POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Dipartimento di Elettronica, Informazione e Bioingegneria

Master of Science Degree in Computer Science and Engineering

RoadmapImporter modules for GMQL-Importer: integration of Roadmap Epigenomics Project data and metadata into a GDM repository

Supervisor:

PROF. MARCO MASSEROLI

Co-supervisor:

ARIF CANAKOGLU, PHD

Master Graduation Thesis by:

RICCARDO MOLOGNI
Student Id n. 852416

Accademic Year 2016-2017

Agli amici e compagni della RES-BG che,
nonostante tutto, resistono.

Abstract

One of the major concerns of bioinformatics has always been the development of tools capable of integrating the large amount of biological data available through different sources, with the purpose to allow a more easy, efficient, and effective extraction of novel and useful knowledge. Since the development of the Next Generation Sequencing (NGS) techniques, a more cost-effective and fast method of DNA sequencing that has led to an enormous increase of the volume of genomic data available, the problem of integrating a large amount of heterogeneous data accessible through a variety of technology has become even more pressing.

In this context, the Genomic Computing (GeCo) team of Politecnico di Milano has developed the Genomic Data Model (GDM) and the GenoMetric Query Language (GMQL) which provide abstractions for genomic data and their metadata, and the possibility to query them in a simple and high-level fashion. To be queried, though, the data must be downloaded from the sources, transformed to be compliant to the GDM, and then added to a GDM repository connected to a GMQL implementation.

The objective of this thesis project is to integrate in an automated way the Roadmap Epigenomics Project (REP) biological data and metadata into a GDM repository, so that the REP data can be queried using the GMQL language and potentially contribute to the discovery of new knowledge in different areas of biology and medicine. The data integration is performed by developing some additional modules for GMQL-Importer, a modular and fully configurable tool regularly used by the GeCo team to add biological data and metadata to GDM repositories. These modules are extending the capacity of the GMQL-Importer program, adding the REP source to the pool of sources from which data and metadata can be automatically downloaded, transformed, and imported in a GDM repository, fully supporting the maintenance and extension of such a repository.

Estratto

L'integrazione di dati biologici rappresenta da sempre una sfida. Questo è dovuto sia all'estrema complessità ed eterogeneità intrinseca dei dati biologici che all'enorme quantitativo di sorgenti che si sono andate cumulando nel corso degli anni. Infatti ogni progetto che prevede la condivisione di dati pubblici è gestito in modo indipendente dagli altri, da gruppi di persone diversi senza alcun tipo di accordo o convenzione e utilizzando tecnologie e infrastrutture estremamente variegate. Questa mancanza di coordinamento rende difficile l'integrazione dei dati biologici finalizzata all'estrazione di nuove conoscenze. Dal punto di vista computazionale un processo di integrazione deve tenere in considerazione alcuni aspetti fondamentali come: la differenza di dimensione, formato e numero di attributi dei dati; la presenza di dati errati e da scartare; la selezione di dataset rilevanti rispetto al contesto; l'integrazione dei dataset selezionati, spesso con caratteristiche discordanti tra loro e la capacità del processo di integrazione di gestire grandi quantità di dataset mantenendo prestazioni accettabili. L'eterogeneità dei dati biologici è dovuta anche alla gran varietà di metodi e tecnologie usate per produrli. Inoltre le tecniche usate e le tecnologie sono in continua evoluzione e il loro miglioramento ha consentito, negli ultimi anni, di incrementare enormemente il volume dei dati prodotti. In particolare, le tecniche di Next Generation Sequencing (NGS) per il sequenziamento del materiale genetico hanno permesso di ridurre enormemente i costi e i tempi di produzione. La possibilità di ottenere facilmente e in modo economico un gran quantitativo di dati ha spostato l'attenzione della comunità scientifica da come ottenere questi dati a come gestirli in maniera efficiente ed efficace al fine di facilitare l'estrazione di informazioni utili. Allo stesso tempo, è diventato evidente che la conoscenza che è possibile estrarre da dati provenienti da più sorgenti, quindi coinvolgendo tipi di dati molto diversi tra loro, sia decisamente maggiore di quella ottenibile tramite analisi di singole sorgenti, spesso specializzate. Ciò ha portato allo sviluppo di un numero sempre maggiore di approcci per l'integrazione di dati, solitamente basati sull'intelligenza artificiale o tecniche di big data.

In questo scenario, anche il gruppo di Genomic Computing (GeCo) del Politecnico di Milano ha proposto una sua soluzione per l'integrazione di dati genomici e l'estrazione di informazioni dalle collezioni di dati ottenute. La loro soluzione prevede, innanzitutto, l'utilizzo di un modello di dati chiamato Genomic Data Model (GDM) per uniformare tutti i dati da integrare. Una volta che l'integrazione è avvenuta, questi possono essere interrogati usando il GenoMetric Query Language (GMQL). GMQL è un linguaggio di alto livello che permette di eseguire le classiche operazioni tipiche dei database relazionali, estese per permettere di lavorare con regioni del genoma basandosi sul concetto di distanza genomica. GMQL è pensato per effettuare analisi terziaria su dati genomici, generando come risultato delle query uno o più dataset di interesse. Prima di poter essere interrogati, però, i dati devono essere integrati

e resi conformi a quanto previsto da GDM. Questo è fatto grazie a un programma appositamente sviluppato, chiamato GMQL-Importer. GMQL-Importer è progettato per scaricare i dati e i relativi metadati da una o più sorgenti e trasformarli fino a renderli compatibili con il GDM. GMQL-Importer è stato sviluppato in modo totalmente generale e modulare, quindi è possibile aggiungere nuovi moduli che consentano di integrare dati provenienti da sorgenti nuove e trasformare tipi di dati precedentemente non previsti.

Lo scopo di questo progetto di tesi è quello di sviluppare una serie di nuovi moduli per GMQL-Importer, chiamati RoadmapImporter, per consentirgli di scaricare e trasformare dati provenienti dal Roadmap Epigenomics Project (REP). I dati così integrati verranno poi aggiunti a un repository GDM. I moduli RoadmapImporter sono due: RoadmapDownloader, che consente di scaricare i dati e metadati specificati dall'utente dalla sorgente dati di REP, e RoadmapTransformer che, invece, è dedicato alla trasformazione dei diversi dati scaricati in dataset omogenei compatibili con GDM. La tesi è motivata dal desiderio di estendere l'attuale bacino di dati su cui è possibile effettuare query GMQL (che attualmente comprende dati principalmente provenienti da ENCODE e TCGA) e potenzialmente contribuire alla scoperta di nuovo sapere nei campi della biologia e della medicina.

Il progetto è diviso in tre fasi distinte. Nella prima fase abbiamo analizzato la sorgente e tutti i dati e metadati che mette a disposizione. Questa fase è particolarmente delicata perché i dati a disposizione sono molto vari, sono prodotti di diverse fasi di elaborazione (si spazia dai dati grezzi prodotti dagli esperimenti biologici a dati ricavati da analisi secondarie) e sono distribuiti e/o duplicati in diverse repository. Spesso i dati disponibili non sono documentati o non sono documentati in modo approfondito. Una volta che i dati e metadati disponibili sono stati identificati, è stato necessario selezionare i dati rilevanti per il contesto applicativo da importare nel repository GMQL. I dati selezionati sono soggetti a un'ulteriore analisi per identificare le trasformazioni necessarie per rendere il formato dei dati e metadati compatibili con quanto richiesto da GDM. L'ultima fase riguarda l'effettiva implementazione dei moduli partendo dalle analisi effettuate in precedenza. Durante questa fase, oltre ai due moduli necessari a portare a termine l'integrazione, sono state sviluppate anche altre nuove funzionalità per GMQL-Importer insieme ad alcune migliorie generali del codice.

La tesi è strutturata in modo da ricalcare le varie fasi del progetto. Nel Capitolo 2 vengono forniti alcuni concetti, prerequisiti per la comprensione della tesi e delle sue motivazioni. Nella prima parte del capitolo vengono introdotti brevemente GDM e GMQL presentando gli aspetti più rilevanti per questo contesto specifico. Nella seconda parte del capitolo è presente una breve introduzione di REP, elencando le sue motivazioni, obiettivi, importanza e organizzazione. Nel Capitolo 3 viene descritto lo stato dell'arte di GMQL-Importer prima dell'inizio di questo progetto. Il Capitolo 4 corrisponde alla descrizione della prima fase del progetto e viene descritta l'analisi della sorgente REP effettuata, elencando i dati e metadati che fornisce e dove sono localizzati. Nel Capitolo 5 viene descritto il processo di selezione dei dataset rilevanti e di progettazione delle trasformazioni da applicare, descrivendo dati e metadati prima e dopo il processo di trasformazione. Il capitolo 6 è dedicato invece alla terza fase del progetto e presenta i moduli implementati concentrandosi sui metodi più rilevanti, le interazioni tra i moduli e il corpo principale del programma e quelle tra i vari metodi. Le scelte di implementazione prese vengono giustificate. Il Capitolo

7 contiene informazioni analoghe al capitolo precedente, ma riferite ai metodi e funzionalità esterne ai moduli, ovvero quelle funzionalità che possono essere eseguite indipendentemente dai moduli scelte. Nel Capitolo 8 presentiamo i risultati ottenuti insieme ad alcuni dati statistici e alla descrizione dei test effettuati. Il Capitolo 9 è dedicato ad alcune considerazioni finali sul lavoro svolto e gli obbiettivi raggiunti. Nell'ultimo capitolo, il Capitolo 10, presentiamo alcune possibilità di estensione futura.

