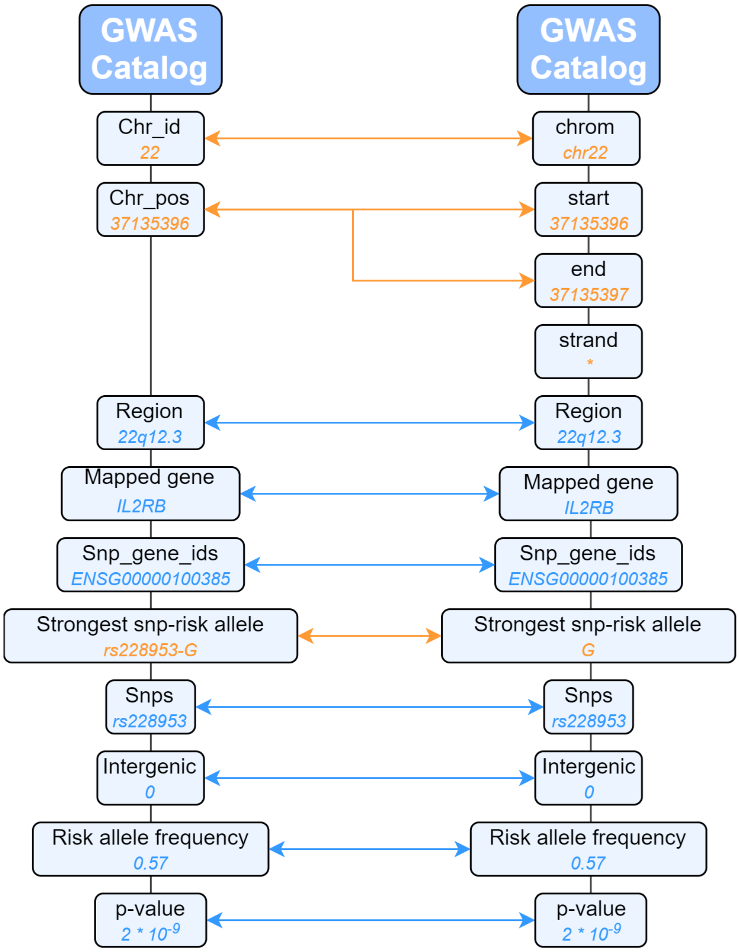


Figure 5.5: This diagram represents how the source-specific region attributes of GWAS Catalog are transformed to achieve data interoperability between the two genomic sources GWAS Catalog and FinnGen and to fulfil the Genomic Data Model format. Each attribute is coupled with an example value; the color *orange* represents values that are modified, while the *light blue* represents unchanged values. Please note that in this diagram are reported only some relevant region attributes of GWAS Catalog; for the full list refer to section 4.1.

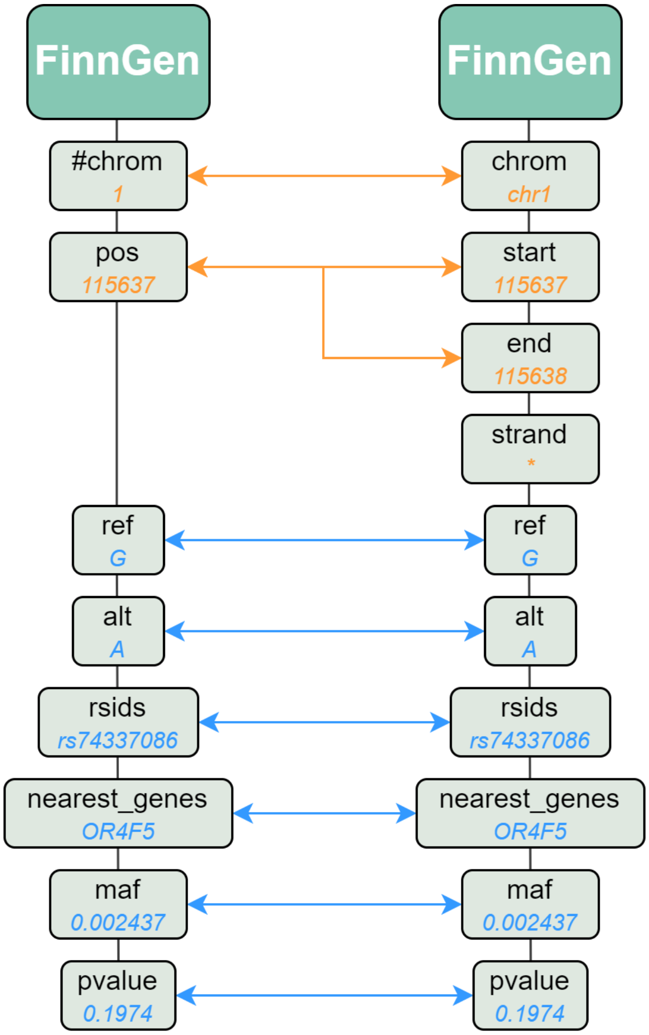


Figure 5.6: This diagram represents how the FinnGen source-specific region attributes are transformed to achieve data integration with the genomic source GWAS Catalog and to fulfil the Genomic Data Model format. Each attribute is coupled with an example value; the color *orange* represents values that are modified, while the *light blue* represents values that remain unchanged. In this figure are reported only some relevant region attributes of FinnGen, for the full list the reader is invited to read section 4.2.

5.2.2 Region attributes correspondences

Back in this chapter, in section 5.2.1, is shown the process through which the region attributes of the two genomic sources GWAS Catalog and FinnGen are obtained from the source-specific attributes; in Figure 5.8 are highlighted some correspondences between region attributes of the two genomic sources though arrows. The main goal of the META-BASE architecture is to integrate the metadata of all the genomic sources that it encloses. The focus is on metadata, not on region attributes. Actually, all the data included in the architecture need to fit the Genomic Data Model [19]. This is very important to guarantee the data interoperability and integrity.

The META-BASE architecture [1] has been developed and expanded in this thesis with the goal of mapping all the metadata against the Genomic Conceptual Model [2]. Moreover, after all the steps of the architecture have been executed, the data (region and metadata) are uploaded into the GMQL architecture. This query language, presented in section 3.3 is based both on metadata and on region attributes. The user can query the genomic data specifying the values of some metadata, as well as some region coordinates or other region features. The four coordinates essential in the Genomic Data Model are *chrom*, *start*, *end* and *strand*. The other region attributes are source-specific and their integration is out of the goal of the META-BASE architecture.

The GWAS Catalog and the FinnGen dataset are both Genome-Wide Association Study repositories, so there is a correspondence between some of their region attributes. The GWAS studies outcomes are, for a specified phenotype, all the SNPs (regions of length one base-pair) associated to it. Each entry of a GWAS regions file corresponds to a SNP. All the attributes are useful to identify the region over the genome and to express the "importance" of the association between the current SNP and the phenotype under consideration.

The position over the genome is expressed though the four coordinates *chrom*, *start*, *end* and *strand*. Since the regions are SNPs, the *end* attributes is the *start* increased by one. The four coordinates attributes are present in both genomic sources, so they are connected though arrows 1, 2, 3 and 4 in Figure 5.8.

The human genome, as well as genomes of many living species, have been sequenced and many genes have been identified. The genes are the basic unit of heredity, so they are sequences of nucleotides that are inherited together and they contain the information to synthesize RNA or proteins. The first big attempt to identify human genes is the Human Genome Project [5], back in 1990 by the National Institute of Health. Now all the known genes can be browsed through many repositories, between them the web interface of the National Center for Biotechnology Information available at NCBI NIH website. The description of the source-specific attributes of the two genomic sources is available at [6] and [9].

The main difference between the two schemes is about the gene in which the mutation is located. In GWAS Catalog is specified whether the SNP is inside a gene or between genes. In the former case, the single gene is specified into the attributes *Mapped gene* and *Snp_gene_ids*. In the latter case, the genes that enclose the SNP are listed into the attribute *Mapped gene*. Then are indicated the identifiers of the genes in the upstream and downstream of the SNP (respectively attributes *Upstream_gene_id* and *Downstream_gene_id*), and the distance between the SNP and the reported genes (respectively attributes *Upstream_gene_distance* and *Downstream_gene_distance*). On the other hand, the FinnGen repository is less precise and does not distinguish these two cases. The attribute *nearest_genes* encloses both the cases, without specifying if the SNP is within a gene or between genes. Downstream of this long consideration, the attributes that can be considered having the same meaning are *Mapped gene* of GWAS Catalog with *nearest_genes* of FinnGen (arrow 5).

Another difference from the two sources is the terminology used to identify the alleles of the mutations associated with the phenotype under consideration. As shown in Figure 5.7 the studies are conduced comparing two groups of people, the `cases` and the `controls` groups. If, for a particular position over the genome, the `cases` group has an allele significantly different from the `controls` group, then the most frequent allele for that position in the `cases` group is said to be the "risk allele". Another way to refer to the risk allele is "alternative allele" or "effect allele". The "reference" allele, on the other hand, is the most frequent one in the `controls` group. At the light of this consideration, the attribute *Strongest snp-risk allele* of GWAS Catalog corresponds to the attribute

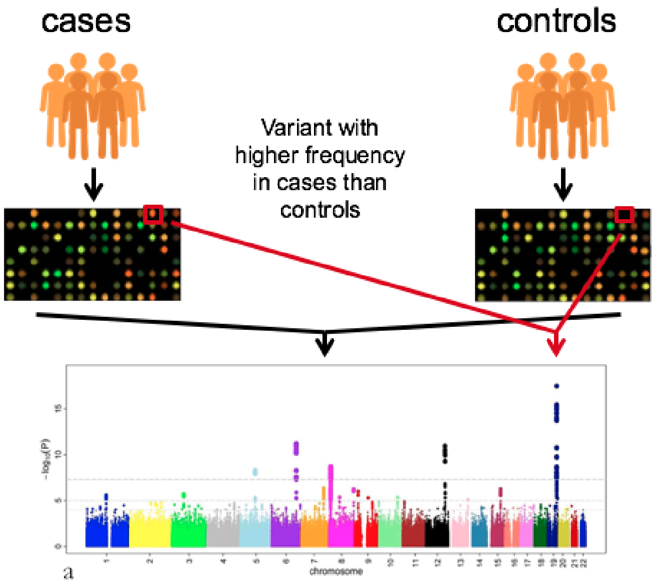


Figure 5.7: GWAS studies compare the `cases` and the `controls` groups to find the "risk allele" or "effect allele" for the phenotype under consideration. The "risk allele" is found at a higher frequency in `cases` rather than in `controls` group.

alt of FinnGen (arrow 6). GWAS Catalog does not specify which is the reference allele.

Also the attribute *Snps* of GWAS Catalog and *rsids* of FinnGen have the same meaning, that is the identifier of the variant represented by the current entry of the regions file (arrow 7). The identifiers are expressed using the same reference, that it the “rsId”. All the known SNPs can be browsed through the web interface of the National Center for Biotechnology Information available at NCBI NIH website. After having identified the position of the SNP through the four coordinates and the allele which is associated to the phenotype under consideration, the remaining attributes provide some statistics about the “strength” of the association found.

The terminology used in the two genomic sources is a bit different, but with some analogies. In GWAS Catalog is present the attribute *Risk allele frequency*, which of course refers to the frequency of the risk allele. In FinnGen is present the attribuite *maf* which stands for “minor allele ferquency”. According to its definition, the minor allele is “the second most common allele that occurs in a given population”. In many GWAS studies on complex diseases, the minor allele can be considered the risk allele [16], despite their definitions differ. So the attribute *Risk allele frequency* of GWAS Catalog and the attribute *maf* of FinnGen can be considered of equivalent meaning, so they are linked with arrow 8 in Figure 5.8.

The last two correspondences are obvious and regard the attributes *p-value* and *pvalue* (arrow 9) and the attributes *Or or beta* and *beta* (arrow 10).

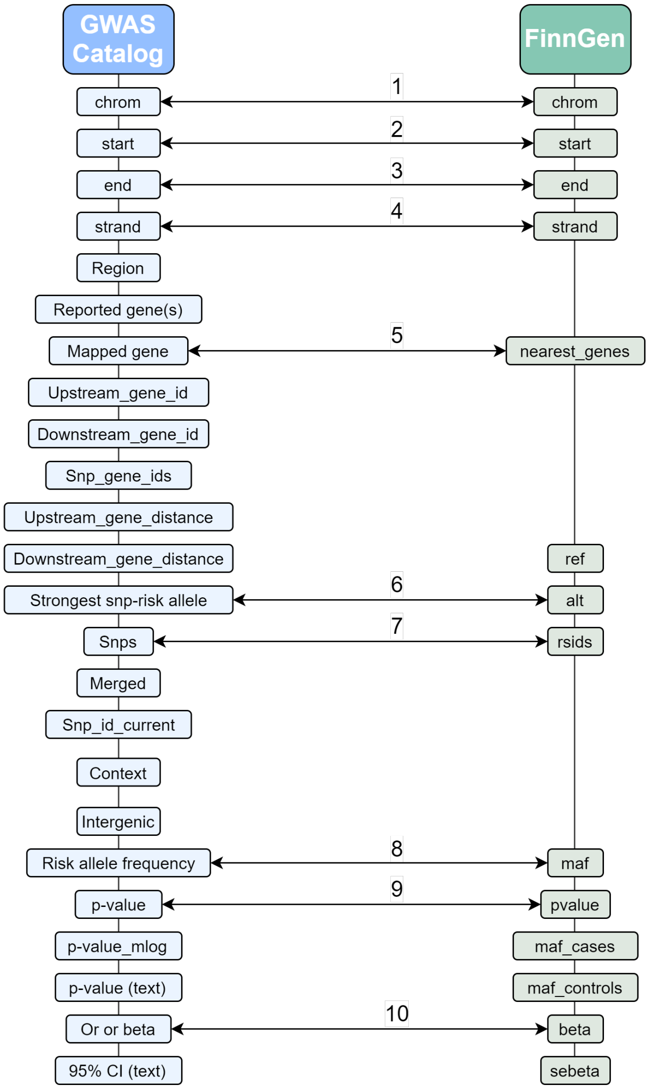


Figure 5.8: The items in this figure are the region attributes of the genomic sources GWAS Catalog and FinnGen. They are the outcome of the `transformation` phase of the META-BASE architecture. The attributes are already introduced in section 5.2.1. In this section is presented a possible correspondence between the two sources, represented by the arrows in this figure.

Chapter 6

Implementation

In this chapter is described the implementation of the Metadata-Manager [1] module for the genomic sources GWAS Catalog [3] and FinnGen [9]. The existing architecture is described in Chapter 3, where the steps of the pipeline are shown. The proposed description in that chapter doesn't go into the implementation details of the architecture.

The project is written with the Scala programming language [28], a general-purpose programming language providing support for both object-oriented programming and functional programming. It is compiled to Java bytecode, so that the resulting executable code runs on a Java virtual machine. Moreover, the Scala language provides interoperability with Java.

The Metadata-Manager implementation code is publicly available in the GitHub repository `https://github.com/DEIB-GECO/Metadata-Manager`. The project is composed by multiple steps: `Data Download`, `Transformation`, `Cleaning`, `Mapping`, `Normalization`, `Enrichment` and `Integrity Checker`. The source-specific steps are only the former four; after the `Mapper` step the data are instances of the Genomic Data Model [19] and the metadata are mapped as instances of the Genomic Conceptual Model [2]. When the `Mapper` step has been executed, data from different genomic sources share the same schema and format, so they can be queried using the GMQL language [18].

In Chapter 4 are described the data structures of the genomic sources GWAS Catalog and FinnGen. In Chapter 5 are presented the modelling tasks faced to model the two new sources into the GCM also by introducing the two entities `Ancestry` and `Cohort`.

The work is carried on as follow: in section 6.1 are described the classes and methods implemented to perform the `Download` and `Transformation` steps for GWAS Catalog; in section 6.2 are described the methods and classes implemented to perform the first two steps for FinnGen; in section 6.3 are presented the methods and classes required to develop the `Mapper` step for both the two sources.