Contents

List of Figures

List of Tables

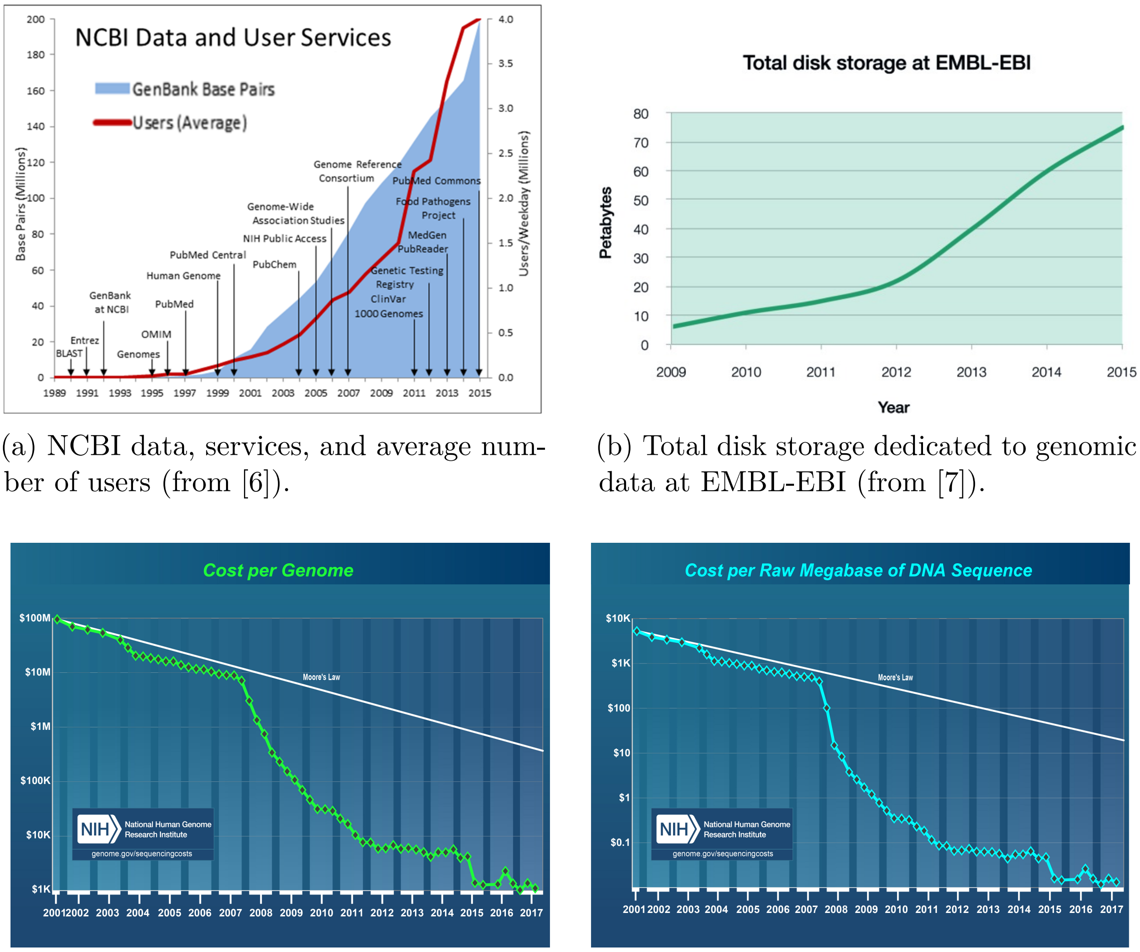
Chapter 1

Introduction

The integration of biological data has always been challenging. This is due to an extreme heterogeneity and intrinsic complexity of the data itself, but also the fragmentation of the data among different sources, usually managed by the team or institution that has produced them without any coordination with other groups. When approaching the problem of integrating biological data coming from different sources with the goal to extract additional biological knowledge from multiple datasets that can not be gained from any single dataset alone, some computational issues must be taken into account:

- different size, format and dimensionality of datasets;
- presence of noise and data collection biases in datasets;
- effective selection of informative datasets;
- effective incorporation of concordant and discordant datasets;
- scalability with the number and size of datasets.

The biological data, in fact, are produced using a large number of various techniques and technologies in constant evolution. In particular, the development of new high-throughput techniques for DNA sequencing, such as the Next Generation Sequencing (NGS) technologies, that allow to produce a large amount of genomic data at a small cost and in short period of time, has shifted the focus of the research community from how to produce the data to how to store the increasing volume of data generated by biological experiments in an effectively way, and to retrieve useful information when they are needed, such that new knowledge can be extracted. In Figure 1.1 some charts show the continuously increase of genomic data volume (1.1a, 1.1b) and the related cost decrease (1.1c, 1.1d) thanks to the introduction of new sequencing technology. Meanwhile, in the last years, it has become evident to the research community that to analyze data of different types coming from multiple sources is much more meaningful from the point of view of the knowledge that can be potentially extracted analyzing a single source, as it can be seen by looking at the rich literature produced about the topic (see, for example, [1–5]). New approaches to biological data integrations have been developed by numerous research teams, mainly based on machine learning and big data approaches.



(a) NCBI data, services, and average number of users (from [6]).

(b) Total disk storage dedicated to genomic data at EMBL-EBI (from [7]).

(c) Cost per genome trend against the Moore's Low (from [8]).

(d) Cost per Megabase trend against the Moore's Low (from [8]).

Figure 1.1: The rapid increasing of genomic data volumes coincide with an even faster decrease of cost thanks to the introduction of new sequencing technologies.

The Genomic Computing (GeCo) team of the Politecnico di Milano [9] has proposed a solution to the integration of heterogeneous genomic data collected from different sources and the information retrieval from the collection of integrated data. First they have developed the *Genomic Data Model (GDM)* which provides a powerful abstractions and format-independent representations for a large variety of genomic data based on DNA region coordinates and related metadata. Once the data are integrated into a GDM repository, they can be queried using the *GenoMetric Query Language (GMQL)*, that allows to perform the traditional relational operations over datasets and samples, plus some genomic-specific operations based on genomic distance. GMQL allows to perform high-level operations over datasets originally coming from different sources and containing different types of data (e.g., mutations, expressions, regulation experiments), and it produces in output one or more datasets of relevant data and the related metadata. The actual data integration is performed when the data are transformed from the source format to a format compatible with GDM. This is done using an ad-hoc program called *GMQL-Importer*. GMQL-Importer is designed to perform all the required operations in the most general and modular way, so that new sources and types of data can be gradually added to the set of integrated data imported in a GDM repository. GMQL-Importer is also fully configurable by the user to fit his needs. The interaction occurring between GDM, GMQL, and GMQL-Importer are represented in Figure 1.2

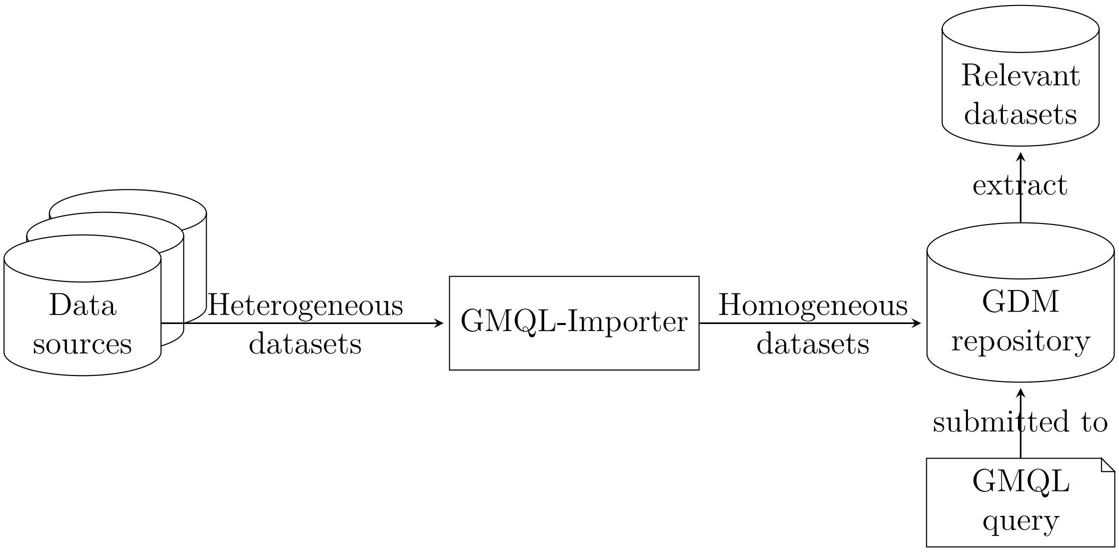


Figure 1.2: Schematic representation of the interaction between some data sources, GMQL-importer, GDM and GMQL-Importer. GMQL-download datasets of heterogeneous data, transform them in homogeneous datasets compatible with GDM, and load them in a GDM repository target, completing the data integration. Data in a GDM repository can be queried using GMQL to extract relevant data. It is important to notice that the data and metadata are still heterogeneous from a semantic point of view after the transformation. The integration process only alters their structure to make them compliant to GDM, but remain unaffected from the contents point of view.

The goals of this thesis are:

- to identify the most relevant Roadmap Epigenomics Project (REP) data to make available for tertiary analysis through GMQL;
- to group the identified REP data into semantically homogeneous datasets;
- to design, for each dataset, the most appropriate transformations to make them compatible with the GDM;
- to associate to each sample as many useful metadata as possible and retain the immaterial ones;
- to identify the metadata transformations required to make the metadata human readable and compliant to GDM format;
- to actually integrate in a GDM repository the semantically homogeneous datasets of relevant REP data and metadata by applying the designed transformation in an automated way.

To perform the integration process, we exploit GMQL-Importer, developing some new modules to extend the current functionalities of the application, so as to make it capable to integrate in an automated way the relevant REP data and metadata into a GDM repository. In this way we take advantage of a well-functioning and tested data transformation pipeline, along with all its pre-existing functions. A side goal is to add as many improvements and new functionalities to GMQL-Importer as possible. The new group of modules in charge of the REP integration is called *RoadmapImporter*, and they consist in two distinct and potentially independent modules:

1. *RoadmapDownloader*: the module in charge of downloading the relevant REP data and metadata specified by the user from the source repository;

2. *RoadmapTransformer*: the module devoted to transform different type of REP data in homogeneous datasets compatible with the GDM and ready to be imported in a GDM repository.

The thesis is motivated by the wish of enriching the current pool of data and metadata available to be queried using GMQL and all the related services (e.g., REST APIs, PyGMQL, RGMQL). In this way the REP data are at the disposal of the research community to be referred, compared with data from other sources (currently data from ENCODE[1] and TCGA[2] are available in GMQL) and used to generate new custom datasets that may be used to discover new knowledge in the different fields of biology and medicine.

The project is divided in three distinct phases. The first phase consists in the analysis of the source and all the available data and metadata. This phase is not trivial because the data are coming from different laboratories, they are of different types and stages[3] of elaboration (from the raw data coming from the biological experiment to secondary analysis data), and they are distributed and/or duplicated in different repositories. Frequently, data and metadata are not well documented. Once that the source has been analyzed and all the data and metadata are identified and located, the second phase, consisting on the selection of the relevant data and metadata to import into the GDM repository, can start. In this phase, the data selected to be imported are subjected to an additional analysis to decide which transformation operate over the data, how to group them in coherent and homogeneous data, and which data with a format incompatible with GDM (for example binary data) to exclude. A particular attention must be dedicated to metadata. Metadata must be in the format required by GDM (files of tab separated name-value pairs), but only the useful ones must be retained, and the duplications have to be detected and filtered out. The metadata names must be transformed in alphanumeric strings, be

[1] The ENCODE (Encyclopedia of DNA Elements) is an international collaboration of research groups funded by the National Human Genome Research Institute (NHGRI) pursuing the goal of building a comprehensive parts list of functional elements in the human genome, including elements that act at the protein and RNA levels, and regulatory elements that control cells and circumstances in which a gene is active [10].

[2] The Cancer Genome Atlas (TCGA) is a collaboration between the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) that has generated comprehensive, multi-dimensional maps of the key genomic changes in 33 types of cancer. The TCGA dataset, comprising more than two petabytes of genomic data, has been made publically available, and this genomic information helps the cancer research community to improve the prevention, diagnosis, and treatment of cancer [11].

[3] NGS technology produces raw data, i.e., short reads of DNA or RNA, that are then processed in a variety of ways. First, with "primary analysis", reads must be produced; then with "secondary analysis data must be aligned to the reference genome, and "features" (i.e., properties of DNA or RNA regions) have to be "called" (i.e., computed by programs); finally, "tertiary analysis" occurs, focused on mastering and understanding processed data.

human readable and meaningful. The last phase is about the actual implementation of the modules. The new datasets identified in the phase two are actually created by retrieving the selected data and metadata in applying the previously defined transformations. Some new files are generated starting from these provided by the source. During the implementation phase, along with the two modules, also other functionalities for GMQL-Importer are developed from scratch or by rewriting some already implemented (or partially implemented) functionalities; various code improvements are done too.

The structure of the thesis follows the phases of the project. In Chapter 2 we introduce some basic concepts required to understand what is done in this project thesis and its background context. The definitions of GDM and GMQL are given along with a description of their interactions. It follows a brief introduction about REP, describing how it is organized, its goals, motivations, and relevance. In Chapter 3 we describe the state of the art of GMQL-Importer. In this chapter all the most relevant functionalities are presented and the state of development before this project thesis started. In Chapter 4 we provide an in-depth analysis of the REP source listing where the data and metadata are located, how they are organized and their formats. In Chapter 5 we describe the process of datasets selection and design of the transformations required to integrate the data and metadata in a GDM repository. We list the relevant data grouped by semantic type, and for each group we describe the data and metadata before and after the transformation process. Chapter 6 is where we describe the implemented modules, focussing on the most relevant methods, the interaction among methods inside modules and between modules and the main body of the program, and the reason behind the design choices. In Chapter 7 we make analogous observations to that made in Chapter 6, but for methods implemented outside of the RoadmapImporter modules. In fact, we developed new functions for the main body of GMQL-Importer (executable also with modules different from RoadmapImporter) and performed various code improvements. Chapter 8 explains the results obtained by means of some statistics. We also describe the tests performed. The Chapter 9 is dedicated to some final considerations about the project and the objectives achieved. In Chapter 10, the final chapter, we discuss some future extensions of the project.

Chapter 2

Background information

In order to fully understand this thesis, the concepts of GDM and GMQL must be clear. The first section of this chapter briefly presents their core concepts, focusing on the aspects more relevant for this project. In the second section, instead, a general description on what the REP is, the motivations behind it, the goals pursued and why the REP data and metadata are relevant for the scientific community are presented.

2.1 GDM and GMQL

The GDM is an abstraction used to homogeneously describe semantically heterogeneous genomic (and epigenomic) data and metadata. GDM represents genomic information as datasets containing multiple samples. Each sample is described by means of two fundamental abstractions:

1. *genomic regions*, representing a portion of a reference epigenome identified by their genomic coordinates plus a region id (id and coordinates are the five region mandatory attribute). The region coordinates are qualified by the quadruples:

   $$< chr, left, right, strand >,$$

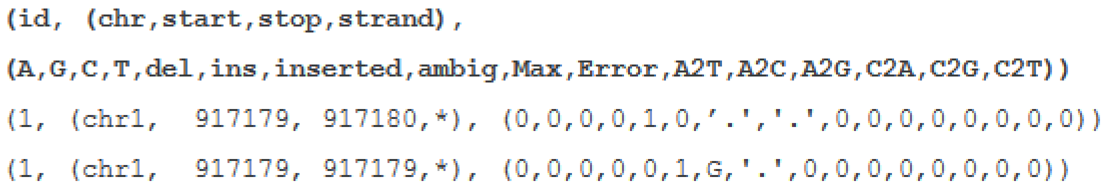
   where the *chr* element is the name of the chromosome, $left$ identifies the starting base pair, *right* identifies the ending base pairs, and *strand* represents the DNA strand with a "+" for the positive strand, a "-" if negative, and a "*" if missing. Coordinates are represented in 0-based notation. Regions are also provided with some optional attributes storing different information regarding the region in relation of a particular sample (e.g., region length, statistical significance). The attributes associated to a region are described by a XML schema file associated to a dataset.

2. *metadata*, storing all the knowledge about the particular sample. They are files containing list of attributes represented as name-value pairs and they are used to keep trace of experimental, biological and clinical information associated with each genomic data sample.