The `Mapper` step is shared between the two sources, so the classes have to be implemented only once for both GWAS Catalog and FinnGen.

6.1 Downloader & Transformer for GWAS Catalog

GWAS Catalog summary statistics are available as three files `Ancestry`, `Studies` and `Associations`. They are tab-separated values files and they are downloaded from the dedicated FTP server (`ftp.ebi.ac.uk`).

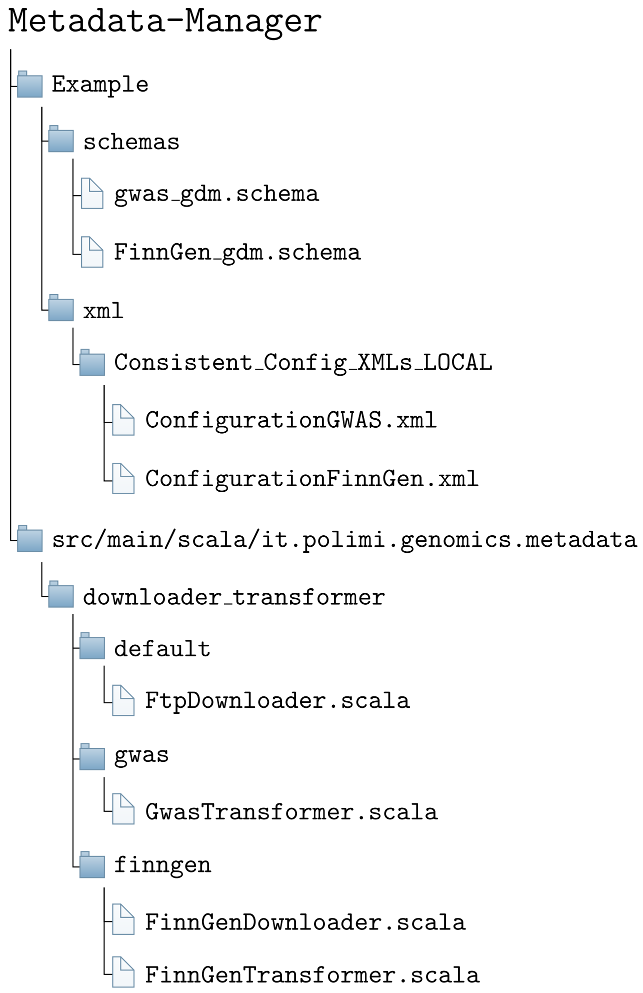


Figure 6.1: These Scala classes and xml files are the ones relevant for the execution of the `download` and `transformation` stages of the Metadata-Manager program. In this diagram appear only the files implemented during this thesis or the ones off-the-shelf, like *FtpDownloader.scala*. Many other used classes are not reported in this schema but their are important as well (e.g. *Program.scala* and *FileDatabase.scala*).

In Figure 6.1 are shown the Scala classes and files interested in these two former stages. In particular, the class *FtpDownloader.scala* has already been implemented in the original architecture since it is used to download other genomic sources as TCGA.

The execution of the program is driven by the file *ConfigurationGWAS.xml*. It contains many information, among which:

- the credentials of the database in which the data are traced
- the stage to be executed
- the url of the FTP server through which the summary statistics are downloaded
- the local paths to the classes *FtpDownloader.scala* and *GwasTransformer.scala*
- the local path to reach the gdm schema of the new source
- the regular expression to locate the three files to be downloaded, in the server previously specified

The Figure 6.2 briefly shows the flow through which the files of the GWAS Catalog are downloaded.

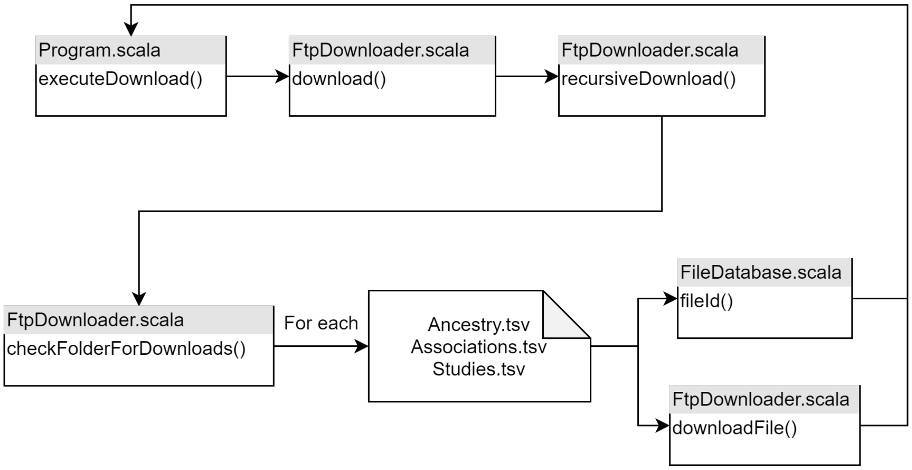


Figure 6.2: This is the schematic execution flow of the `download` phase of the GWAS Catalog genomic source. For each item of the flow is indicated the class and the method, omitting the input parameters for graphical reasons. For the full description of the flow, the reader is invited to read section 6.1.

The main class of the program is called *Program.scala*. Assuming that in the configuration file has been selected the `download` phase, the method *executeDownload()* is run. It calls the *download()* method of the *FtpDownloader.scala* class, which recursively calls methods *recursiveDownload()* and *checkFolderForDownloads()* of the same class. This last method uses the regular expression in the configuration file to list the files to be downloaded from the FTP server. Than it has a cycle that, for each of the three selected files, launches the method *fileId()* of the class *FileDatabase.scala* and the method *downloadFile()* of the class *FtpDownloader.scala*. The former method adds the file to the database while the latter downloads the file on a local copy on the machine. The same method also checks that the download goes smooth, by computing an hash while downloading and by compare it with the one computed on the local copy.

The database which references the downloaded files is called *gmql_importer* and is different from the database filled in the `Mapping` stage, which is called *gmql_metadata*. The schema of the *gmql_metadata* is the new GCM presented in section 5.1.3. The *gmql_importer* has a different schema and it keeps information about the files during the `download` and `transformation` stages and some logs about the execution of these two phases.

When all the files are downloaded locally and are traced on the database, the execution flow goes back to the class *Program.scala* that computes some statistics about the execution and prints them on the console. After the `download` phase for GWAS Catalog as been executed, in the selected path appear the files as follow:

```
gwas/latest
└ Downloads
    ├ gwas-catalog-ancestry.tsv
    ├ gwas-catalog-associations_ontology-annotated.tsv
    └ gwas-catalog-studies_ontology-annotated.tsv
```

while in the *gmql_importer* database the same files appear coupled with some information, among which:

file_id	dataset_id	name	stage	status
36287	102	ancestry.tsv	DOWNLOAD	UPDATED
36288	102	associations.tsv	DOWNLOAD	UPDATED
36289	102	studies.tsv	DOWNLOAD	UPDATED

The execution flow of the `transformation` phase is shown in Figure 6.3.

After setting the `transformation` label in the configuration file, the method *executeLevel()* of the class *Program.scala* calls the method *execute()* of the class *TransformerStep.scala*. This method generates the candidate names of the files to be created during the `transformation` phase, by calling the method *getCandidateNames()* of the class *GwasTransformer.scala*. Each of the new candidates

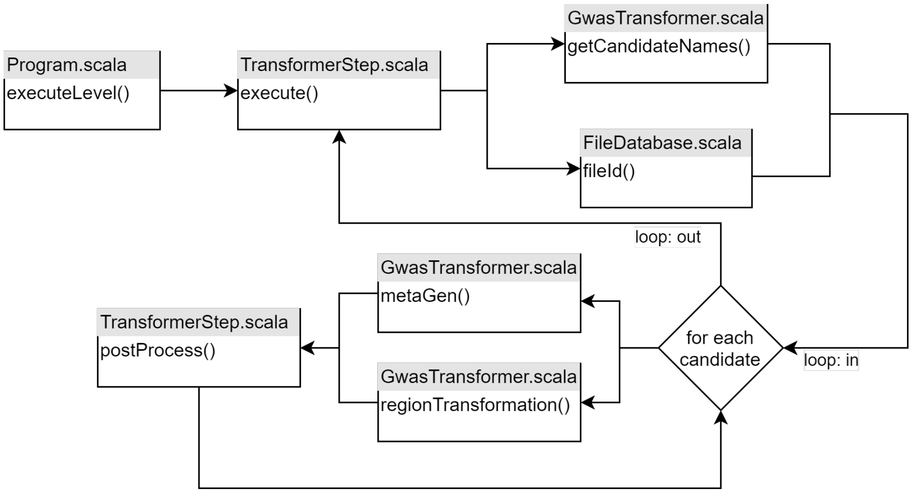


Figure 6.3: This is the execution flow of the `transformation` phase of the GWAS Catalog genomic source. For the full description of the flow, the reader is invited to read section 6.1.

is added to the database *gmql_importer*, tracing its status and some other information about the file. This last operation is done by the method *fileId()* of the class *FileDatabase.scala*.

Then the flow turns back to the class *TransformerStep.scala* which, for each candidate file, calls the proper method of the class *GwasTransformer.scala*. If the candidate name ends with “.gdm.meta” is called the method *metaGen()* while if it ends with “.gdm” is called the method *regionTransformation()*.

The method *metaGen()* takes the information about the current study accession from the three downloaded files and write them in the file “study_accession .gdm.meta” as a flat list of < key > < value > pairs of metadata.

The method *regionTransformation()* takes the region data from the file `Associations.tsv` and transforms writing them in the file “study_accession.gdm”. After the transformation, the flow goes on to method *postProcess()* of the class *TransformerStep.scala*. If the current file contains metadata, this method adds some manually curated information like the file size or its download date. Otherwise, if the current file contains region data, it calls the method *checkRegionData()* of the class *GwasTransformer.scala*, which checks the integrity of the region files against the following schema:

```
<?xml version=”1.0” encoding=”UTF−8”?>
<gmqlSchemaCollection name=”gwas”
xmlns=”http://genomic.elet.polimi.it/entities”>
    <gmqlSchema type=”TAB”>
        <field type=”STRING”>chrom</field>
        <field type=”LONG”>start</field>
        <field type="LONG">end</field>
        <field type="CHAR">strand</field>
        <field type="STRING">REGION</field>
        <field type="STRING">REPORTED GENE(S)</field>
        <field type="STRING">MAPPED GENE</field>
        <field type="STRING">UPSTREAM_GENE_ID</field>
        <field type="STRING">DOWNSTREAM_GENE_ID</field>
        <field type="STRING">SNP_GENE_IDS</field>
        <field type="LONG">UPSTREAM_GENE_DISTANCE</field>
        <field type="LONG">DOWNSTREAM_GENE_DISTANCE</field>
        <field type="STRING">STRONGEST SNP–RISK ALLELE</field>
        <field type="STRING">SNPS</field>
        <field type="INTEGER">MERGED</field>
        <field type="LONG">SNP_ID_CURRENT</field>
        <field type="STRING">CONTEXT</field>
        <field type="INTEGER">INTERGENIC</field>
        <field type="DOUBLE">RISK ALLELE FREQUENCY</field>
        <field type="DOUBLE">P–VALUE</field>
        <field type="DOUBLE">PVALUE_MLOG</field>
        <field type="STRING">P–VALUE (TEXT)</field>
        <field type="DOUBLE">OR or BETA</field>
        <field type="STRING">95% CI (TEXT)</field>
    </gmqlSchema>
</gmqlSchemaCollection>
```

The transformation of the region data is simple, and it concerns only the attributes `chrom`, `start`, `end` and `strand`. The other attributes in the schema are written in the region file as they are in the file `Associations.tsv`. The attribute `chrom` is taken from the Catalog attribute "CHR_ID" when present or extracted from the attribute "STRONGEST SNP-RISK ALLELE". The `start` is taken from the Catalog attribute "CHR_POS". The `end` is the `start` increased by one, since the SNPs are region of length one base pair. Finally the `strand` is setted as "*" since there is no information available from the Catalog.

After the `transformation` phase for GWAS Catalog has been executed, in the selected path appear the files as follow:

```
gwas/latest
└ Transformations
   ├ GCST007269.gdm
   ├ GCST007269.gdm.meta
   ├ GCST009379.gdm
   ├ GCST009379.gdm.meta
   ├ ***
   └ gwas.schema
```

while in the *gmql_importer* database the same files appear coupled with some information, among which:

file_id	dataset_id	name	stage	status
36295	102	GCST007269.gdm	TRANSFORM	UPDATED
36296	102	GCST007269.gdm.meta	TRANSFORM	UPDATED
36293	102	GCST009379.gdm	TRANSFORM	UPDATED
36294	102	GCST009379.gdm.meta	TRANSFORM	UPDATED

6.2 Downloader & Transformer for FinnGen

FinnGen summary statistics are available through the URLs that are present in the manifest of the release to be downloaded. At the moment of writing (June 2021) the latest release available is the fifth. Unlike GWAS Catalog that for the `download` stage uses an already implemented Scala class, for FinnGen we have implemented the class *FinnGenDownloader.scala* to fulfil the peculiarities of this genomic source.

In Figure 6.1 are listed the most relevant Scala classes and xml files for the execution of the `download` and `transformation` stages for the FinnGen source. The execution of both phases is driven by the configuration file *ConfigurationFinnGen.xml* which includes some information, between them:

- the local path where to save the downloaded files
- the credentials to authenticate into the databases which trace the files and the metadata mapped against the GCM schema, respectively *gmql_importer* and *gmql_metadata*
- the label referencing the stage to be executed
- the URL where to retrieve the manifest of the latest release

- the local paths to the classes *FinnGenDownloader.scala* and *FinnGenTransformer.scala*
- the local path of the schema for the region data

The rationale is straightforward and it is divided into two subsequent parts. First of all the manifest of the latest release is downloaded. It contains one row for each endpoint available in the database, each one having the URL to download the corresponding summary statistics file. The second phase consists of downloading one file for each endpoint saving them in the specified local folder and tracing them on the database *gmql_importer*.

The Figure 6.4 illustrates briefly the execution flow of the `download` phase for the FinnGen genomic source. The main class is the *Program.scala* and, having properly setted the `download` label in the configuration file, it runs its method *executeDownload()*. This calls the method *download()* of the source-specific class for the current stage to be run, that is *FinnGenDownloader.scala*. First of all it calls the method *getManifest()* of the same class. It uses the URL specified in the configuration file to retrieve the manifest file from the FinnGen database and saves it in the specified local folder. The class *FileDatabase.scala* through the method *fileId()* traces the downloaded manifest to the database *gmql_importer*.

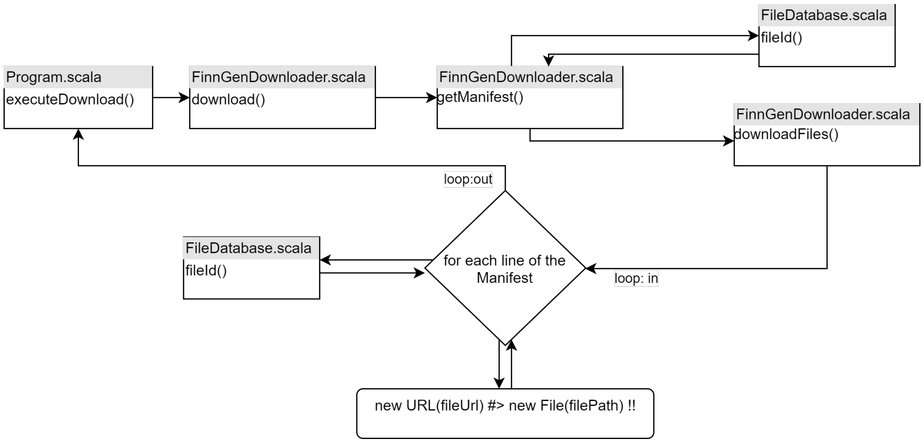


Figure 6.4: This is the schematic execution flow of the `download` phase of the FinnGen genomic source. For each item of the flow is indicated the class and the method, omitting the input parameters for graphical reasons. For the full description of the flow, the reader is invited to read section 6.2.