Formally, in the GDM a sample s is defined as a triple:

$$s =< id, \{< r_i, v_i >\}, \{m_j\} >,$$

where id is the sample identifier, $\{< r_i, v_i >\}$ are the regions of the sample, r_i and v_i the mandatory and optional attribute of region i respectively, and $\{m_j\}$ the list of metadata associated to the sample s. In Figure 2.1 is shown an example of GDM schema and a corresponding instance of the schema. While the regions inside a dataset are homogeneous w.r.t the dataset schema, the metadata can differ from sample to sample, even in the same dataset due to the great heterogeneity of the metadata. In this way the model allows data of different type, coming from different datasets and sources to interoperate without loosing flexibility and maintaining the capacity of fully represents them.



```
(id, (chr,start,stop,strand),
(A,G,C,T,del,ins,inserted,ambig,Max,Error,A2T,A2C,A2G,C2A,C2G,C2T))
(1, (chr1,  917179, 917180,*), (0,0,0,0,1,0,'.','.',0,0,0,0,0,0,0,0))
(1, (chr1,  917179, 917179,*), (0,0,0,0,0,1,G,'.',0,0,0,0,0,0,0,0))
```

Figure 2.1: Schema of a DNA-seq GDM schema representing a mutations and two possible instances of that schema (from [12]).

GMQL is a high-level query language providing a set of traditional database-like operation (i.e., select, project, group, order, difference, merge, cover, join, map) extended to deal with gnomic region. GMQL operation are base on the concept of genomic distance between regions calculated using the coordinates required in the GDM. Formally, the structure of all the GMQL operation can be described as:

$$< variable >=< operator > (< parameters >) < variable >,$$

where the $< varible >$ represents an input or output dataset depending if it is positioned before or after the "=" symbols. A GMQL operator can act on region, metadata or multiple datasets. GMQL is designed to provide to researchers without a strong computer science training an easy to learn and use platform to perform ternary analysis over the widest possible set of semantically heterogeneous data, with the objective of extract new knowledge. With simplicity of access in mind, GMQL can be used through various interfaces (i.e. web interface, rest API, RGMQL package, PyRGMQL), both locally or in a cloud environment. GMQL, in fact , has been designed to be scalable and performing in a big-data context, so it offers the best performance when running in a cloud environment to process thousand of experimental samples. Then GMQL allows to solve some of the most major issue regarding data integration and information retrieval from big volume of data:

- scalability, providing best performance with big dataset;
- portability, because it can be used in combination of many IT architecture, from personal computers to cloud computing infrastructure, and used through different interface as well;
- declaratives, since it allows to formulate high-level query allowing to focusing on complex biological problems instead of computation over single sample.

So, to sum up, the combined use of the GDM and the GMQL allows comprehensive integration and processing of multiple heterogeneous data, and supports the development of domain-specific data-driven computations and bio-molecular knowledge discovery [13–15].

2.2 The Roadmap Epigenomics Project

The REP [16] is a public resource of human epigenomic data created and maintained by *NIH*[1] *Roadmap Epigenomics Mapping Consortium* with the purpose to provide reference data for basic biology and disease-oriented research. The data are produced using NGS and *microarray* technologies to map DNA methylation[2], histone modifications[3], chromatin accessibility, and RNA transcript, including small RNA[4], in stem cells and primary ex vivo tissues selected to represent the normal counterparts of tissues and organ systems frequently involved in human diseases. The samples are selected to deliver a collection of normal epigenomes that provide a framework for future studies. A central role in this program is played by the *Reference Epigenome Mapping Centers* (REMCs), which systematically characterized the epigenomic landscapes of representative primary human tissues and cells. The row data produced by REMCs are subject to further analysis by the Epigenomics Data Analysis and Coordination Center (EDACC), which will provide data analysis and coordination for all of the Reference Epigenome Mapping Centers, as well as import all other Roadmap Epigenomics Program data generated outside of the mapping centers. The data published in the final release include data subjected to various analyses, from row data to secondary analysed data. All the project is coordinated by the NIH Roadmap Epigenomics Mapping Consortium with the primary objective to provide to reference data for the research community to reuse in further analysis and comparisons, but, along with the main goal, some secondary objectives are pursued by the Consortium such as:

1. develop standardized platforms, procedures, and reagents for epigenomics research;
2. conduct demonstration projects to evaluate how epigenomes change;
3. develop new technologies for single cell epigenomic analysis and in-vivo imaging of epigenetic activity;
4. develop software tools publicly accessible to be used in other contexts;
5. publish scientific papers to present the results obtained.

[1] The National Institutes of Health (NIH) is the primary agency of the United States government responsible for biomedical and public health research.

[2] DNA methylation is a heritable epigenetic mark involving the addition of methyl groups to the DNA molecule that can change the activity of a DNA segment without changing its sequence. It is essential for normal development and it is associated with a number of key processes and, when located in a gene promoter, it typically acts to repress gene transcription [17, 18].

[3] Histone modifications can impact gene expression by altering chromatin structure or recruiting histone modifiers; they act in diverse biological processes such as transcriptional activation/inactivation, chromosome packaging, and DNA damage/repair [19].

[4] Small RNAs are short (approximately 18 to 30 nucleotides), non-coding RNA molecules that can regulate gene expression through interactions with coding RNA [20].

The REP is motivated by the overall hypothesis that the origins of health and susceptibility to disease are, in part, the result of epigenetic regulation of the genetic blueprint. Specifically, epigenetic mechanisms that control stem cell differentiation and organogensis contribute to the biological response to endogenous and exogenous forms of stimuli that result in disease.

It is important to integrate REP data into a GDM repository to provide some reference data to use as comparison against pathological data during tertiary analysis performed using GMQL.

Chapter 3

State of the art: GMQL-Importer

GMQL-Importer [21] is an application developed by Pena [22]. GMQL-importer is a modular and user-configurable software tool designed to download data from any sources, elaborate them to match the desired data model and then load them to a remote target server. GMQL-Importer is regularly used by the GeCo team of Politecnico di Milano to integrate data coming from heterogeneous data sources (i.e., ENCODE, TCGA) and add them to a GDM repository, but GMQL-importer is designed to be as flexible as possible thanks to its modular structure, and it can be potentially used to operate with other data sources and data models, just adding new modules.

The execution of GMQL-importer is divided into three phases (i.e, download, transformation, load) that match the logical steps that the data must go through during any integration process. The files to be integrated are placed in a local repository[1] managed through a local database by GMQL-importer. The database keeps track of each file, their status, and their stage of elaboration. The files are organized by sources and datasets through a hierarchy of directories; to each dataset corresponds a directory containing downloaded and transformed data, each grouped into subdirectory. Each phase is carried out by a separate module that can be set by means of a configuration file (written in XML) editable by the user. Through the configuration file is possible to manipulate the program execution by setting a lot of parameters (local paths, sources, files to downloads, metadata renaming, etc.) such that it fits the user needs. Figure 3.1 reports a schematic representation of how GMQL-Importer and its components interact with source and target repository.

3.1 The downloading phase

During the downloading phases, the files containing the data to integrate and related metadata are downloaded from the sources, collected in datasets and saved in the local repository. Multiple download methods can exist depending on how the data are shared, so for each downloads methods is required to use different modules.

[1] As local repository we intend that a copy of the files to integrate is made on the device where GMQL-Importer is running. This imply that the hardware on which GMQL-Importer is executed must fulfil a storage memory capacity requirement defined by the size of the files to import. When estimating the storage capacity required to perform the integration, we must consider that the local repository will not only contains a copy of the original files downloaded from the source, but also a copy of the transformed files to upload in the target repository.

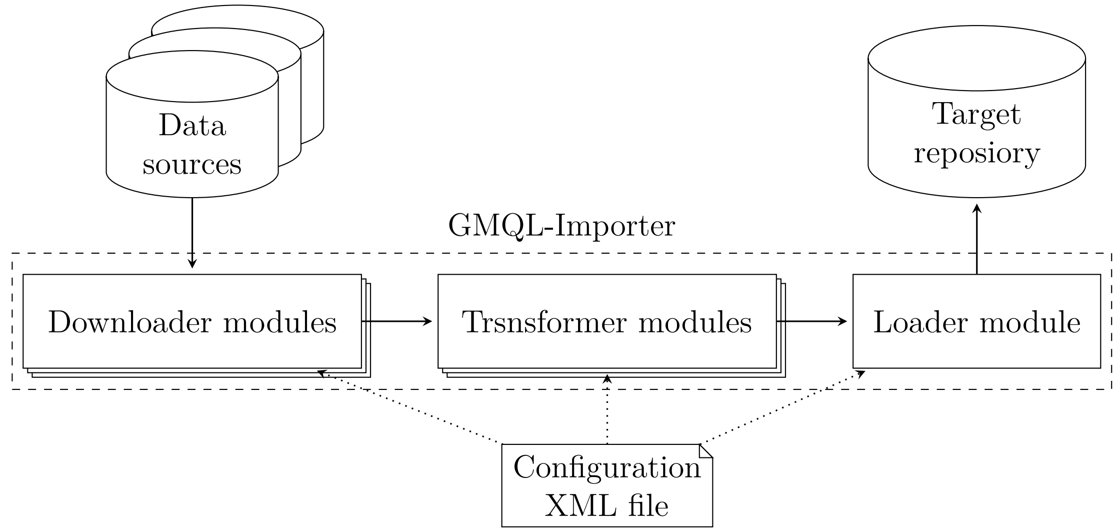


Figure 3.1: Flow of data between the logical components of GMQL-Importer and the other major participants involved in the integration process. The data are downloaded from a Source repository (e.g., REP, ENCODE, TCGA) by the downloader modules. The data are saved in a local directory and they are ready to be transformed by the transformer modules. Once the transformation is complete, the loader module load the data into the target repository (GDM repository). The user can customize the process through the configuration file by selecting the modules used and configuring them, as well as defining the source and the target repository and their propriety.