After getting the manifest, the flow goes on to the method *downloadFiles()* of the class *FinnGenDownloader.scala*. This method reads from the manifest the available endpoints files and, for each of them, it downloads the file and traces it on the database *gmql_importer*. After having downloaded all the endpoints files (or a portion of them if specified properly) the flow goes back to the method *executeDownlaod()* of the main class that computes some statistics about the current execution and prints them on the console. After the `download` phase has been correctly executed, in the apposite local folder appear the following files:

```
FinnGen/R4
└ Downloads
   ├ ManifestR5.tsv
   ├ AB1_ARTHROPOD.gz
   ├ AB1_BACT_BIR_OTHER_INF_AGENTS.gz
   ├ AB1_HELMINTIASES.gz
   └ ***
```

while in the *gmql_importer* database the same files appear coupled with some information, among which:

file_id	dataset_id	name	stage	status
36303	101	ManifestR5.tsv	DOWNLOAD	UPDATED
36304	101	AB1_ARTHROPOD.gz	DOWNLOAD	UPDATED
36305	101	AB1_BACT_BIR_OTHER_INF_AGENTS.gz	DOWNLOAD	UPDATED
36306	101	AB1_HELMINTIASES.gz	DOWNLOAD	UPDATED

The flow of the execution of the `transformation` phase is illustrated in Figure 6.5.

The method *executeLevel()* of the main class launches the *execute()* method of the class *TransformerStep.scala*. This passage works thanks to the configuration of the `transformation` label and to the specification of the path to reach the source-specific transformation class.

The flow proceeds to the method *getCandidateNames()* of the class *FinnGenTransformer.scala*. This method extracts the names of the files that are the output of the `transformation` phase. For each endpoint in the FinnGen database, the candidates are the two files containing the metadata and the region data of the current endpoint. The metadata file is named with the current endpoint name with the extension ".gdm.meta", while the region data file has the extension ".gdm". Each of the candidate file is traced on the database *gmql_importer*.

The flow goes on with a loop over all the candidates just generated. If the file has the extensions ".gdm.meta", the flow goes through the method *metaGen()* of the class *FinnGenTransformer.scala*. It obtains the metadata of the current endpoint from the manifest and writes them in a flat list of pairs <key>

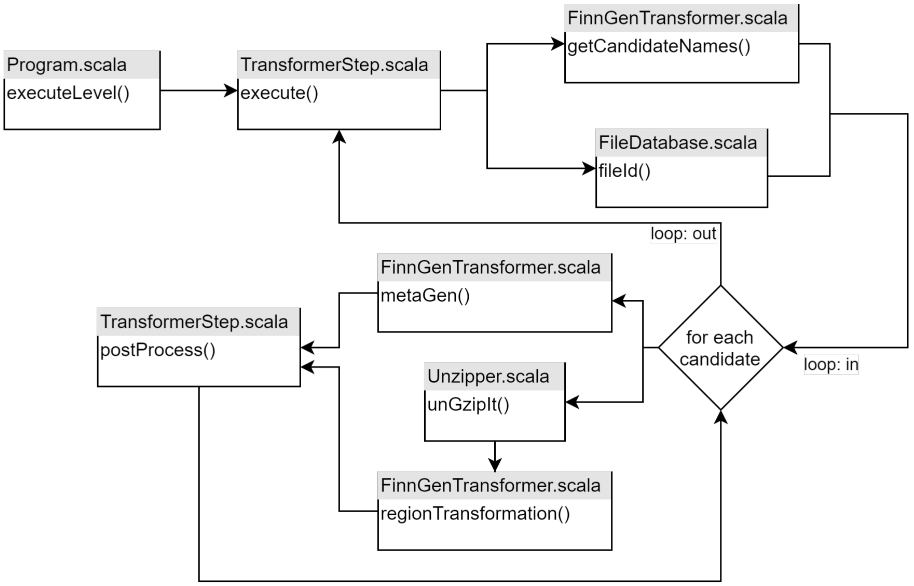


Figure 6.5: This is the execution flow of the `transformation` phase of the FinnGen genomic source. For the full description of the flow, the reader is invited to read section 6.2.

<value>. Otherwise, if the file has the extension ".gdm", the endpoint file is unzipped and then the region data are transformed by the method *regionTransformation()* of the class *FinnGenTransformer.scala.*

The transformation of the region data concerns only the attributes `chrom`, `start`, `end` and `strand`. The remaining attributes in the schema are copied into the region file as they are from the current endpoint file. The attribute `chrom` is taken from the endpoint column "#chrom". The `start` is taken from the attribute "pos". The `end` is the `start` increased by one, since the SNPs are regions of length one base pair. Finally the `strand` is setted as "*" since there is no information available from the FinnGen source.

The method *postProcess()* of the class *TransformerStep.scala* has a different behaviour if the actual file contains region data or metadata. For region data files, it checks their integrity against the following schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<gmqlSchemaCollection name="FinnGen"
xmlns="http://genomic.elet.polimi.it/entities">
    <gmqlSchema type="TAB">
        <field type="STRING">chrom</field>
        <field type="LONG">start</field>
        <field type="LONG">end</field>
        <field type="CHAR">strand</field>
        <field type=”STRING”>ref</field>
        <field type=”STRING”>alt</field>
        <field type=”STRING”>rsids</field>
        <field type=”STRING”>nearest_genes</field>
        <field type=”DOUBLE”>pval</field>
        <field type=”DOUBLE”>beta</field>
        <field type=”DOUBLE”>sebeta</field>
        <field type=”DOUBLE”>maf</field>
        <field type=”DOUBLE”>maf_cases</field>
        <field type=”DOUBLE”>maf_controls</field>
    </gmqlSchema>
</gmqlSchemaCollection>
```

If the current file has the extension “.gdm.meta”, the method *postProcess()* adds some manually curated metadata like the file size or its download date.

The flow ends turning back to the main class, which computes some statistics about the current execution and prints them on the console.

After the `transformation` phase has been correctly carried on, in the specified local path are saved the files as follow:

```
FinnGen/R5
└ Transformations
   ├ AB1_ARTHROPOD.gdm
   ├ AB1_ARTHROPOD.gdm.meta
   ├ AB1_BACT_BIR_OTHER_INF_AGENTS.gdm
   ├ AB1_BACT_BIR_OTHER_INF_AGENTS.gdm.meta
   ├ ***
   └ FinnGen.schema
```

while in the *gmql_importer* database the same files appear coupled with some information, among which:

file_id	dataset_id	name	stage	status
36307	101	AB1_ARTHROPOD.gdm	TRANSFORM	UPDATED
36308	101	AB1_ARTHROPOD.gdm.meta	TRANSFORM	UPDATED
36309	101	AB1_BACT_BIR_OTHER_INF_AGENTS.gdm	TRANSFORM	UPDATED
36310	101	AB1_BACT_BIR_OTHER_INF_AGENTS.gdm.meta	TRANSFORM	UPDATED

6.3 Mapper

After the `download` and `transformation` phases, now it's the turn of the `mapper`. Other genomic sources require a further step before the `mapper`, that is the `clean` phase. For that genomic sources whose metadata are derived from structured json files, many times is appropriate to add the `clean` phase to reduce the complexity of the metadata attributes names. When passing from a structured json file to a metadata file composed of pairs <key><value>, the names of the attributes become very complex, not the ideal input of the `mapper` phase. Since both *GWAS Catalog* and *FinnGen* metadata are extracted from flat files of format tab-separated values, this further step is not required.

In Figure 6.6 are reported the main Scala classes and xml files required to carry on the `mapper` phase for both *GWAS Catalog* and *FinnGen* genomic sources. Unlike the `download` and `transformation` phases that are presented separated for the two sources in section 6.1 and 6.2, the `mapper` is presented in a unique section. In fact this last step is not source-specific, so it boasts the code re-usability feature.

In particular, the entities freshly introduced in the Genomic Conceptual Model in section 5.1.3 are implemented as the Scala traits (i.e. interfaces in Java language) *Ancestry.scala* and *Cohort.scala*. Then all the traits are implemented as the Scala classes *AncestryGwas.scala*, *CohortGwas.scala*, *ItemGwas.scala*, *BioSampleGwas.scala*, *CaseGwas.scala*, *CaseItemGwas.scala*, *DatasetGwas.scala*, *DonorGwas.scala*, *ExperimentTypeGwas.scala*, *GwasTable.scala*, *ProjectGwas.scala*, *ReplicateGwas.scala* and *ReplicateItemGwas.scala*. Furthermore are implemented the classes *AncestryList.scala*, *GwasTableId.scala* and *GwasTables.scala*. The role of this latter list of classes is explained later in this section. The classes that implement the `Biological View` of the GCM remain empty after the `mapper` phase has been executed for GWAS sources.

The class *DbHandler.scala* has been extended by adding all the methods needed to insert or modify the entries for the items `Ancestry` and `Cohort` of the database `gmql_metadata`.

The flow of execution is briefly shown in Figure 6.7.

```
Metadata-Manager
├─ Example/schemas
│  └─ xml
│     ├─ Consistent_Config_XMLs_LOCAL
│     │  ├─ ConfigurationGWAS.xml
│     │  └─ ConfigurationFinnGen.xml
│     ├─ settingsGWAS.xml
│     └─ settingsFinnGen.xml
└─ src/main/scala/it.polimi.genomics.metadata/mapper
   ├─ RemoteDatabase
   │  └─ DbHandler.scala
   ├─ GWAS
   │  ├─ Tables
   │  │  ├─ AncestryGwas.scala
   │  │  ├─ CohortGwas.scala
   │  │  ├─ ItemGwas.scala
   │  │  ├─ ***
   │  │  └─ GwasTable.scala
   │  ├─ Utils
   │  │  └─ AncestryList.scala
   │  ├─ GwasTableId.scala
   │  └─ GwasTables.scala
   ├─ Ancestry.scala
   ├─ Cohort.scala
   └─ Tables.scala
```

Figure 6.6: These are the Scala classes and xml files concerned with the `mapper` step. For the full explanation about their roles, please refer to section 6.3.

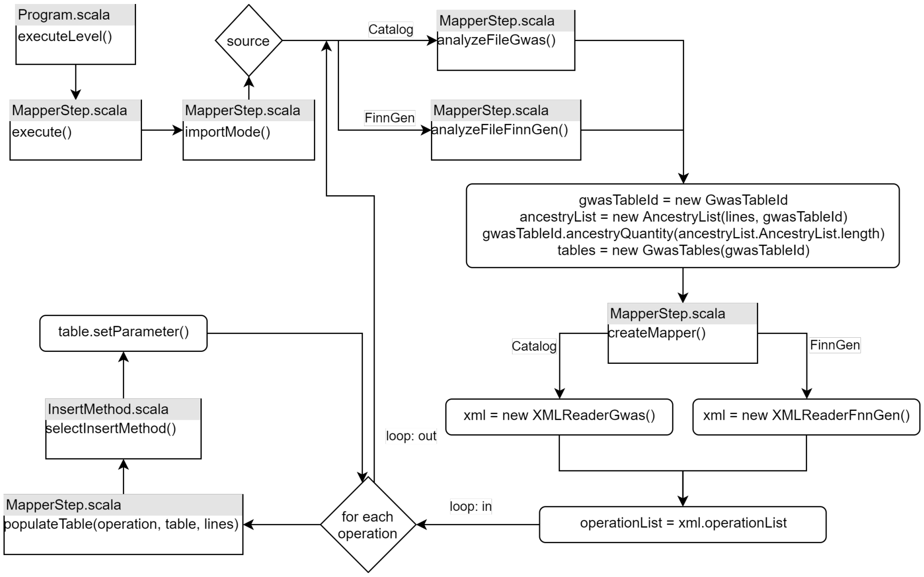


Figure 6.7: This is the execution flow of the `mapper` phase for both the gwas sources. For the full description of the flow, the reader is invited to read section 6.3.

The method *executeLevel()* of the main class *Program.scala*, having properly setted the `mapper` label in the configuration file, launches the method *execute()* of the class *MapperStep.scala*. The flow goes on through the method *importMode()* of the same class, whose behaviour is different respect the current gwas source in execution. If the program is running with GWAS Catalog, is called the method *analyzeFileGwas()* of the class *MapperStep.scala*, while if it is running with FinnGen, is called the method *analyzeFileFinnGen()* of the same class. The behaviour of these two methods is very similar, but it differs in handling the xml setting files. All the operations described from now on are repeated once for each metadata file to be mapped, so the execution flow becomes a big loop.

The four lines of code reported in Figure 6.7 at this point of the execution flow are relevant for the correct creation of the tables that are going to fill the database `gmql_metadata`. They are executed once for each metadata file, and each file can contain information about more than one `ancestry`. As example, lets take the metadata file "GCST007269.gdm.meta", produced as output of the `transformation` phase. The `Cohort` that refers to the study accession "GCST007269" is linked to seven different `Ancestries`. Part of the content of the file is:

broad_ancestral_category_1 European
broad_ancestral_category_2 Asian unspecified
...
country_of_origin_1 NR

country_of_origin_2 NR
...
country_of_recruitment_1 NR
country_of_recruitment_2 U.S.
...

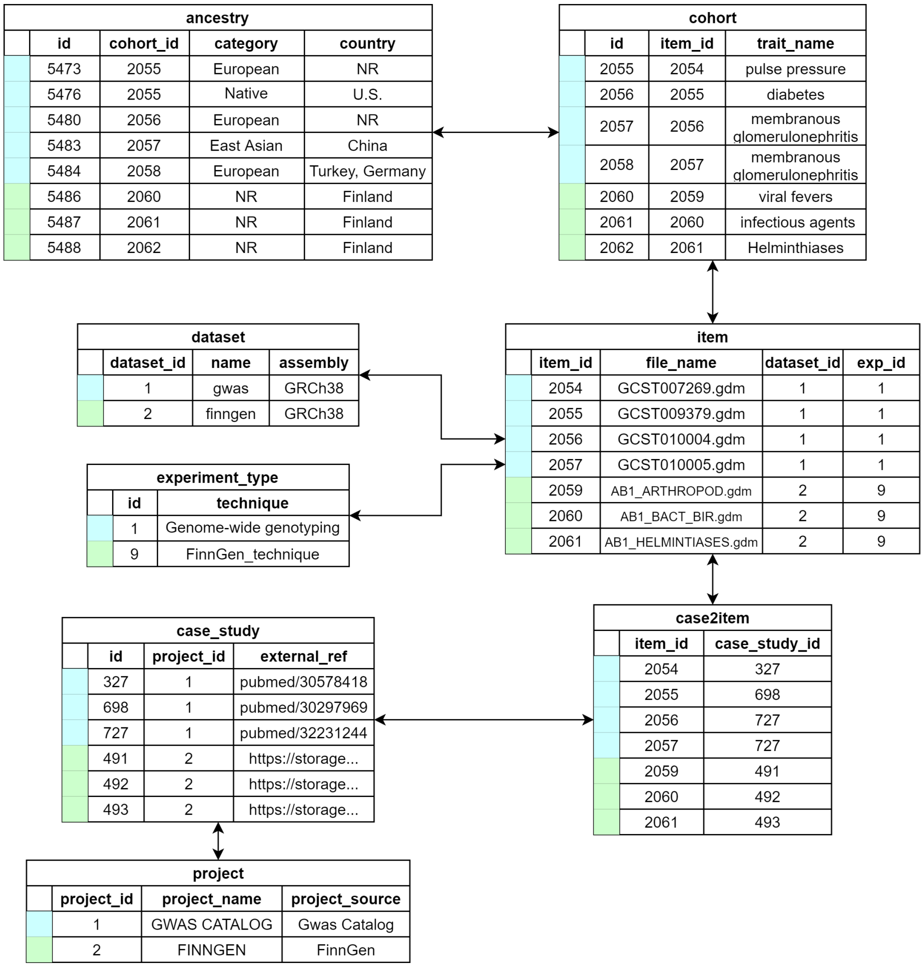
The *ancestryList* contains the information of all the seven different ancestries related to the item with study accession “GCST007269”. The *gwasTableId* keeps trace about the number of ancestries. The *tables* are then created with the proper cardinality, so that seven different ancestry tables are created.

The reader can see an illustrative example of the output of the `mapper` phase in Figure 6.8. The first two entries of the `ancestry` table are taken from the seven different ancestries contained in the file “GCST007269.gdm.meta”. At this point of the execution flow, each table corresponds to an entry of the database `gmql_metadata`. In the reported examples, seven different ancestry tables are created.

The method *createMapper()* of the class *MapperStep.scala* transforms the lines of the input metadata files into a structure useful for the following part of the flow.

At this point are read the files *settingsGWAS.xml* and *settingsFinnGen.xml*. They are read respectively by the methods *xmlReaderGwas()* and *xmlReaderFinnGen()*, based on the current xml file to be read. The role of these files is the one explained in Chapter 5. They contain the information on how to map the metadata into the `gmql_metadata` database. Each row of the xml setting file is called “operation”. An example of the file *xmlReaderGwas()* is:

```
<table name=”ANCESTRIES”>
    <mapping>
        <source_key>broad_ancestral_category_X</source_key>
        <global_key>broadAncestralCategory</global_key>
    </mapping>
    <mapping>
        <source_key>country_of_origin_X</source_key>
        <global_key>countryOfOrigin</global_key>
    </mapping>
    <mapping>
        <source_key>country_of_recruitment_X</source_key>
        <global_key>countryOfRecruitment</global_key>
    </mapping>
    <mapping>
        <source_key>number_of_individuals_X</source_key>
        <global_key>numberOfIndividuals</global_key>
    </mapping>
    <mapping>
        <source_key>study_accession</source_key>
        <global_key>sourceId</global_key>
    </mapping>
```



ancestry

	id	cohort_id	category	country
🟦	5473	2055	European	NR
🟦	5476	2055	Native	U.S.
🟦	5480	2056	European	NR
🟦	5483	2057	East Asian	China
🟦	5484	2058	European	Turkey, Germany
🟩	5486	2060	NR	Finland
🟩	5487	2061	NR	Finland
🟩	5488	2062	NR	Finland

cohort

	id	item_id	trait_name
🟦	2055	2054	pulse pressure
🟦	2056	2055	diabetes
🟦	2057	2056	membranous glomerulonephritis
🟦	2058	2057	membranous glomerulonephritis
🟩	2060	2059	viral fevers
🟩	2061	2060	infectious agents
🟩	2062	2061	Helminthiases

dataset

	dataset_id	name	assembly
🟦	1	gwas	GRCh38
🟩	2	finngen	GRCh38

item

	item_id	file_name	dataset_id	exp_id
🟦	2054	GCST007269.gdm	1	1
🟦	2055	GCST009379.gdm	1	1
🟦	2056	GCST010004.gdm	1	1
🟦	2057	GCST010005.gdm	1	1
🟩	2059	AB1_ARTHROPOD.gdm	2	9
🟩	2060	AB1_BACT_BIR.gdm	2	9
🟩	2061	AB1_HELMINTIASES.gdm	2	9

experiment_type

	id	technique
🟦	1	Genome-wide genotyping
🟩	9	FinnGen_technique

case_study

	id	project_id	external_ref
🟦	327	1	pubmed/30578418
🟦	698	1	pubmed/30297969
🟦	727	1	pubmed/32231244
🟩	491	2	https://storage...
🟩	492	2	https://storage...
🟩	493	2	https://storage...

case2item

	item_id	case_study_id
🟦	2054	327
🟦	2055	698
🟦	2056	727
🟦	2057	727
🟩	2059	491
🟩	2060	492
🟩	2061	493

project

	project_id	project_name	project_source
🟦	1	GWAS CATALOG	Gwas Catalog
🟩	2	FINNGEN	FinnGen

Figure 6.8: In this figure is shown the partial content of the database `gmql_metadata`, filled with the metadata from five GWAS Catalog files (highlighted in light blue) and from 3 FinnGen files (highlighted in light green). In this example, the reader can find the files reported as examples in section 6.1 and 6.2 when the `download` and `transformation` phases are described. Note that each arrow corresponds to a 1:1 relation since the many-to-many relations are translated by adding proper support tables (see *case2item* table).

</table>

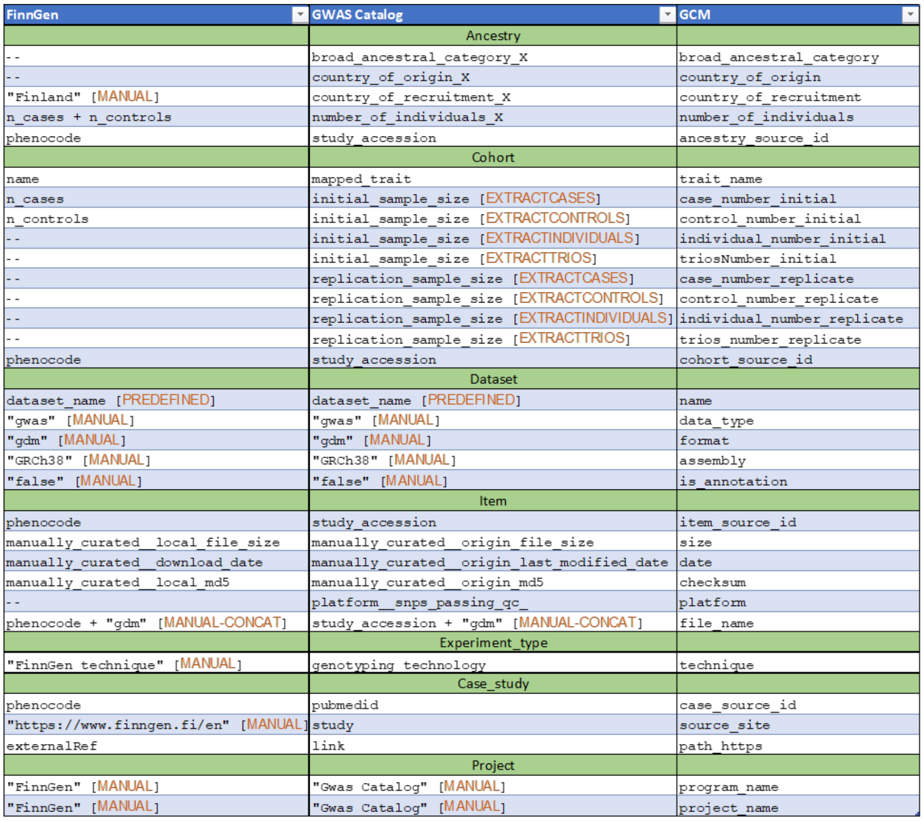
Based on these setting files, in Figure 6.9 are shown how the source-specific metadata are mapped to the GCM attributes. The database is filled with a loop over all the "operations". To refer to the reported xml file, one "operation" is to fill the attribute "broadAncestralCategory" of one *ancestry* table, with the value "European". Another operation is to fill the same attribute of another *ancestry* table with the value *Asian unspecified*.

The loop starts with the method *populateTable()* of the class *MapperStep.scala* which, given the current "operation", selects the proper table to be filled and the corresponding value to be inserted. The method *selectInsertMethod()* of the class *InsertMethod.scala* provides some tool to properly fill the tables. In fact there are some metadata values that have to be elaborated or manually added. The method is specified in the files *settingsGWAS.xml* and *settingsFinnGen.xml*.

For clarity purpose, let's consider the pair <initial_sample_size><450 Japanese ancestry cases, 5,774 Japanese ancestry controls>. This metadata are going to fill the attributes *caseNumber_initial* and *controlNumber_initial*, by specifying the methods "EXTRACTCASES" and "EXTRACTCONTROLS" respectively. Another possible method is "MANUAL" and it is used when the metadata value is added manually. An example of this latter case is the value "Finland" which is inserted in the attribute "countryOfRecruitment" of all the ancestries of the FinnGen dataset.

The last step of the loop consists of inserting the row in the current table, with the method *setParameter()*. When all the "operations" are performed and all the tables of the `gmql_metadata` database are properly filled with the actual values of the current metadata file, the flow goes on with the following metadata file.

After all the metadata files for the current genomic source have been mapped, the flow reaches the end by calculating some statistics about the current execution of the program.



FinnGen	GWAS Catalog	GCM
	Ancestry	
--	broad_ancestral_category_X	broad_ancestral_category
--	country_of_origin_X	country_of_origin
"Finland" [MANUAL]	country_of_recruitment_X	country_of_recruitment
n_cases + n_controls	number_of_individuals_X	number_of_individuals
phenocode	study_accession	ancestry_source_id
	Cohort	
name	mapped_trait	trait_name
n_cases	initial_sample_size [EXTRACTCASES]	case_number_initial
n_controls	initial_sample_size [EXTRACTCONTROLS]	control_number_initial
--	initial_sample_size [EXTRACTINDIVIDUALS]	individual_number_initial
--	initial_sample_size [EXTRACTTRIOS]	triosNumber_initial
--	replication_sample_size [EXTRACTCASES]	case_number_replicate
--	replication_sample_size [EXTRACTCONTROLS]	control_number_replicate
--	replication_sample_size [EXTRACTINDIVIDUALS]	individual_number_replicate
--	replication_sample_size [EXTRACTTRIOS]	trios_number_replicate
phenocode	study_accession	cohort_source_id
	Dataset	
dataset_name [PREDEFINED]	dataset_name [PREDEFINED]	name
"gwas" [MANUAL]	"gwas" [MANUAL]	data_type
"gdm" [MANUAL]	"gdm" [MANUAL]	format
"GRCh38" [MANUAL]	"GRCh38" [MANUAL]	assembly
"false" [MANUAL]	"false" [MANUAL]	is_annotation
	Item	
phenocode	study_accession	item_source_id
manually_curated__local_file_size	manually_curated__origin_file_size	size
manually_curated__download_date	manually_curated__origin_last_modified_date	date
manually_curated__local_md5	manually_curated__origin_md5	checksum
--	platform__snps_passing_qc_	platform
phenocode + "gdm" [MANUAL-CONCAT]	study_accession + "gdm" [MANUAL-CONCAT]	file_name
	Experiment_type	
"FinnGen technique" [MANUAL]	genotyping technology	technique
	Case_study	
phenocode	pubmedid	case_source_id
"https://www.finngen.fi/en" [MANUAL]	study	source_site
externalRef	link	path_https
	Project	
"FinnGen" [MANUAL]	"Gwas Catalog" [MANUAL]	program_name
"FinnGen" [MANUAL]	"Gwas Catalog" [MANUAL]	project_name

Figure 6.9: The attributes in the right column are from the Genomic Conceptual Model that are about to be filled with metadata values from the FinnGen and GWAS Catalog. In the left column appear the metadata attributes from FinnGen after that the `transformation` phase has been performed. In the central column there are the metadata attributes of the GWAS Catalog, they too are taken from the metadata files that are the output of the `transformation` stage. In this table is shown how the source-specific attributes are mapped to the GCM ones. In square brackets is specified the "method" used to map the attributes. In particular `MANUAL` means that the GCM attribute is filled with the value specified, so the name does not refer to an attribute but it is treated as a value. `MANUAL-CONCAT` adds the specified value over an attribute's value. `PREDEFINED` uses as value an internal attribute and not a metadata. The methods `EXTRACTCASES`, `EXTRACTCONTROLS`, `EXTRACTTRIOS` and `EXTRACTINDIVIDUALS` are developed to derive the corresponding values from a verbose metadata, which is the description of the sample of the initial stage or of the replication one.

6.4 Time and space requirements

The two datasets are periodically updated into the corresponding source repositories. A new version of the GWAS Catalog is made publicly available monthly, while new versions of FinnGen dataset are public released every six months. The FinnGen project will be concluded by 2023 and its last release will be available at the beginning of 2025. Each new release increases in size, since new GWAS studies are added to the repositories. At the moment of writing (June 2021), the most recent available version of GWAS Catalog has been made available on May 6$^{\text{th}}$ 2021 and it contains 16854 different GWAS studies, while the most recent version of FinnGen is the 5$^{\text{th}}$ release public available from May 2021 and it contains 2804 different phenotypes.

In Table 6.1 are reported the execution times and the space requirements of the three steps of the integration pipeline (`downloader`, `transformer` and `mapper`). The space requirements reported in that table are referred to the disk space of the downloaded and transformed files; it does not take into account the space used over the two databases gmql_importer and gmql_metadata.

Table 6.1: In this table are reported the execution times of the three integration steps of the pipeline and their corresponding space requirements. The symbol $^{(*)}$ means that the time or space is estimated and not observed.

step	time (hh:mm:ss)	space
Catalog - Downloader	00:00:10	169M
Catalog - Transformer	03:32:54	111M
Catalog - Mapper	02:13:08	–
FinnGen Downloader	23:10:00$^{(*)}$	1542G$^{(*)}$
FinnGen - Transformer	402:26:00$^{(*)}$	4065G$^{(*)}$
FinnGen - Mapper	01:27:00$^{(*)}$	–

Chapter 7

GMQL queries

Thanks to Genome-wide association studies we know the correlations between many phenotypes and their corresponding mutations of DNA. The exact interpretation of that SNPs is not trivial at all for at least two reasons. First, the output of GWASs are often large clusters of SNPs in linkage disequilibrium, making it difficult to distinguish causal SNPs from neutral variants in linkage. Second, even assuming the causal variants can be identified, interpretation is limited by incomplete knowledge of non-coding regulatory elements, their mechanisms of action and the cellular states and processes in which they function. Indeed, it's more difficult to understand the relation between a SNP in a non-coding region and its associated phenotype.

For the aforementioned reasons, it's important to further investigate GWAS data by merging different genomic datasets or by performing some analysis. For this purpose is used the GenoMetric Query Language already introduced in section 3.3. The GMQL allows to conduce multi-omic studies by creating queries over different omic sources. It aims at improving the knowledge in the genomic field of biology and making progress in disease treatments and prevention.

In next sections are presented some GMQL queries useful to extract interesting information from the mapped GWAS sources. The reported queries are representative of the most biologically interesting cases.

The studies from GWAS Catalog have been pre-processed before being uploaded on GMQL server. The process is very simple and allows to execute more powerful queries. Each GWAS study can have more than one ancestry and after the `transformation` phase of the META-BASE pipeline, the different ancestries are referred with an ordinal number. The pre-process concerns the metadata attributes:

- broad_ancestral_category_[0_9]
- country_of_origin_[0_9]
- country_of_recruitment_[0_9]

if all the ancestries (identified with progressive numbers) of a study have all the same value for one of the three listed attributes, then a new attribute is created containing the corresponding value. Let's suppose that the study GCST1234 have the two metadata country_of_origin_1 and country_of_origin_2 having the same value "Italy", then the new metadata "country_of_origin" is created containing the value "Italy". In this way, in some queries could be useful to select all the studies whose cohorts are recruited in Italy, both if they are cases or controls.

For the purpose of the proposed examples, in some of them we want to restrict GWAS samples based on the values of the newly computed metadata.

7.1 Queries upon GWAS Catalog studies

7.1.1 Common SNPs between African and European cohorts

The following query is built upon genomic data coming from GWAS Catalog. It exploits the GenoMetric Query Language to compare different cohorts of people.

Given the studies from GWAS Catalog based on "European" and "African" cohorts, map each SNP from the former set to overlapping SNPs of the latter one, joining only studies mapped to the same trait.

```
1  /* load the studies from GWAS Catalog whose cohorts are
2     "European" */
3  EU = SELECT(broad_ancestral_category == "European") GWAS;
4
5  /* load the studies from GWAS Catalog whose cohorts are
6     "African American or Afro-Caribbean" */
7  AF = SELECT(broad_ancestral_category == "African␣American"
8        "or␣Afro-Caribbean") GWAS;
9
10 /* find the SNPs which are in common for each trait*/
11 RES = MAP(joinby: mapped_trait) EU AF;
12 FIL = SELECT(region: count_EU_AF > 0) RES;
13 MATERIALIZE FIL into COMMON_SNPS ;
```

This query compares, for each trait, the SNPs found in studies based on "European" cohorts against "African" cohorts. The first two `SELECT` operators in line 3 and 7 load the studies from GWAS Catalog based on the cohorts under consideration.

The core operation of this query is the `MAP` operator in line 11. For each region in EU dataset counts the overlapping SNPs from the AF dataset. Are mapped only SNPs from samples mapped to the same trait, as specified by the *joinby* condition.

In Table 7.1 is reported a small portion of the output of this query. The three listed SNPs, each one mapped to a specified trait, are found both on "European" and "African" cohorts. Since the `MAP` has the condition *joinby: mapped_trait*, each entry is taken from a different file of the output (identified by the column "sample ID" in the proposed table).

statistics:

- *Execution time: 00:40:09*
- *Number of regions: 90*
- *Number of samples: 64*
- *Size: 0.24 MB*

Table 7.1: It contains a small portion of the output of the proposed GMQL query in section 7.1.1.

chr	left	right	mapped_trait	snps	p-value	count	sample ID
chr10	112998590	112998591	Type 2 diabetes	rs7903146	0	1	S_00000
chr19	44878777	44878778	Alzheimer's disease	rs6859	0	1	S_00012
chr2	233759924	233759925	Bilirubin levels	rs887829	0	1	S_00026

7.1.2 Frequent mutations for each trait

This query exploits the GenoMetric Query Language to compute some statistics about the content of GWAS Catalog.

Given all the studies from GWAS Catalog, count how many times every SNP has been found in the whole Catalog, aggregating them over traits.

```
/* load the GWAS dataset */
GWAS = SELECT() GWAS;

```

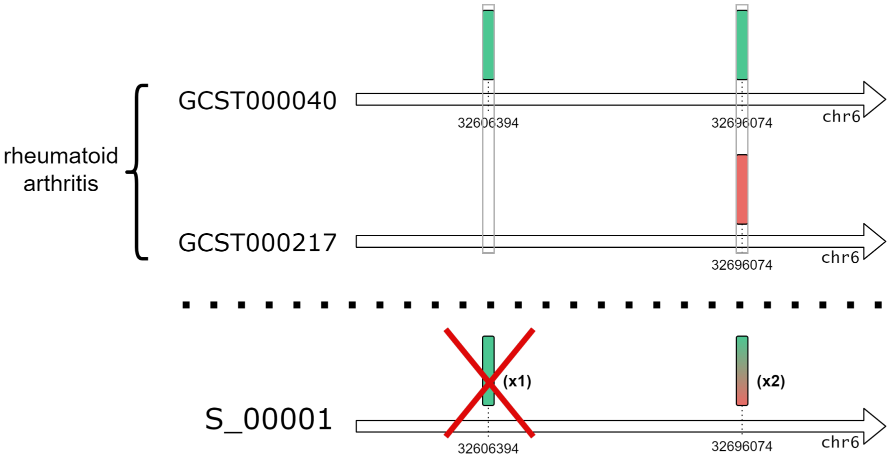


Figure 7.1: Schematic representation of the query in section 7.1.2. For each trait are kept only those SNPs which have been identified at least two times in the whole GWAS Catalog.

```
 4  /* merge all the studies that share the "mapped_trait" into
 5     a single sample */
 6  MER = MERGE(groupby: mapped_trait) GWAS;
 7
 8  /* group overlapping SNPs into a single region */
 9  GRO = GROUP(region_aggregates: reg_num AS COUNT()) MER;
10  FIL = SELECT(region: reg_num > 1) GRO;
11  ORD = ORDER(region_order: reg_num ASC) FIL;
12  MATERIALIZE ORD into FREQUENT_SNPS;
```

This query is performed on a single dataset and starts by loading the whole GWAS Catalog.

The `MERGE` operation in line 6 groups all the samples mapped over the same trait into a single one. This step is the basis for the next operation in line 9.

The `GROUP` operator, for each sample, considers the SNPs which share the same coordinates as single regions. For each computed region is introduced the attribute "reg_num" which counts how many overlapping SNPs are grouped into a single region. The operation in line 10 filters out the regions which are made of less than two overlapping SNPs. Finally the resulting regions are ordered, for each single trait, according to their frequency.

In Table 7.2 is reported a small fragment of the output of this query. Are reported a few SNPs mapped to three different traits, which have been found in more than 2 studies. Since the `MERGE` operator has the condition *groupby:mapped_trait*, the reported entries of the following table are taken from three different files of the output (identified with the attribute "sample ID").

statistics:

- *Execution time: 00:06:00*
- *Number of regions: 30017*
- *Number of samples: 893*
- *Size: 5.98 MB*

Table 7.2: It contains a small portion of the output of the GMQL query proposed in section 7.1.2. Each sample of the output refers to a different trait. In the table are reported a couple of SNPs for three samples taken from the output.

mapped_trait	chr	left	right	count	sample ID
HIV-1 infection	chr6	31423624	31423625	2	S_00012
		. . .			
	chr6	31464003	31464004	3	
adolescent idiopathic scoliosis	chr2	221895559	221895560	2	S_00091
		. . .			
	chr10	101219450	101219451	4	
amyotrophic lateral sclerosis	chr14	30678292	30678293	2	S_00842
		. . .			
	chr19	17641880	17641881	5	

7.1.3 Counting distinct DNA mutations in ancestral groups

This query exploits the GenoMetric Query Language to aggregate different GWAS studies and to compute some statistics about their contents.

Given the whole GWAS Catalog, aggregate the studies based on the attributes "mapped_trait" and "broad_ancestral_category"; for each region of the defined samples, count how many SNPs overlap and count how many distinct regions are contained in each defined sample.

```
/* load the GWAS dataset */
MUTATION = SELECT(broad_ancestral_category == '*') GWAS;

/* consider regions defined by at least one SNP, grouped
```

```
5     by "mapped_trait" and "ancestral_category" */
6  MUTATION_ANCE = COVER(1, ANY; groupby: mapped_trait,
7                  broad_ancestral_category; aggregate:
8                  overlap_count AS COUNT()) MUTATION;
9
10 /* count the number of SNPs and extend it as metadata */
11 MUTATION_COUNT = EXTEND(mutation_count AS COUNT())
12                  MUTATION_ANCE;
13 MATERIALIZE MUTATION_COUNT INTO MUTATION_COUNT;
```

This query is performed on a single dataset. It starts with a `SELECT` operation which selects all the studies from GWAS Catalog which have a non-empty value for metadata "broad_ancestral_category".

Line 6 contains the `COVER` operation which considers all areas defined by a minimum of one region to any number of overlapping SNPs. The *groupby* option allows this operation to be performed only on studies which share the values for the specified metadata. For each resulting region is computed the attribute "overlap_count", which counts how many overlapping SNPs compose the current region.

Finally, line 11 counts how many regions belong to each defined samples and store the computed number as the metadata "mutation_count".

In Table 7.3 are reported a couple of entries for three samples from the output of this query. For each ancestral category and each trait, are counted how many SNPs have been found.

statistics:

- *Execution time: 00:07:19*
- *Number of regions: 151647*
- *Number of samples: 3386*
- *Size: 13.51 MB*

Table 7.3: It contains a small portion of the output of the proposed GMQL query in section 7.1.3. The attribute "ove_count" indicates how many overlapping SNPs identify the current region; the attribute "mut_count" identifies how many regions are contained into the current sample.

mapped_trait	ancestral_category	chr	left	right	ove_count	mut_count	sample ID
systolic blood pressure	European	chr19	116226419	116226420	1	1026	S_00235
			. . .				
		chr17	46935905	46935906	3		
mathematical ability	European	chr8	140982679	140982680	2	2124	S_00209
			. . .				
		chr13	88581860	88581861	4		
prostate carcinoma	East Asian	chr11	7526356	7526357	1	117	S_00237
			. . .				
		chr10	46046326	46046327	5		

7.2 Queries upon multiple datasets: GWAS Catalog, FinnGen, TCGA, GENCODE, 1000 Genomes Project and Encode

7.2.1 Cancer mutations from TCGA and GWA studies for "breast carcinoma"

The Cancer Genome Atlas Program includes multiple genomic datasets all related to 37 different types of cancer [14]. TCGA includes gene expression profiling, copy number variation profiling, SNP genotyping, genome wide DNA methylation profiling, microRNA profiling and exon sequencing.

GWAS Catalog dataset includes 124 different types of cancer, mapped over the EFO ontology. The mapping between SNPs from GWA studies and the TCGA profiles of gene expression for a given type of cancer can result in better understanding the risk factors for cancer.

The following query maps particularly expressed genes from TCGA dataset with SNPs from GWAS, both associated to "breast cancer". The query focuses on the genes BRCA1 and BRCA2, since germ-line mutations in those genes are the main part of genetic and hereditary factors for breast cancer [27].

Given the mutational data from the TCGA dataset referred to "breast cancer" *and restricted to genes BRCA1 and BRCA2, filter only regions with a high level of expression and find the ones having at least one overlapping SNP taken from GWAS studies mapped to the same trait.*

```
1  /* load "breast cancer" mutations from GWAS Catalog */
2  GWAS = SELECT(mapped_trait == "breast␣carcinoma") GWAS;
3
4  /* load genes "BRCA1" and "BRCA2" from TCGA */
5  TCGA = SELECT(biospecimen__admin__disease_code == "BRCA";
6         region: gene_symbol == "BRCA1" or gene_symbol ==
7         "BRCA2") GRCh38_TCGA_gene_expression;
8
9  /* merge all TCGA samples into a single one */
10 MER = MERGE() TCGA;
11
12 /* filter TCGA regions which are particularly expressed */
13 EXT = EXTEND(quart3 AS q3(fpkm)) MER;
14 GENE_EXP = SELECT(region: fpkm > META(quart3)) EXT;
15
16 /* MAP TCGA regions to overlapping GWAS SNPs */
17 RES = MAP() GENE_EXP GWAS;
18 FIL = SELECT(region: count_GENE_EXP_GWAS > 0) RES;
19 MATERIALIZE FIL into CANCER;
```

This query starts as usual with a `SELECT` operator in line 2 which loads the studies from GWAS Catalog mapped to trait “breast carcinoma”.

Line 5 contains another `SELECT` operator which loads the data referred to genes “BRCA1” and “BRCA2” from the dataset TCGA_gene_expression.

The operator `MERGE` in line 10 creates a single sample grouping all the regions from the samples selected from the `SELECT` operation in line 5. This step prepares the dataset for the next operation in line 13.

The `EXTEND` operation in line 13 computes the third quartile of all the values of the region attribute “fpkm”, which indicates the level of expression of the corresponding region. The computed quartile is written into the new metadata called “quart3”.

In line 14, by means of a `SELECT` operator, are filtered only those regions from the MER dataset which have the value of “fpkm” greater than the value of the metadata “quart3”, that is its third quartile. This operation keeps only the regions which have a “high” level of expression.

The core operator of the query is in line 17. The `MAP` operator counts, for each region of the dataset EXP, how many SNPs from GWAS dataset overlap.

In Table 7.4 is reported a single entry taken from the output of the query. All the regions in the output are referred to the gene “BRCA2”; no overlapping SNPs are found for gene “BRCA1”.

statistics:

- *Execution time: 00:06:56*
- *Number of regions: 440*
- *Number of samples: 5*
- *Size: 17.24 MB*

Table 7.4: It contains a small portion of the output of the proposed GMQL query in section 7.2.1.

chr	left	right	gene	fpkm	count_snps	quart3
chr13	32315473	32400266	BRCA2	347792385	1	283041330,3

7.2.2 SNPs occurring in untranslated regions

The GENCODE project was founded in 2003 as part of the pilot phase of the ENCODE project. Today the GENCODE consortium is a long-running partnership of manual annotation and it is the reference annotation of choice adopted by a lot of large international consortia including ENCODE, TCGA and many others [10]. The consortium annotates protein-coding genes, pseudogenes, long non-coding RNAs (lncRNAs) and small non-coding RNAs.

Between the available annotations there are the UTRs, which stand for "untranslated regions". Mutations occurring in those regions are difficult to interpret and to associate with their consequences. This is the reason why coupling SNPs from GWAS dataset with annotations from GENCODE consortium is potentially very powerful.

Genetic variants in the coding sequence of a gene (exons), because of their easier interpretation, have often been given priority, although it has long been clear that coding sequence variants per se were insufficient for mapping complex diseases. However, variants in the intervening sequences (introns) or in the untranslated regions (UTRs), although not changing the predicted protein sequence, may be pivotal in the regulation of gene expression [26].

The UTRs are the mRNA sequences flanking the beginning and end of the coding sequences; as their name suggests, UTRs are part of the mRNA but are not translated into proteins; the role of UTR sequences is briefly described in Figure 7.2.

Following is proposed a bunch of queries that exploit UTR regions from GENCODE dataset.

Given all the SNPs mapped to trait "primary biliary cirrhosis" from GWAS

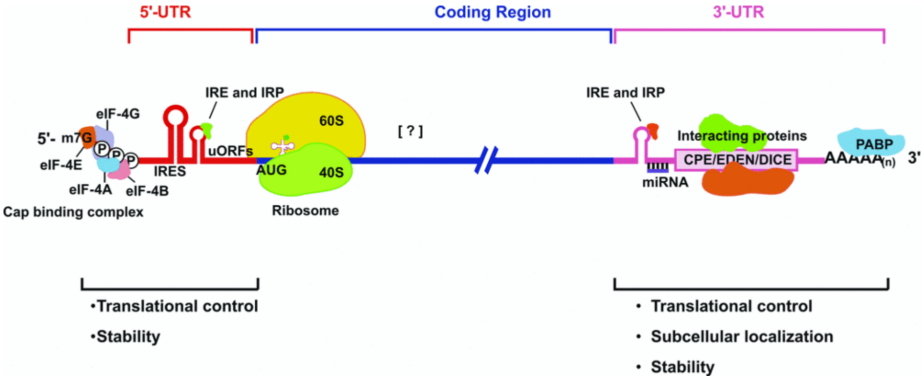


Figure 7.2: Gene expression is regulated at the RNA level by virtue of the presence of 5' and 3'UTR regulatory elements such as upstream open reading frames (uORFs), internal ribosome entry sites (IRESs), as well as the UTR's secondary structure, sequence composition and length. The majority of regulatory elements are recognized by RBPs or by non-coding RNAs (ncRNAs) such as miRNAs. Overall, these mechanisms modulate the mRNA stability, localization and translation.

Catalog, filter out only those occurring in UTR regions:

```
1  /* load SNPs associated to "primary biliary cirrhosis"
2     from GWAS Catalog */
3  CIR = SELECT(mapped_trait == "primary␣biliary␣cirrhosis")
4         GWAS;
5
6  /* load untranslated regions from GENCODE dataset */
7  UTR = SELECT(annotation_type == "UTR" AND release_version
8         == "27") GRCh38_ANNOTATION_GENCODE;
9
10 /* find SNPs which have at least an overlapping
11    untranslated region */
12 MUT = MAP() CIR UTR;
13 MUT_fil = SELECT(region: count_CIR_UTR >= 1) MUT;
14 MATERIALIZE MUT_fil INTO UTR;
```

This query begins with a `SELECT` operation at line 3, loading from GWAS Catalog the studies mapped to trait "primary biliary cirrhosis".

At line 7 another `SELECT` operation loads UTR regions from the latest release of the GENCODE dataset.

The `MAP` and `SELECT` operations at line 12 and 13 find, for each SNP of CIR dataset, the overlapping UTR regions. In the output dataset appear only those

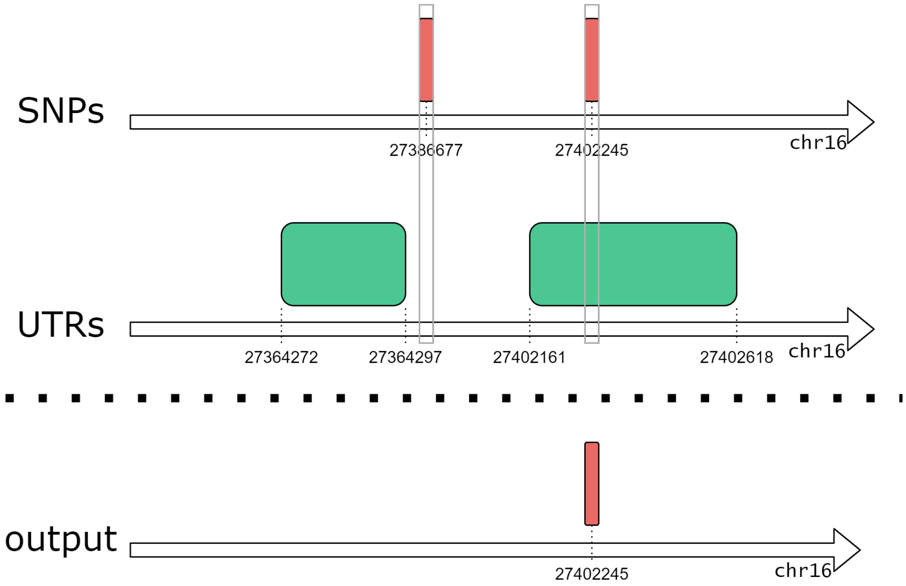


Figure 7.3: Schematic representation of the query in the first part of section 7.2.2. The green regions are the untranslated regions identified from GENCODE dataset while the red ones are the SNPs taken from GWAS Catalog. For each SNP are identified the overlapping UTR regions; if at least one UTR region overlaps the current SNP, it is kept as output of the query.

SNPs which occur in UTR regions.

In Table 7.5 is shown a small fragment of the output of this query. The first SNP of the table is the *rs2189521* occurring in gene *IL21R*. Qiu and colleagues [26] reported that the risk allele for primary biliary cirrhosis regulates differential IL21R expression; this variant is also highly correlated with multiple SNPs in the IL21R region, suggesting that variation in IL21R expression may explain this signal. By applying several histochemical experiments, they showed that the enhanced expression in PBC livers (in the hepatic portal tracks) of IL21R and of its ligand, IL21, support an involvement of IL21 signalling pathway deregulation in the disease mechanism.

statistics:

- *Execution time: 00:04:21*
- *Number of regions: 6*
- *Number of samples: 3*
- *Size: 0.01 MB*

Table 7.5: It contains a small portion of the output of the proposed GMQL query in the first part of section 7.2.2. The column “count” indicates how many UTR regions overlap the current SNP.

chr	left	right	mapped_gene	snps	context	count
chr16	27402245	27402246	IL21R	rs2189521	5’_UTR	2
chr15	81305928	81305929	IL16, AC103858.1	rs11556218	missense	3
chr3	119431242	119431243	TMEM39A	rs3732421	3’_UTR	3

Map each UTR region from Gencode dataset with overlapping SNPs from GWAS Catalog; compare the results selecting SNPs mapped to different traits:

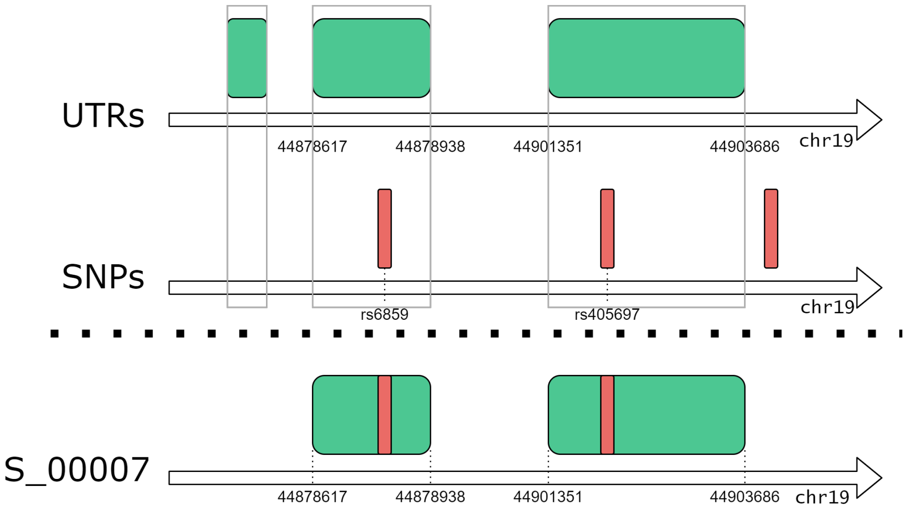


Figure 7.4: Schematic representation of the query in the second part of section 7.2.2. The green regions are the untranslated regions from GENCODE dataset while the red ones are the SNPs mapped to trait “Alzheimer’s disease”. The query filters only those UTRs that have at least one overlapping SNP.

```
/* load SNPs associated to "primary biliary cirrhosis"
   from GWAS Catalog */
CIR = SELECT(mapped_trait == "primary␣biliary␣cirrhosis")
      GWAS;

/* load untranslated regions from GENCODE dataset */
UTR = SELECT(annotation_type == "UTR" AND release_version
      == "27") GRCh38_ANNOTATION_GENCODE;

/* MAP UTR regions with overlapping SNPs from CIR dataset */
MUT = MAP(bag AS BAG(SNPS)) UTR CIR;
MUT_fil = SELECT(region: count_CIR_UTR >= 1) MUT;

/* remove all unnecessary region attributes */
PRO = PROJECT(gene_name, bag) MUT_fil;
MATERIALIZE PRO INTO UTR;
```

statistics:

- *Execution time: 00:03:10*
- *Number of regions: 21*
- *Number of samples: 3*
- *Size: 0.01 MB*

```
/* load SNPs associated to "coronary artery disease"
   from GWAS Catalog */
CAD = SELECT(mapped_trait == "coronary␣artery␣disease")
      GWAS;

/* load untranslated regions from GENCODE dataset */
UTR = SELECT(annotation_type == "UTR" AND release_version
      == "27") GRCh38_ANNOTATION_GENCODE;

/* MAP UTR regions with overlapping SNPs from CAD dataset */
MUT = MAP(bag AS BAG(SNPS)) UTR CAD;
MUT_fil = SELECT(region: count_UTR_CAD >= 1) MUT;

/* remove all unnecessary region attributes */
PRO = PROJECT(gene_name, bag) MUT_fil;
MATERIALIZE PRO INTO UTR;
```

statistics:

- *Execution time: 00:07:40*
- *Number of regions: 135*

- *Number of samples: 18*
- *Size: 0.06 MB*

```
/* load SNPs associated to "Alzheimer's disease"
   from GWAS Catalog */
AZD = SELECT(mapped_trait == "Alzheimer's␣disease")
      GWAS;

/* load untranslated regions from GENCODE dataset */
UTR = SELECT(annotation_type == "UTR" AND release_version
      == "27") GRCh38_ANNOTATION_GENCODE;

/* MAP UTR regions with overlapping SNPs from AZD dataset */
MUT = MAP(bag AS BAG(SNPS)) UTR AZD;
MUT_fil = SELECT(region: count_UTR_AZD >= 1) MUT;

/* remove all unnecessary region attributes */
PRO = PROJECT(gene_name, bag) MUT_fil;
MATERIALIZE PRO INTO UTR;
```

statistics:

- *Execution time: 01:20:27*
- *Number of regions: 36*
- *Number of samples: 10*
- *Size: 0.03 MB*

```
/* load SNPs associated to "bipolar disorder" from GWAS
   Catalog */
BPD = SELECT(mapped_trait == "bipolar␣disorder") GWAS;

/* load untranslated regions from GENCODE dataset */
UTR = SELECT(annotation_type == "UTR" AND release_version
      == "27") GRCh38_ANNOTATION_GENCODE;

/* MAP UTR regions with overlapping SNPs from BPD dataset */
```

```
MUT = MAP(bag AS BAG(SNPS)) UTR BPD;
MUT_fil = SELECT(region: count_UTR_BPD >= 1) MUT;

/* remove all unnecessary region attributes */
PRO = PROJECT(gene_name, bag) MUT_fil;
MATERIALIZE PRO INTO UTR;
```

statistics:

- *Execution time: 00:38:25*
- *Number of regions: 30*
- *Number of samples: 9*
- *Size: 0.03 MB*

In Table 7.6 are reported some regions from the results of the four previous queries. Each row of the table contains one region extracted from the query having the corresponding mapped trait. Each row is a UTR region extracted from the Gencode dataset and it contains the overlapping SNP(s) from GWAS studies mapped to the corresponding trait. If for a single UTR region are found more than one SNPs, are added the round brackets with the amount of overlapping SNPs. The last column of the table "# reg" indicates how many different UTR regions result from the query built upon the same trait of the corresponding rows.

Table 7.6: It contains a small portion of the output of the proposed GMQL queries in the second part of section 7.2.2.

trait	chr	start	end	gene	SNP(s)	# reg
primary biliary cirrhosis	19	10352433	10352524	TYK2	rs34536443	21
			. . .			
	17	39765283	39765792	IKZF3	rs907091	
coronary artery disease	7	130023510	130023723	ZC3HC1	rs11556924(x2)	135
			. . .			
	3	138402266	138405534	MRAS	rs9818870	
Alzheimer's disease	19	44888376	44889228	NECTIN2	rs6857(x2)	36
			. . .			
	2	127638425	127639283	LIMS2	rs78022502	
bipolar disorder	17	44123503	44123702	HDAC5	rs112114764	30
			. . .			
	19	19249908	19252233	NCAN	rs1064395	

7.2.3 Match GWAS mutations with variants from 1000 Genomes Project

The 1000 Genomes Project was born in 2008 as an international research effort to establish by far the most detailed catalogue of human genetic variations [12]. Genome-wide association studies can discover new loci that contribute to common human diseases. For each such locus, it is currently necessary to sequence the newly discovered region to define all common and rare variants.

The GWA studies carried on so far explained a modest fraction of all the disease risks; some of this uncaptured risk is due to alleles of lower frequency but larger effect. If such alleles are in genes already localized by GWAS, then targeted sequencing may find them. Similarly, some of the uncaptured risk is due to the effects of structural variants that are not in linkage disequilibrium with common SNPs. Thus, a more complete understanding of the role of genetic variation in disease requires a deeper catalog of genetic variation.

The genomes sequenced in the 1000 Genomes Projects are unselected with regard to phenotype, so to provide a resource of variants to support deeper understanding of newly discovered loci influencing human disease. The projects include SNPs with allele frequencies as low as 1% across the genome and 0.1-0.5% in gene regions, as well as structural variants like CNVs. It includes genomes from 26 different populations, including Finnish.

Following is proposed a collection of queries that exploit the FinnGen dataset, the GWAS Catalog and genome data from 1000 Genomes Project.

For each relevant SNP on chromosome 2 from FinnGen study associated to Schizophrenia, find the closest deletion from 1000 Genomes dataset referred to Finnish people:

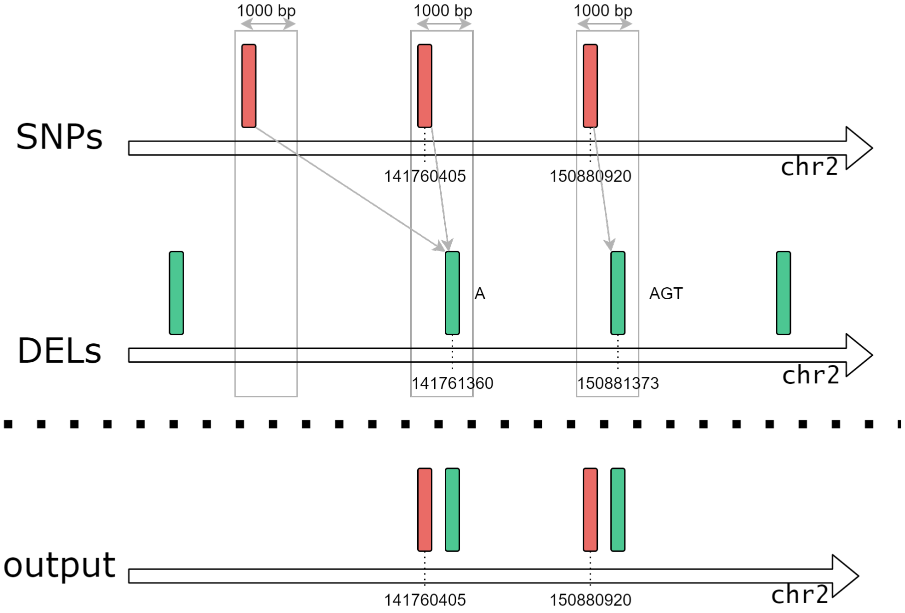


Figure 7.5: Schematic representation of the query in the first part of section 7.2.3. The green regions are the deletions identified from 1000 Genomes Projects while the red ones are the SNPs taken from FinnGen dataset. For each deletion, is considered the closest SNP and it is kept only if it falls within 1000 base pairs from the considered deletion.

```
/* load deletions from 1000 Genomes */
OKG = SELECT(population == "FIN"; region: chr == chr2
      and mut_type == "DEL") GRCh38_1000GENOMES_2020_01;

/* load data from FinnGen, filtered by phenotype and pval */
FIN = SELECT(name == "Schizophrenia"; region: chr == chr2
      and pval < 0.0005) FinnGen;

/* find deletions close to SNPs */
RES = JOIN(MD(1), DLE(1000)) FIN OKG;
PRO = PROJECT(FIN.ref, FIN.alt, FIN.rsids, OKG.ref,
12        OKG.mut_type) RES;
13  MATERIALIZE PRO into DELETIONS;
```

The first operation at line 2 selects the samples from 1000 Genomes dataset which are referred to Finnish population. For those files, filters only those regions on chromosome 2 which are deletions.

Line 6 selects the samples from FinnGen dataset referring to phenotype "Schizophrenia". For the resulting sample (only one) it filters the regions based on chromosome and p-value.

Line 10 uses the JOIN operator to find for each pair of samples, one from FIN dataset (only one sample) and the other one from OKG dataset (211 samples), the closest DELETION from each FinnGen SNP only if its distance is less than 1000 bp from the SNP.

Line 11 exploits the operator PROJECT to remove superfluous region attributes, keeping only the relevant ones.

In Table 7.7 is proposed the partial output resulting from this query, for clarifying purpose.

statistics:

- *Execution time: 00:12:14*
- *Number of regions: 8375*
- *Number of samples: 105*
- *Size: 2.20 MB*

Table 7.7: It contains a small portion of the output of the GMQL query proposed in the first part of section 7.2.3. Each row contains the closest deletion to each SNP only if its distance is less than 1000 bp from it. The attributes "F.ref", "F.alt" and "F.rsids" derive from FinnGen dataset while "O.ref" and "O.m_type" are from 1000 Genomes.

chr	left	right	F.ref	F.alt	F.rsids	O.ref	O.m_type
chr2	150880920	150881373	G	A	rs149379995	AGT	DEL
chr2	61881936	61882711	C	A	rs542459233	TTT	DEL
chr2	193889624	193889867	A	G	rs182720836	CTC	DEL

Join each variation from 1000 Genomes dataset referred to Finnish people with overlapping relevant SNPs on chromosome 2 from FinnGen study associated to Schizophrenia:

```
1  /* load data from 1000 Genomes and FinnGen */
2  OKG = SELECT(population == "FIN"; region: chr == chr2)
3        GRCh38_1000GENOMES_2020_01;
4  FIN = SELECT(name == "Schizophrenia"; region: chr == chr2
5        AND pval < 0.0005) FinnGen;
6
7  /* join each variations from 1000 Genomes with overlapping
8     SNPs associated to "Schizophrenia" */
9  RES = JOIN(distance<1; output: BOTH) OKG FIN;
10
11 /* filter only the resulting variations which are SNPs */
12 SEL = SELECT(region: OKG.mut_type == "SNP") RES;
13 MATERIALIZE SEL into VARIANTS;
```

This query starts with two `SELECT` operations: the first one at line 2 loads the DNA variations from 1000 Genomes dataset on chromosome 2 about Finns while the second one at line 4 loads relevant SNPs (pval <0.0005) from FinnGen study associated to "Schizophrenia".

Line 9 contains the `JOIN` operator which finds, for each variation from 1000 Genomes dataset, the overlapping SNPs from FIN dataset.

Line 12 exploits the `SELECT` operator to filter out the previously found variations, keeping only the ones which are SNPs.

statistics:

- *Execution time: 00:33:53*
- *Number of regions: 5822*
- *Number of samples: 105*
- *Size: 2.94 MB*

After the query has been executed, a post-process step is performed to keep only that variations which correspond exactly to their joined SNPs. This last job compares the alternative alleles from 1000 Genomes and FinnGen SNPs, excluding those regions in which they don't coincide. At the end of this filtering process, remain 5,589 out of 5,822 identified SNPs.

Join each variation from 1000 Genomes dataset referred to Japanese people with overlapping SNPs from GWAS Catalog studies associated to Schizophrenia and referred to the same population:

```
/* load data from 1000 Genomes and GWAS Catalog */
JPT = SELECT(population == "JPT"; region: chr == chr2)
```

```
      GRCh38_1000GENOMES_2020_01;
JAP = SELECT(mapped_trait == "schizophrenia" AND
      country_of_recruitment == "Japan") GWAS;

/* join each variations from 1000 Genomes with overlapping
   SNPs associated to "Schizophrenia" */
RES = JOIN(distance<1; output: BOTH) JPT JAP;

/* filter only the resulting variations which are SNPs */
SEL = SELECT(region: JPT.mut_type == "SNP") RES;
MATERIALIZE SEL into VARIANTS;
```

This query is very similar to the previous one, so for details the reader is invited to read the explanation in the previous lines. The main difference is that are selected the SNPs from Japanese cohorts instead of Finnish ones, and they are selected from GWAS Catalog instead of FinnGen dataset.

statistics:

- *Execution time: 04:51:55*
- *Number of regions: 1036*
- *Number of samples: 178*
- *Size: 3.53 MB*

Also for the outcome of this query is performed a filtering step to filter out only those variations corresponding exactly to the joined SNPs. After this filtering process, remain 396 out of 1,036 identified SNPs.

Join each variation from 1000 Genomes dataset referred to Chinese people with overlapping SNPs from GWAS Catalog studies associated to Schizophrenia and referred to the same population:

```
/* load data from 1000 Genomes and GWAS Catalog */
CHB = SELECT(population == "CHB" OR population == "CHS" OR
      population == "CDX") GRCh38_1000GENOMES_2020_01;
CHINA = SELECT(mapped_trait == "schizophrenia" AND
        country_of_recruitment == "China") GWAS;

/* join each variations from 1000 Genomes with overlapping
   SNPs associated to "Schizophrenia" */
RES = JOIN(distance<1; output: BOTH) CHB CHINA;

/* filter only the resulting variations which are SNPs */
```

```
SEL = SELECT(region: CHB.mut_type == "SNP") RES;
MATERIALIZE SEL into VARIANTS;
```

This query is very similar to the two previous ones, so for details the reader is invited to read the explanation in the previous lines. In this query are selected SNPs from Chinese people to compare them with the ones from Finnish and Japanese cohorts.

statistics:

- *Execution time: 14:00:17*
- *Number of regions: 1016*
- *Number of samples: 1450*
- *Size: 18.77 MB*

Also for the outcome of this query is performed a filtering step to filter out only those variations corresponding exactly to the joined SNPs. Unfortunately, all the SNPs referred to Chinese people from GWAS Catalog lack of the information about the alternative allele. For this reason, after this filtering process, remain 0 out of 1,450 identified SNPs.

Join each variation from 1000 Genomes dataset referred to people recruited in U.K. with overlapping SNPs from GWAS Catalog studies associated to Schizophrenia and referred to the same population:

```
/* load data from 1000 Genomes and GWAS Catalog */
GBR = SELECT(population == "GBR" OR population == "ITU" OR
      population == "STU") GRCh38_1000GENOMES_2020_01;
UK = SELECT(mapped_trait == "treatment␣refractory"
     "schizophrenia,␣response␣to␣clozapine" AND
     country_of_recruitment == "U.K.") GWAS;

/* join each variations from 1000 Genomes with overlapping
   SNPs associated to "Schizophrenia" */
RES = JOIN(distance<1; output: BOTH) GBR UK;

/* filter only the resulting variations which are SNPs */
SEL = SELECT(region: GBR.mut_type == "SNP") RES;
MATERIALIZE SEL into VARIANTS;
```

This query is analogous to the preceding ones, so for details the reader is invited to read the explanation in the previous lines. In this query are selected SNPs from people recruited in U.K. mapped to schizophrenia (mapped_trait = “treatment refractory schizophrenia, response to clozapine” since there are no studies from GWAS Catalog with cohorts from U.K. and mapped to "schizophrenia") to compare them with the cohorts of the previous queries.

statistics:

- *Execution time: 08:26:02*
- *Number of regions: 685*
- *Number of samples: 473*
- *Size: 8.72 MB*

Also for the outcome of this query is performed a filtering step to filter out only those variations corresponding exactly to the joined SNPs. After this filtering process, remain 423 out of 685 identified SNPs.

In Table 7.8 are compared the results of the four preceding queries. For the four nations under consideration, is reported how many variations from 1000 Genomes Project are found, how many SNPs from FinnGen or GWAS Catalog are found and the count of the resulting regions for the corresponding queries (column "# correspondences").

Table 7.8: It contains some statistics about the outcomes of the four preceding queries. The table include the cardinalities of the input variables as well the cardinality of the output of the proposed queries.

Nation	# variations from 1kG	# snps	# correspondences
Finland	33743673	1353351	5589
Japan	409769205	14	396
China	1211308383	9	0
U.K.	1200326717	6	423

7.2.4 Mutations occurring in cell-specific enhancers

In [8] the authors developed a new fine-mapping algorithm to identify candidate causal variants for 21 autoimmune diseases from genotyping data. They found out that about 60% of likely causal variants map to enhancer-like elements, with preferential correspondence to stimulus dependent CD41 T-cell enhancers that respond to immune activation by increasing histone acetylation and transcribing non-coding RNAs. Unfortunately, it is not trivial to associate the enhancer with its corresponding gene, since it is situated within some hundreds of kilobases from the gene it regulates.

The Post Doc researcher of Polimi, Pietro Pinoli, in his PhD thesis [20] has extended that work with the help of Noam Shoresh, one of the author of the aforementioned study. In particular, they extended the investigation of such phenomenon to many different human cell lines. They attempt to study whether mutations which occur in cell specific enhancers are related with any particular disease or trait.

The computational experiment has been carried on by formulating a complex GMQL query, jointly using two genomic datasets. The mutations data were taken from a GWAS dataset (which was no public available on GMQL web service) while enhancer regions were extracted from the Encode dataset. During the experiment, they focused on a particular histone modification, the acetylation at the 27th lysine residue of the histone protein 3 (H3K27Ac), which can be captured by a ChIP-seq experiment. The modification H3K27Ac is defined as "active enhancer mark" since it is known to encourage enhancer activation.

The query allows to find the mutations occurring in cell-specific enhancers and to understand the resulting disease or phenotypic trait.

The query that is proposed in the following lines has been re-adapted from the one proposed by Pietro Pinoli in his PhD thesis and it is reported as example of richness of expression of the GMQL query language, since two new GWAS datasets have been made available and integrated in the architecture.

The representation in Figure 7.6 and the schema in Figure 7.7 help the reader to understand the operations performed in the query.

```
/* load studies from GWAS Catalog and FinnGen dataset
   mapped to trait "schizophrenia" */
GWAS = SELECT(mapped_trait == "schizophrenia") GWAS;
FINN = SELECT(phenocode == "F5_SCHZPHR") FinnGen;

/* load data from ENCODE dataset */
Ac = SELECT(target__genes__targets == "/targets/H3K27ac"
            "-human/") HG19_ENCODE_NARROW_2020_01 ;

/* update ENCODE regions */
large = PROJECT (region_update : LEFT AS LEFT + peak -
                 1500 , RIGHT AS LEFT + peak + 1500 ) Ac ;

/* merge replicas together */
REP = COVER(1, ANY; groupby: biosample__ontology__name)
      large ;

/* find cell type-specific enhancers */
```

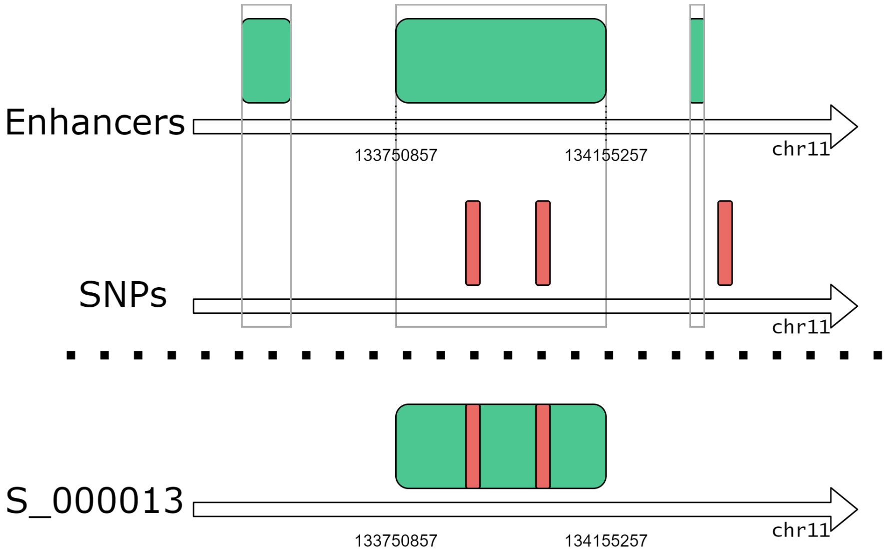


Figure 7.6: Schematic representation of the query in section 7.2.4. The green regions are the cell type-specific enhancers while the red ones are the SNPs. The MAP operator counts, for each enhancer region, how many overlapping SNPs there are in the GWAS datasets. Those identified SNPs are relevant from a biological point of view since they occur in non-coding regions (enhancers).

```
S = COVER (1, 2) REP ;
RepCount = MAP( ) REP S ;
CSE = SELECT (region: count_REP_S > 0) RepCount ;

/* insert the trait into regions */
GWAS_trait = PROJECT(region_update: mapped_trait AS META
                (mapped_trait, STRING)) GWAS;
FINN_trait = PROJECT(region_update: mapped_trait AS META
                (phenocode, STRING)) FINN;

/* union the studies from GWAS Catalog and FinnGen into a
   single dataset */
UNI = UNION() GWAS_trait FINN_trait;