The download procedure depends on the source than the dataset itself, therefore different download methods can be used for multiple sources. The implementation of a downloading modules is done by extending the download process for a generic source. Any download processes has to:

1. check the datasets to download for the source from the configuration file;
2. mark the files belongings to the selected datasets already present in the local database as "compare";
3. scan the source searching all the files meeting the criteria of downloadability defined by user;
4. download the files found not present in the local database or modified since the last source scan and mark as "updated" if the download has been completed with success or "failed" otherwise;
5. mark all the remaining "compare" files as "outdated", since they have not been found in the last source scan, it means that they have been deleted from the source;
6. get from the local database all the "failed" files and try to download them again, so that not the whole downloading process has to be checked again.

The downloaded files are placed in the "downloads" local directory under their dataset directory.

3.2 The transformation phase

During the transformation phase both data and metadata must be modified to meet the requirements imposed by the reference data model. Until now, the only reference data model used during GMQL-Importer transformation phase is GDM, but it can be another one. An important step for the correct transformation is done at metadata level, where the metadata names have to be standardized by modifying them if needed. During the transformation, metadata coming from different sources could be combined and associated with the same sample. The important thing is that, at the end of the transformation each sample has his associated metadata file. Each metadata file must have the same name of the sample to which it refers plus mandatory extension ".meta", as it is required by GMQL. GMQL-Importer gives the general tools for the correct transformation of the region data and their metadata. First it ensures that the genomic data and their metadata are in a GDM compatible format, therefore it performs the needed modification of the original sources files until it is in GDM format. This operations could include full translation from raw data to GDM or extracting compressed containers, for example, but could potentially be any data modification or translation. These operations are strictly depends from the original format of the files and the data model used to integrate the data, so they are executed by dedicated modules, chosen by the user in the configuration files. Once the NGS data and metadata are GDM friendly, some more operations are carry out independently by the original format of the files or the data model, such as check if the region data fulfils the dataset schema and the metadata names revision and modification. This operations are executed outside of the transformation modules by the main program. The user can still choose if perform them or not in some cases.

As for the download phase, also the transformation modules are developed by extending a generalized process:

1. the files to be transformed are fetch from the database;
2. for each original files, the name of the files (none, one, or more than one) that will be generated at the end of the transformation process;
3. for each of the new file names obtained the transformation is executed;
4. the transformed files are added to the database as "updated";
5. the consistency of the genomic regions with the corresponding schema is checked;
6. the optional metadata attribute name modifications are applied.

Point 2, 3, and 4 of the process are performed inside of a transformation modules, instead points 1, 5 and 6 are executed outside the transformation modules. All the transformed file are placed inside the subdirectory "transformation" of the corresponding dataset directory.

3.3 The loading phase

To load data into a GDM repository, GMQL-importer takes advantage of a dedicated interface provided by GMQL. By giving the GMQL a user name, it is the interface to actually load a source datasets after that it has checked the consistency of the dataset w.r.t. the schema provided with it and that every region data file has its respective metadata file associated. Due to his internal policy, GMQL do not allow to add files to an already created dataset, and, if this happens, GMQLImporter ask to the user to choose if delete previous version stored in the remote directory, or if change the name of the updated dataset using the configuration XML file. During the uploading phase also a description of the file is added to each dataset.

3.4 The database

As already mentioned previously, all the files downloaded or transformed are stored in a local repository. This repository is managed through a local database by GMQL-Importer. The database keeps track of all the sources, datasets and files and all the other entity involved in a run of GMQL-Importer, such as log files and parameters associated with the program, sources, and datasets for a specific run. All the entity represented in the database are reported in Figure 3.2, along with their attributes and relations.

One of the most important function of the database is to store hash, last update date of upload in source, and size in the source of each file. In fact, this data saved in the database when a new file to download is found in the source or the first time and they are used to check if the files in the local repository are updated w.r.t. the files in the source. This way, files already downloaded in previous executions of the program are not downloaded again, unless a file has been modified in the source.

Another fundamental function is to keep track of the state of the files stored in the local repository. The state of the files is used both during the downloading an the transformation phases. The state of a file allows, for example, to know if a file is downloaded correctly or something went wrong and the download has failed, to know if a file has been delated from the sources and so must not be transformed, or to get the data ready to be transformed. The state and the transition among each state is represented in Figure 3.3. The stares of the files are:

- updated, when the file download/transformation is correctly stored in the local repository and its version is the same of that of the file in the source the last time thet a check is done.

- failed, when the file download/transformation failed and the file may not be valid, in this case the file is marked as failed;

- outdated, when the file was removed from the server, in this case the file exists locally but not in the source;

- compare, its a temporary status used used during the execution to know which files do not exist any more on the server, it is changed before the end of the execution.

The state transitions are performed using some database methods:

- `MarkToCompare`: by receiving a dataset, the database changes the status of every file in the dataset as to compare, this method is used to notice which files are no longer in the server side and have to be marked after as outdated in the local copy.

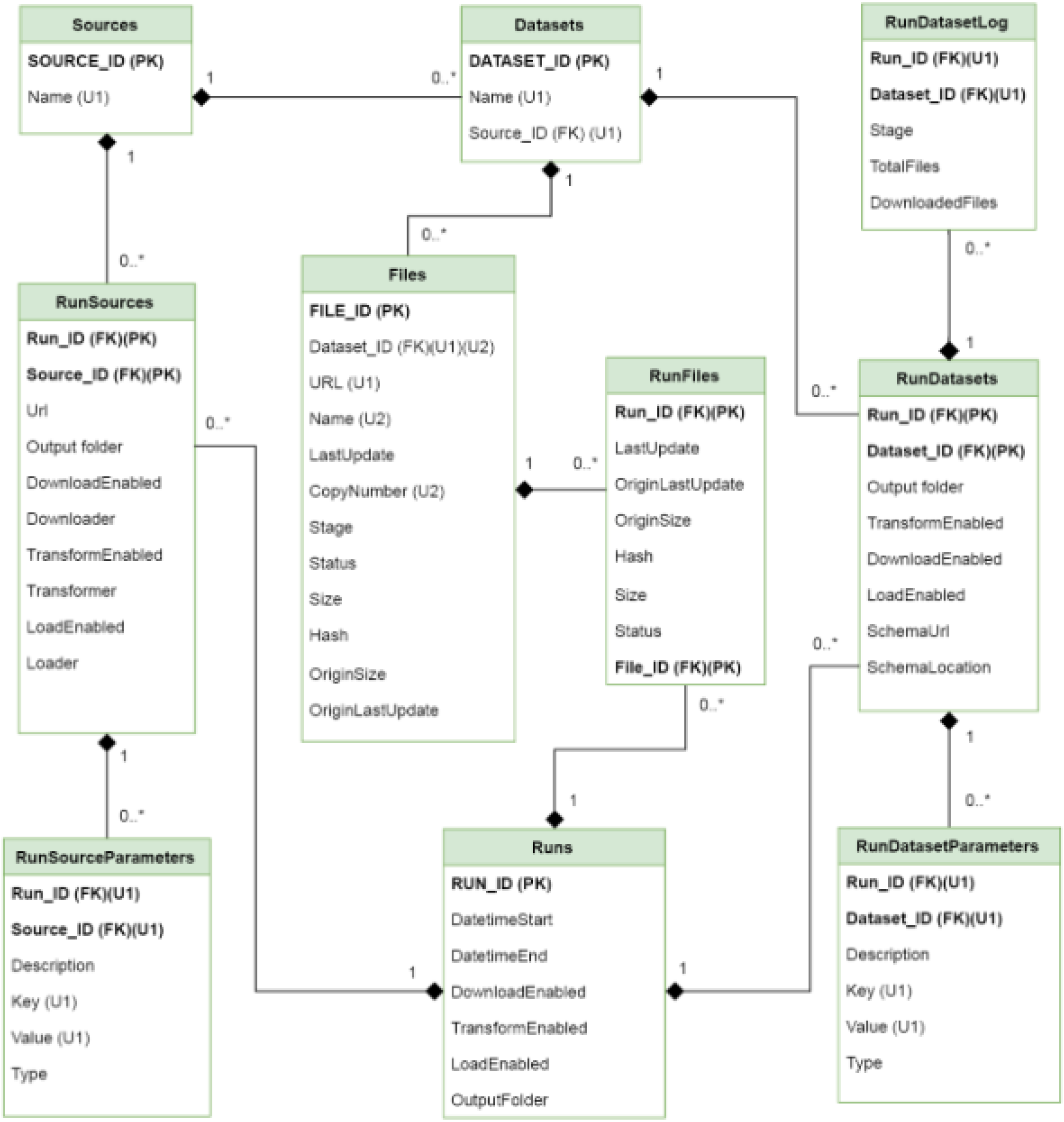


Figure 3.2: Entity-Relationship diagram of the GMQL-Importer database, representing all the entities, relationships, and attributes contained in the database (Source [22]).

- `MarkAsUpdated`: indicates the file was correctly downloaded or transformed and it is ready for the next step that could be transformation or loading.
- `MarkAsFailed`: when trying to download or transform a file, if the procedure fails, the file has to be excluded from further processing and thus it is marked as failed.
- `MarkAsOutdated`: once the whole dataset is downloaded or transformed this procedure allows finding the files that no longer exist in the remote server, and marks those local files as outdated. These files are no longer used in transformation or loading procedures.

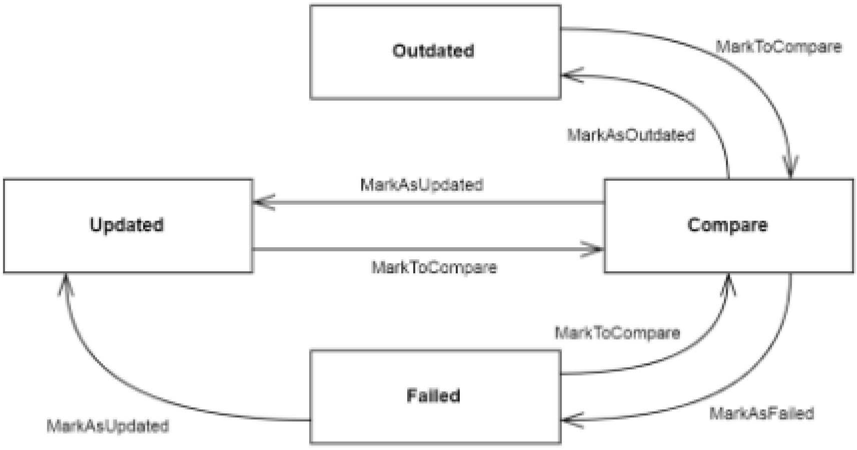
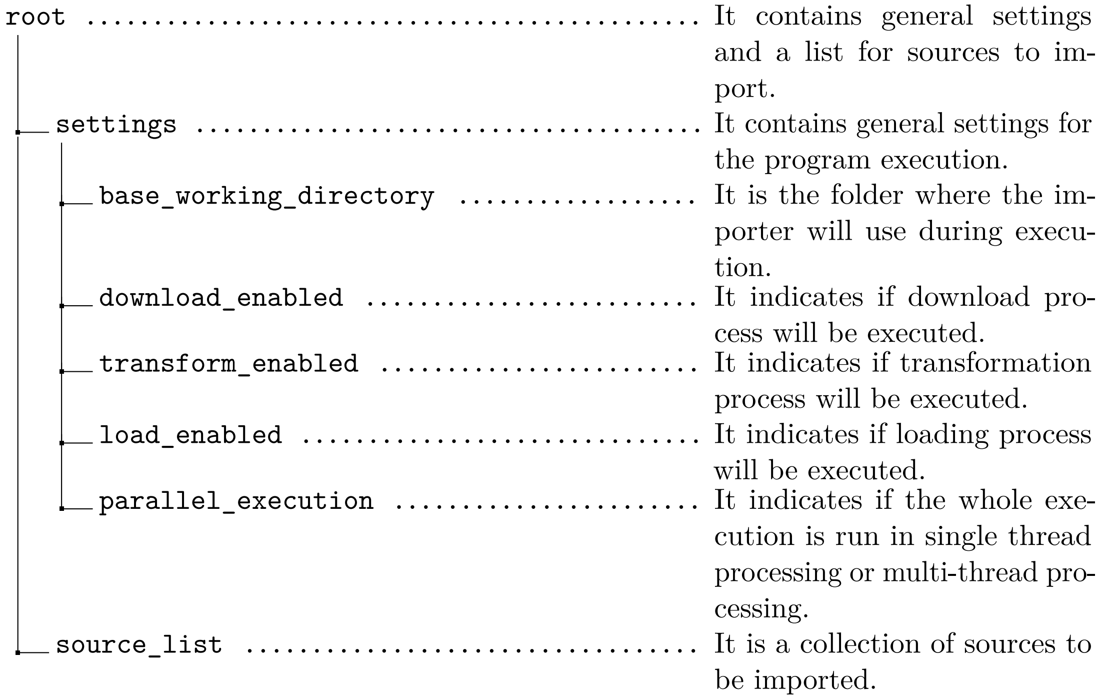


Figure 3.3: State diagram representing the states in which a file stored in a local repository of GMQL-Importer can be, and the state transitions among them (from [22]).

3.5 The configuration file

GMQL-Importer is designed to receive a configuration XML file with the needed parameters to perform the downloading, transforming and loading steps of the datasets, to provide a general approach for the formal creation of this file. Thanks to this configuration file, the user can customize the execution of GMQL-Importer. Any configuration XML file given as input of GMQL-Importer is validate using an XSD schema file. In this way, we are shure that the structure of the configuration file is what GMQL-Importer expects. The schema comprehends a root node where general settings and a source list are stored. Sources, as seen before, represent NGS data providers which provide those genomic data and experimental metadata divided in datasets, each source contains a list of datasets, each dataset after processing, represents a GDM dataset where every sample has a region data file and a metadata and every sample share the same region data schema. The configuration file is organized in a tree structure (schematically represented in Figures 3.4, 3.5, 3.6, 3.7) that match the hierarchical structure of a source. Each of the three main entities in the configuration file (i.e., root, sources, datasets) are associated with the parameters used by the user to configure the corresponding program entity and, so, the result of the execution.



- `root` — It contains general settings and a list for sources to import.
  - `settings` — It contains general settings for the program execution.
    - `base_working_directory` — It is the folder where the importer will use during execution.
    - `download_enabled` — It indicates if download process will be executed.
    - `transform_enabled` — It indicates if transformation process will be executed.
    - `load_enabled` — It indicates if loading process will be executed.
    - `parallel_execution` — It indicates if the whole execution is run in single thread processing or multi-thread processing.
  - `source_list` — It is a collection of sources to be imported.

Figure 3.4: Structure and description of the *root* node of the configuration file. Sub-node *source_list* contains one or more *source* nodes.

- `source_list` .... It is a collection of sources to be imported.
  - `source` .... It represents an NGS databank, contains basic information for downloading, transformingand loading process.
    - `name` .... It identification for the source.
    - `url` .... It address of the source.
    - `source_working_directory` .... It directory where the source's files will be processed.
    - `downloader` .... It indicates the downloading process to be performed to down-load the samples from this source.
    - `transformer` .... It indicates the transformation process to be performed to change the source samples into GDM compatible files for interoperability.
    - `loader` .... It indicates the responsible for loading the processed data into a GDM repository.
    - `download_enabled` .... It indicates if this source is going to be downloaded from the source.
    - `transform_enabled` .... It indicates if transformation process is executed for this source.
    - `load_enabled` .... It indicates if loading into GDM repository is executed for this source.
    - `parameter_list` .... It is a collection of parameters for downloading or loading the source.
  - `dataset_list` .... It is a collection of datasets to import from the source.

Figure 3.5: Structure and description of the *source* node of the configuration file. Sub-node *source_list* contains one or more *source* nodes and sub-node *parameter_list* contains one or more *parameter* nodes

- `dataset` .... It represents a set of samples that share the same region data schema and the same types of experimental or clinical metadata.
  - `name` .... It is an identifier for the dataset.
  - `dataset_working_directory` .... It is the subfolder where the download and transformation of this dataset is performed.
  - `schema_url` .... It is the address where the schema file can be found.
  - `schema_location` .... It indicates whether the schema is located in FTP, HTTP or LOCAL destination.
  - `download_enabled` .... It indicates if the download process will be performed for this dataset.
  - `transform_enabled` .... It indicates if the transformation process will be performed for this dataset.
  - `load_enabled` .... It indicates if the loading process will be executed for this dataset.
  - `parameter_list` .... It is a list of dataset specific parameters for downloading, transforming or loading this dataset.