/* find mutations occurring in those enhancers */
M = MAP(bag AS BAG(mapped_trait)) CSE UNI ;
N = SELECT (region: count_CSE_UNI > 0) M;
P = PROJECT (count_CSE_UNI , bag) N;
MATERIALIZE P into MUTATION ;
```

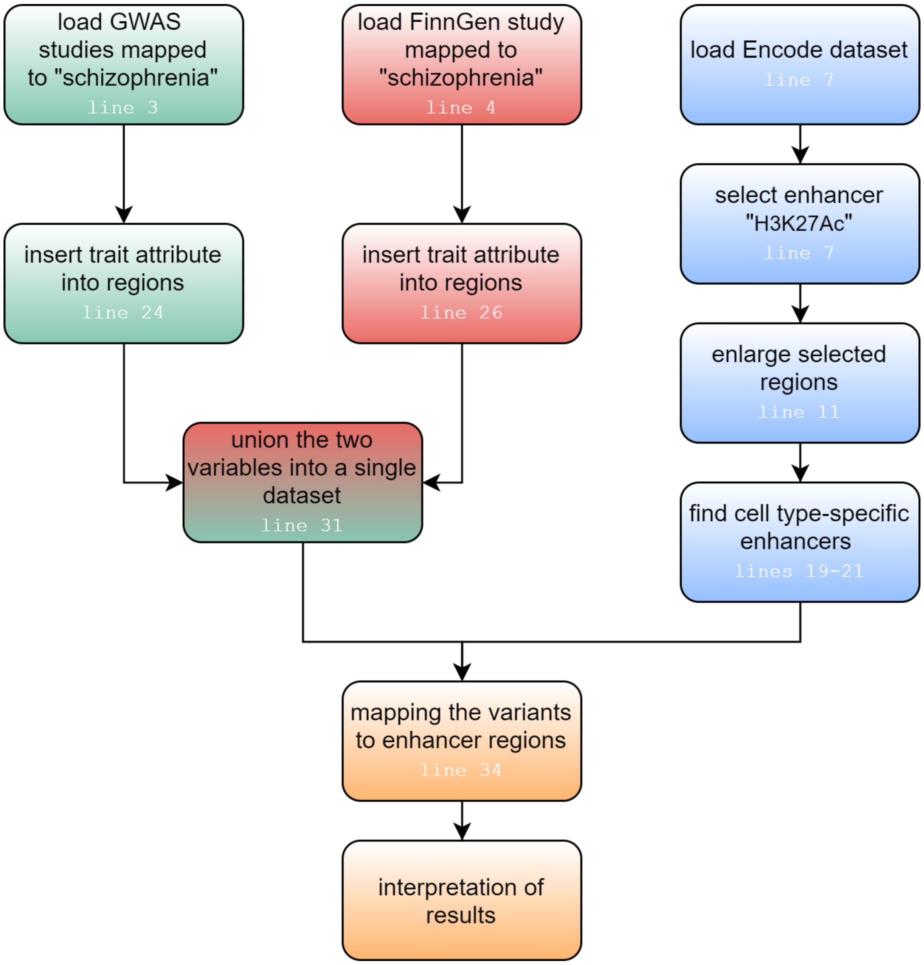

load GWAS studies mapped to "schizophrenia"
line 3
load FinnGen study mapped to "schizophrenia"
line 4
load Encode dataset
line 7
insert trait attribute into regions
line 24
insert trait attribute into regions
line 26
select enhancer "H3K27Ac"
line 7
enlarge selected regions
line 11
union the two variables into a single dataset
line 31
find cell type-specific enhancers
lines 19-21
mapping the variants to enhancer regions
line 34
interpretation of results

Figure 7.7: Schematic flow of execution of the proposed GMQL query 7.2.4. The three datasets are first pre-processed separately. The studies from GWAS Catalog and FinnGen are united into a single dataset and then enhancer regions from Encode dataset are mapped into regions from the unified one. In each box of the flow is indicated the line number of the corresponding operation of the query. The aim of this schema is to help the reader, who may is not familiar with the GMQL query language, to understand the underlying reasoning.

For further details about the reported GMQL operators, please refer to [22] and to section 3.3.

In the first lines are uploaded the GWAS studies mapped to "schizophrenia" from GWAS Catalog and FinnGen dataset; it is also loaded the most recent available version of HG19_ENCODE_NARROW dataset. This last operation is done using the GMQL predicate `SELECT`, specifying for Encode the DNA regions which are enriched by H3K27Ac.

Line 11 allows to update the coordinates of the previously selected Encode regions, enlarging them by 3000 base pairs around the peaks. Region updates can be performed using the predicate `PROJECT`. This operation defines the enhancer regions.

Line 15 applies the operator `COVER` over the Encode samples, using the *groupby* option. It computes the result grouping the input dataset samples by the values of their "biosample__ontology__name" metadata attribute.

Lines 19, 20 and 21 filter the regions which are cell type-specific enhancers. To distinguish cell type-specific enhancers from shared ones, we considered their frequency; we are looking for those peaks of H3K27Ac that occur in no more than two cell lines among all the samples that we considered. The `COVER (1, 2)` operation considers all areas defined by a minimum of one overlapping region up to two of them; output region attributes include only region coordinates. The operation `MAP ()` allows to retrieve the original regions with all their region attributes, adding the information of their frequency. Finally, using the `SELECT` operator are extracted only the regions identified in line 20.

Lines 24 and 25 exploit the operator `PROJECT` allowing to insert for each region, the region attribute "mapped_trait". In GWAS studies the phenotype attribute is a metadata, therefore it is necessary to copy it also as region attribute, to reproduce the proposed query. These operations simply copy the metadata "mapped_trait" (for studies from GWAS Catalog) or "phenocode" (for studies from FinnGen) into the region attribute "mapped_trait".

Line 31 creates a dataset called UNI containing all the samples from GWAS_trait and FINN_trait datasets.

Line 34 and 35 are the core operations of the whole query. The `MAP` operator adds to each region of the Encode dataset (previously pre-processed in the former lines) a counter corresponding to the number of overlapping regions (e.g. same coordinates) of UNI dataset. The option "bag" adds a further region attribute to Encode regions, filling it with the values of the attributes "mapped_trait" of the mapped GWAS regions, comma-separated. The operator `SELECT` extracts only the Encode regions which have at least one corresponding GWAS mutation.

Line 36 uses the operator `PROJECT` to remove from the output regions all the attributes apart from the coordinates and the two specified ones.

As last operation, the dataset P is materialized so it can be explored or downloaded.

The interpretation of the output of this query is at the same time simple and very important for its biological meaning. The output is made of region and metadata files; a small examples of resulted regions is proposed in Table 7.9. Metadata files contain a lot of attributes resulting from the operations `MAP` and `COVER`, obtained by the transformation of the metadata of the input files.

The resulting region files are easy to interpret even for people who are not domain-expert. Observing the first entry of Table 7.9, we can see that in the current enhancer region occurred a mutation of a nucleotide (since the attribute `count` is equal 1), which is known to be associated with “schizophrenia”. In the tables appear only the traits “schizophrenia” or “F5_SCHZPHR” since are the only selected ones. If for an enhancer region are found more than one overlapping SNPs (attribute “count” >1), in the column “trait” is reported the same multiplicity.

The GMQL query doesn’t aim to filter out only the truly causal variants (which can be performed with many FINE-MAPPING algorithms), but *its strength is to identify the variants which occur in non-coding regions, in particular the enhancer regions in which occurred the modification H3K27Ac.*

As described in the introduction of this chapter, the main difficulty of interpreting the SNPs occurring in non-coding regions is due to the limited and incomplete knowledge of non-coding regulatory elements, their mechanisms of action and the cellular states and processes in which they function. Thanks to the expressiveness of this query, the comprehension of the consequences of some variants occurring in cell type-specific enhancer is made easier.

Table 7.9: It contains a small portion of the output of the proposed GMQL query in section 7.2.4. Each rows represents a region, uniquely identified by its coordinates and by the two additional attributes “count” and “trait”. The attribute “count” stores the number of mutations from the GWAS datasets that occur in the current region; the attribute “trait” contains the trait(s) mapped to the SNPs that occur in that region.

chr	left	right	strand	count	trait
chr6	25163149	25184115	*	1	schizophrenia
chr11	133750847	134155257	*	2	schizophrenia (x2)
chr3	177104856	177107871	*	14	F5_SCHZPHR (x14)

statistics:

- *Execution time: 07:58:17*
- *Number of regions: 39008*
- *Number of samples: 318*
- *Size: 4269.22 MB*

Chapter 8

Conclusions and future perspectives

This thesis describes the entire process performed to achieve the integration of the two GWAS sources `GWASCatalog` and `FinnGen` into the META-BASE architecture. It has been the first attempt to integrate GWA studies into the integration framework developed in the context of the GeCo project of Polimi.

The META-BASE architecture is a consolidated framework to integrate many genomic sources with each other, allowing to create complex queries between multi-omic sources using the GMQL query language or to surf genomic data using the GenoSurf web interface. It reaches the integration starting from mapping the original sources into a shared conceptual schema, the Genomic Conceptual Model.

Genomic data are made available by many consortia each one using its own data schema and it is difficult for researchers or biologists to study data that are mapped over different schemes. The most challenging tasks concerning genomic data is the *tertiary analysis*, which goal is to extract meaningful information from raw genomic data.

In order to help researchers to study the human genome and its functional role, the GeCo project has mapped the genomic datasets from the major consortia around the world into a common conceptual schema.

The GCM used so far couldn't gather GWA studies, since they are conducted in a case-control setup and following the phenotype-first approach. The GCM has been extended to accept also studies based on cohort of people and not on single person. GWA studies which have been integrated have not privacy issues, since they have been already anonymized by the source consortia. Data are aggregated over cohort of people, so information about single person that participated in the studies are not reachable.

After a modelling work, this thesis proceeded with the implementation of the integration pipeline also for these two GWAS sources, writing the Scala classes and methods required for the integration.

As results of the efforts spent in this thesis, the two genomic sources `GWAS Catalog` and `FinnGen` are perfectly integrated into the META-BASE architecture and are publicly available to be queried using the GMQL query language. As example of the potentiality of writing GMQL queries over multiple integrated datasets, included GWAS data, has been reported a few queries which emphasize the importance of coupling GWAS data with annotation genomic sources.

This thesis has extended the GCM to fit also GWAS data, laying the foundations to integrate many other GWA sources. All GWA studies are based on a case-control setup, so integrating new sources is made easier by the newly proposed GCM.

With the proposed GMQL queries of Chapter 7 we have shown that the META-BASE integration architecture allows to run multi-omic queries, which provide very important insights driving biological discoveries.

Future possible works could be related to the improvement of the GenoSurf browser, to allow users to surf even on GWAS data.

Future tasks could concern the implementation of a new module of the META-BASE, performing fine-mapping over the integrated GWA studies. The main issue with GWA studies is that is difficult to identify truly causal SNPs for a given phenotype, due to linkage disequilibrium. The challenging part of implementing a fine-mapping algorithm would be to retrieve all the data that are needed as input since fine-mapping algorithms often require sensitive data difficult to retrieve for privacy issues.

Finally, the integration modules implemented in this thesis could be optimized to reduce the integration time during future runs or allowing multi-threads executions.

Chapter 9

Dictionary

Readers who are not biologist may face some terms or acronyms difficult to understand. Following are listed the main biological names used in this thesis with a short explanation.

- alternative allele (effect allele): synonyms of “risk allele”. It is the allele which is associated with the phenotype under consideration, since it is more frequent in cases with respect to controls group.
- ancestry: it encloses a bunch of information about the cohort origin. The country of recruitment and the category of belonging of people (e.g. Asian) may are included.
- association: a correlation statistically significant between a SNP and the trait, phenotype or disease under study. The association are often improved with measures like p-value, which represents its importance.
- cases: a group of people affected by a phenotype, trait or disease that is compared with controls group to find SNPs in which the two groups significantly differ.
- cohort: a group of people that shares a characteristic. A GWA study is based on a cohort of people, made of cases and controls.
- controls: a group of people not affected by the phenotype, trait or disease that is the subject of the study. They are compared with the cases group and they build the cohort of the GWA study.
- copy number variants: it is a phenomenon in which sections of the genome are repeated and the number of repeats in the genome varies between individuals. They are the most common structural variations.
- DNA: is a molecule composed of two polynucleotide chains that form a double helix structure. The sequence of the nucleotides encodes all the information about development, functioning, growth and reproduction of all living organisms.

- DNA annotation: is the process of identifying the locations of genes and all of the coding regions in a genome and determining their functional role.
- endpoint: it indicates the medical measures referring to occurrence of a disease or a trait. In the context of the FinnGen project, it is used as synonym of the phenotype under consideration.
- enhancer: it is a short (50–1500 base-pairs) region of DNA that can be bound by proteins (activators) to increase the likelihood that transcription of a particular gene will occur.
- fine-mapping: is the process by which a trait-associated region from a GWA study is analysed to identify the particular genetic variants that are likely to causally influence the examined trait. The SNPs are filtered taking into consideration the linkage disequilibrium between the loci to which the SNPs belong.
- gene: is a basic unit of heredity and a sequence of nucleotides in DNA or RNA that encodes the synthesis of a gene product, either RNA or protein.
- GWAS: Genome-wide association study. Two cohorts of people (cases and controls) are compared to detect the SNPs in which they significantly differ.
- haplotype: it is a group of alleles in an organism that are inherited together from a single parent.
- histone acetylation: is an epigenetic modification that is unequivocally associated with increasing the propensity for gene transcription. Acetylation removes the positive charge on the histones and, as a consequence, the condensed chromatin is transformed into a more relaxed structure that is associated with greater levels of gene transcription.
- initial: the first stage of a GWA study. Often they are repeated to make stronger the evidences for the found SNPs.
- linkage disequilibrium: it is a phenomenon for which the presence of an allele in a locus is influenced by the presence of an allele in another locus. The two loci are said to be in linkage disequilibrium.
- minor allele: it is the second most common allele that occurs in a given population, for a given phenotype and a position over the genome. The minor allele, in many GWASs on complex diseases, is the risk allele [16], that is the allele associated with the phenotype under consideration.
- nucleotide: is the building block for the DNA and RNA molecules. Many nucleotides arrange together to form the “double helix” structure and their sequence encode the genetic information to produce proteins and to orchestrate life.

- phenotype: it is the observation of a feature of an organism.
- promoter: it is a small portion of DNA located near a gene, upstream on the DNA. It provides to the enzyme RNA polymerase a secure initial binding site so to start the transcription of the gene.
- reference allele: it is the most frequent allele in the controls group for a given position over the genome.
- replication: is the second stage of a GWA study. It should be conducted in an independent dataset drawn from the same population as the initial stage, in an attempt to confirm the effects of the found SNPs in the GWAS target population.
- risk allele: is an allele which is more frequent in the cases respect to controls group. This is called "risk" since there are evidence that it is causal for the phenotype under study.
- RNA: is a macro-molecule assembled as a chain of nucleotides. It is, like DNA, an essential molecule for life. It is implied in various biological roles like coding, decoding, regulation and expression of genes.
- SNP: Single Nucleotide Polymorphism. It is the substitution of a single nucleotide at a specific position in the genome in a relevant percentage of people in the case group respect the control one.
- strand: the DNA is made of two paired sequences of nucleotides that are wrapped around each other. The strand indicates which one of the two sequences is considered. The DNA double helix is made of two strands identified by the notations *5′-3′* and *3′–5′*, spotting the position of the carbon atoms on the deoxyribose molecule.
- structural variation: it consists of many kinds of variation in the genome such as deletions, duplications, copy-number variants, insertions, inversions and translocations.
- TPM: is a normalization method for RNA-seq, should be read as "for every 1,000,000 RNA molecules in the RNA-seq sample, x came from this gene/transcript".
- trait: is a distinct variant of a phenotypic feature of an organism. It may be inherited (e.g. eye color) or influenced by the environment in which you live (e.g. ability to play a sport).
- UTR: stands for "untranslated region", that are that parts of mRNA which are not translated into proteins.

Chapter 10

References

[1] A. Bernasconi, A. Canakoglu, M. Masseroli, and S. Ceri. Meta-base: a novel architecture for large-scale genomic metadata integration. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2020. doi:10.1109/TCBB.2020.2998954.

[2] A. Bernasconi et al. Conceptual modeling for genomics: Building an integrated repository of open data. *Springer*, pages 325–339, 2017. doi:10.1007/978-3-319-69904-2_26.

[3] A. Buniello et al. The nhgri-ebi gwas catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Research*, 47(D1):D1005–D1012, 2019. doi:10.1093/nar/gky1120.

[4] A. Canakoglu et al. Genosurf: metadata driven semantic search system for integrated genomic datasets. *Database*, 2019, 2019. doi:10.1093/database/baz132.

[5] F.S. Collins and L. Fink. The human genome project. *Alcohol Health Res World*, 19(3):190–195, 1995.

[6] EMBL-EBI and NHGRI. *GWAS Catalog website*. URL: https://www.ebi.ac.uk/gwas/.

[7] Genomics England. *100k Genomes Project*, 2012. URL: https://www.genomicsengland.co.uk/about-genomics-england/the-100000-genomes-project/.

[8] KH. Farh, A. Marson, J. Zhu, et al. Genetic and epigenetic fine mapping of causal autoimmune disease variants. *Nature*, 518:337–343, 2015. doi:10.1038/nature13835.

[9] FinnGen. FinnGen documentation of r5 release, 2020. URL: https://finngen.gitbook.io/documentation/.

[10] A Frankish, M. Diekhans, et al. Gencode reference annotation for the human and mouse genomes. *Nucleic Acids Research*, 47(D1):D766–D773, 2019. `doi:10.1093/nar/gky955`.

[11] T. Hubbard et al. The ensembl genome database project. *Nucleic Acids Research*, 30(1):38–41, 2002. `doi:10.1093/nar/30.1.38`.

[12] International Genome Sample Resource (IGSR). *Meeting Report: A Workshop to Plan a Deep Catalog of Human Genetic Variation*, 2007. URL: `https://www.internationalgenome.org/sites/1000genomes.org/files/docs/1000Genomes-MeetingReport.pdf`.

[13] European Molecular Biology Laboratory's European Bioinformatics Institute. *Ensembl website*. URL: `https://www.ensembl.org/info/about/index.html`.

[14] National Cancer Institute. *The Cancer Genome Atlas Program*, 2006. URL: `https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga`.

[15] D. Karolchik, A.S. Hinrichs, and W.J. Kent. The ucsc genome browser. *Current Protocols in Bioinformatics*, Chapter 1:Unit1.4, 2009. `doi:10.1002/0471250953.bi0104s28`.

[16] T. Kido et al. Are minor alleles more likely to be risk alleles? *BMC Med Genomics*, 11(1):3, 2018. `doi:10.1186/s12920-018-0322-5`.

[17] J. MacArthur et al. The new nhgri-ebi catalog of published genome-wide association studies (gwas catalog). *Nucleic Acids Research*, 45:D896–D901, 2017. `doi:10.1093/nar/gkw1133`.

[18] M. Masseroli et al. Genometric query language: a novel approach to large-scale genomic data management. *Bioinformatics*, 31(12):1881–8, 2015. `doi:10.1093/bioinformatics/btv048`.

[19] M. Masseroli et al. Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying. *ScienceDirect*, 111:3–11, 2016. `doi:doi:10.1016/j.ymeth.2016.09.002`.

[20] P. Pinoli. *PhD Thesis: Modeling and Querying Genomic Data*, 2016. URL: `http://hdl.handle.net/10589/132099`.

[21] Data-Driven Genomic Computing (GeCo) project. *GMQL - Biological examples*. URL: `http://www.bioinformatics.deib.polimi.it/genomic_computing/GMQLsystem/doc/GMQL_biological_examples.pdf`.

[22] Data-Driven Genomic Computing (GeCo) project. *GMQL - Introduction to the language*. URL: `http://www.bioinformatics.deib.polimi.it/genomic_computing/GMQLsystem/doc/GMQL_introduction_to_the_language.pdf`.

[23] E.M. Ramos et al. Phenotype-genotype integrator (phegeni): synthesizing genome-wide association study (gwas) data with existing genomic resources. *European journal of human genetics*, 22(1):144–7, 2014. doi:10.1038/ejhg.2013.96.

[24] G. Rudy. *Hitchhikers guide to ngs*, 2010. URL: http://goldenhelix.com/media/pdfs/whitepapers/Hitchhikers-Guide-to-NGS.pdf.

[25] S. Schuster. Next-generation sequencing transforms today's biology. *Nat Methods*, 5:16–18, 2007. doi:10.1038/nmeth1156.

[26] M. Steri, M.L. Idda, M.B. Whalen, and V. Orrù. Genetic variants in mrna untranslated regions. *Wiley interdisciplinary reviews RNA*, 9(4):e1474, 2018. doi:10.1002/wrna.1474.

[27] X. Yang and M.E. Lippman. Brca1 and brca2 in breast cancer. *Breast Cancer Research and Treatment*, 54:1–10, 1999. doi:10.1023/a:1006189906896.

[28] École polytechnique fédérale de Lausanne. *Scala programming language website.* URL: https://www.scala-lang.org/.