Figure 3.6: Structure and description of the *dataset* node of the configuration file. Sub-node *parameter_list* contains one or more *parameter* nodes.

- `parameter` .... It defines specific information for a source or a dataset, this information isuseful for downloading, transforming or loading procedures.
  - `key` .... It is the name for the parameter, its identifier.
  - `value` .... It is the parameter information.
  - `description` .... It explains what the parameter is used for.
  - `type` .... It is the optional tag for the parameter.

Figure 3.7: Structure and description of the *parameter* node of the configuration file.

Chapter 4

Source analysis

To date (April 2018, Realise 9[1]), the REP repository contains a total of 2,827 genome-wide datasets[2], including 1,808 histone modification datasets, 371 DNase datasets, 304 DNA methylation datasets, and 209 RNA-Seq datasets. The remaining 135 datasets come from array-based experiments used to detect polymorphism and expression profiling of exons and genes. All the sequenced datasets are obtained by mapping sequencing reads onto hg19 assembly of the human genome using Pash 3.0 read mapper[3]. This raw data are produced by the REMCs using a diversity of assays, including chromatin immunoprecipitation[4] (ChIP), DNA digestion by DNase I[5] (DNase), bisulfite treatment[6], methylated DNA immunoprecipitation (MeDIP), methylation-sensitive restriction enzyme digestion (MRE), and RNA profiling, each followed by massively parallel short-read sequencing (-seq). In Table 4.2 is reported an exhaustive list of the experiments performed, along with the number of datasets produced.

Since the objective of the project is to produce reference epigenomes data to use in future experiments, the data come from human tissues without any diseases that are frequently involved in pathology development. The resulting datasets were assembled into publicly accessible websites and databases [27–30], which serve as a broadly useful resource for the scientific and biomedical community. Table A.1 in Appendix A reports all the 411 cell lines or tissues considered in REP, the experiments performed for each of them and the number of datasets produced.

Release 9 is focused on a subset of 1,936 datasets grouped into 111 *reference*

[1] REP data are provided in gradual release over the years with an irregular frequency starting from 2010 until 1014.

[2] In REP terminology a dataset are the data produced by a particular experiment on a specific sample, contrary to the notion of dataset in Genomic Data Model (GDM) where a dataset contains data with a homogeneous data type, but coming from multiple samples.

[3] Pash 3.0 is a software package able to perform sequence comparison and read mapping and can be employed as a module within analysis pipelines [23].

[4] Chromatin immunoprecipitation is an experimental technique used to investigate the interaction between proteins and DNA in the cell, to determine whether specific proteins are associated with specific genomic regions and to locate histone modifications, promoters or other DNA binding sites [24].

[5] DNase I is a versatile enzyme used in a range of molecular biology applications for DNA manipulations, including identification of protein binding sequences on DNA (DNase I footprinting) [25].

[6] Bisulfite treatment leaves methylated region unaffected, so it is used to detect DNA methylation pattern [26].

Table 4.1: List of all the experiments performed in the context of REP and the number of datasets produced by each experiment type.

Experiment	**N. of datasets**
Bisulfite-Seq	108
ChIP-Seq input	268
Digital genomic footprinting	21
DNase hypersensitivity	350
Exon array	99
Expression array	17
Genotyping array	19
H2A.Z	9
H2AK5ac	14
H2AK9ac	2
H2BK120ac	13
H2BK12ac	12
H2BK15ac	12
H2BK20ac	5
H2BK5ac	14
H3K14ac	11
H3K18ac	14
H3K23ac	13
H3K23me2	4
H3K27ac	156
H3K27me3	205
H3K36me3	223
H3K4ac	13
H3K4me1	218
H3K4me2	16
H3K4me3	209
H3K56ac	6
H3K79me1	16
H3K79me2	9
H3K9ac	90
H3K9me1	2
H3K9me3	217
H3T11ph	1
H4K12ac	2
H4K20me1	5
H4K5ac	6
H4K8ac	14
H4K91ac	9
MeDIP-Seq	45
MRE-Seq	45
mRNA-Seq	166
RRBS	106
smRNA-Seq	43
Total	**2827**

epigenomes that have been selected to perform further studies. An epigenome is a set of different types of data coming from different samples that characterize a cell line. An epigenome, to be selected as a reference epigenome, must be characterized by a complete set of five core histone marks:

- H3K4me3, associated with promoter regions;
- H3K4me1, associated with enhancer regions;
- H3K36me3, associated with transcribed regions;
- H3K27me3, associated with Polycomb repression[7];
- H3K9me3, associated with heterochromatin regions[8].

Selected epigenomes can also contain a subset of additional epigenomic marks, including:

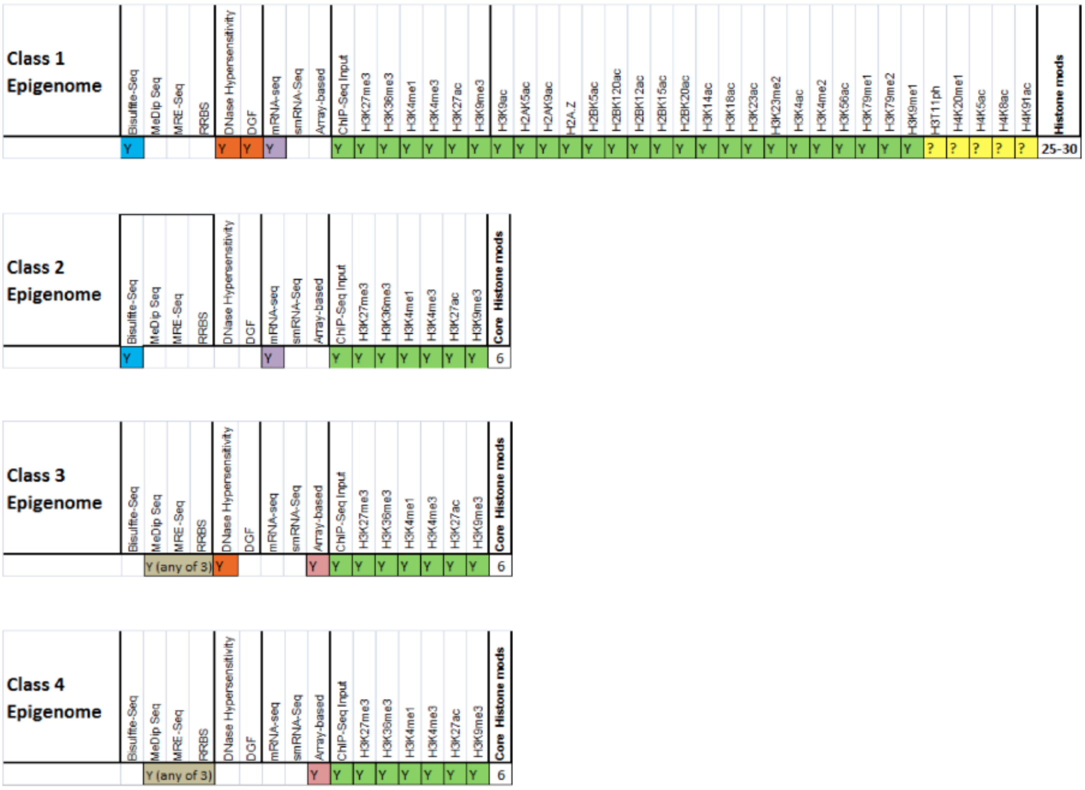
- acetylation marks H3K27ac and H3K9ac, associated with increased activation of enhancer and promoter regions;
- DNase hypersensitivity, denoting regions of accessible chromatin commonly associated with regulator bindings;
- DNA methylation, typically associated with repressed regulatory regions;
- RNA expression levels, measured using RNA-seq and gene expression microarrays.

Lastly, an additional 16 histone modification marks on average were profiled across 7 deeply covered cell types. Based on the assays used to profile them, epigenomes were grouped in 5 specific classes. Epigenomes from class 1 to class 4 are referred as *complete epigenomes* [33], and they were mapped to different optional data types (Figure 4.1). Class 5 epigenomes were subjected to ChIP-seq assays for the minimum set of five core histone modifications. A list of the consolidated epigenomes and their class is reported in Table 4.3.

The reference epigenomes contain datasets corresponding to each of the epigenomic data types coming from 183 biological samples. In other words, a reference epigenome is a cell line with a well defined and formalized epigenomic data associated and mapped along its genome. Data types can be seen as dimensions of the epigenome. Since there often exist multiple samples (technical and biological replicates from multiple individuals and/or datasets from multiple centers) from a particular unique cell type or tissue, the 111 selected reference epigenomes are referred as *unconsolidated epigenomes.* Data associated with the same unconsolidated epigenome, but coming from different samples, can be redundant or inconsistent. So, in order to reduce redundancy, improve data quality and achieve uniformity

[7] Polycomb proteins form chromatin-modifying complexes that implement transcriptional silencing in higher eukaryotes [31].

[8] Heterochromatin is a tightly packed form of DNA that regulates the expression of genes through RNA interaction [32].



Class 1 Epigenome	Bisulfite-Seq	MeDip Seq	MRE-Seq	RRBS	DNase Hypersensitivity	DGF	mRNA-seq	smRNA-Seq	Array-based	ChIP-Seq Input	H3K27me3	H3K36me3	H3K4me1	H3K4me3	H3K27ac	H3K9me3	H3K9ac	H2AK5ac	H2AK9ac	H2A.Z	H2BK5ac	H2BK120ac	H2BK12ac	H2BK15ac	H2BK20ac	H3K14ac	H3K18ac	H3K23ac	H3K23me2	H3K4ac	H3K4me2	H3K56ac	H3K79me1	H3K79me2	H3K9me1	H3T11ph	H4K20me1	H4K5ac	H4K8ac	H4K91ac	Histone mods
	Y				Y	Y	Y			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	?	?	?	?	?	25-30

Class 2 Epigenome	Bisulfite-Seq	MeDip Seq	MRE-Seq	RRBS	DNase Hypersensitivity	DGF	mRNA-seq	smRNA-Seq	Array-based	ChIP-Seq Input	H3K27me3	H3K36me3	H3K4me1	H3K4me3	H3K27ac	H3K9me3	Core Histone mods
	Y						Y			Y	Y	Y	Y	Y	Y	Y	6

Class 3 Epigenome	Bisulfite-Seq	MeDip Seq	MRE-Seq	RRBS	DNase Hypersensitivity	DGF	mRNA-seq	smRNA-Seq	Array-based	ChIP-Seq Input	H3K27me3	H3K36me3	H3K4me1	H3K4me3	H3K27ac	H3K9me3	Core Histone mods
		Y (any of 3)			Y				Y	Y	Y	Y	Y	Y	Y	Y	6

Class 4 Epigenome	Bisulfite-Seq	MeDip Seq	MRE-Seq	RRBS	DNase Hypersensitivity	DGF	mRNA-seq	smRNA-Seq	Array-based	ChIP-Seq Input	H3K27me3	H3K36me3	H3K4me1	H3K4me3	H3K27ac	H3K9me3	Core Histone mods
		Y (any of 3)							Y	Y	Y	Y	Y	Y	Y	Y	6

Figure 4.1: Criterion for complete epigenome classification (from [33]).

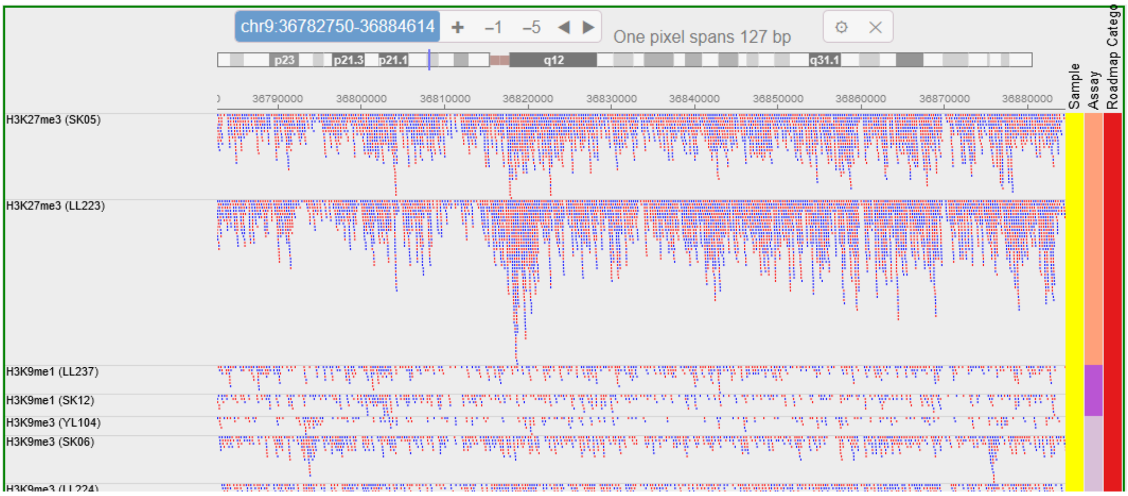


Figure 4.2: The first tracks of the "IMR90 Cell Line" (fetal lung fibroblasts) unconsolidated epigenome. There are more tracks for each data type due to the presence of multiple samples per data type. Each track corresponds to a sample/experiment dataset. Unconsolidated epigenomes are not associated with EIDs or Standardized Epigenome name (generated from [34]).

required for integrative analyses[9], unconsolidated epigenomes data were subjected to additional processing to obtain comprehensive data for the 111 selected cell lines. The processed epigenomes are referred as *consolidated epigenomes*. For each reference epigenome an unconsolidated version and a consolidated version, obtained by processing the unconsolidated one, is provided. In Figures 4.2 and 4.3 an example of

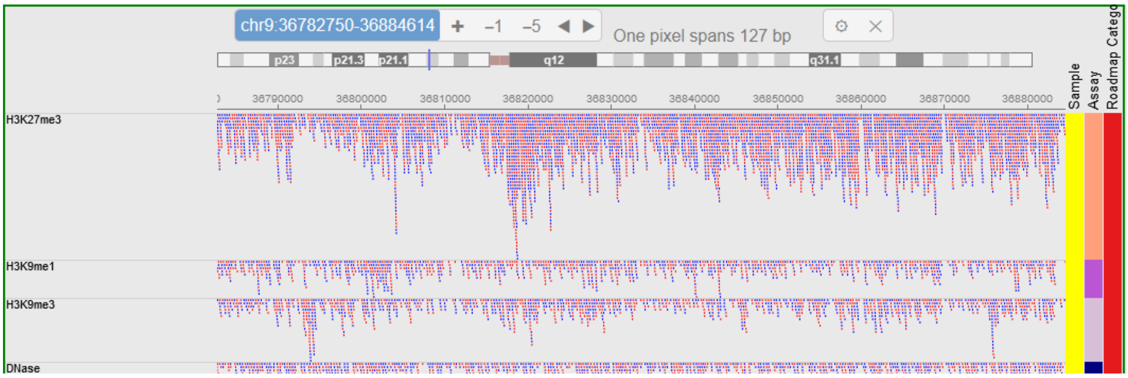


Figure 4.3: The first tracks of the "E017 fetal lung fibroblasts IMR90 Cell Line" consolidated epigenome. There is a single track for each data type and no reference to the samples is made. Each track corresponds to a comprehensive processed set of data coming from the same experiment, but different samples (generated from [34]).

epigenome (unconsolidated and consolidated, respectively) is represented graphically. As it can be seen, in the unconsolidated one different samples exist for the same data type (sample identifier is indicated in parenthesis after data type), but in the consolidated version of the same cell line the data type does not have a sample associated, and it is represented by a single track instead.

Datasets corresponding to 16 epigenomes from *The Encyclopedia of DNA Elements (ENCODE) project* [10] were also processed similarly and used in the integrative analyses, obtaining a total of 127 consolidated epigenomes. The ENCODE epigenomes are listed in Table 4.3 and marked by "ENCODE" as value of the "Class" attribute.

Numeric epigenome identifiers EIDs (e.g. E001) and mnemonics for epigenome names were assigned for each of the consolidated epigenomes; ENCODE epigenome IDs range from E114 to E129. Key metadata such as age, sex, anatomy, epigenome class, ethnicity and solid/liquid status were summarized for the consolidated epigenomes. All data sets from the 127 consolidated epigenomes were subjected to processing filters to ensure uniformity in terms of read-length-based mappability and sequencing depth. The data produced by the integrative analysis are available at the *Supplementary website for the 2015 Consortium paper* [34].

In addition to the 1,936 datasets analysed across 111 reference epigenomes, the NIH Roadmap Epigenomics Project has generated an additional 891 genome-wide datasets, linked from Gene Expression Omnibus (GEO) database [30] and the Human Epigenome Atlas [36], and also publicly and freely available.

[9] All the integrative analysis are described in details in the article "Integrative analysis of 111 reference human epigenomes" [35], along with all the data processing performed to obtain consolidated epigenomes required to perform the analysis. See also Section 4.1.4 and Section 4.2.1

Table 4.3: List of the epigenomes provided by REP, release 9. Each epigenome is characterised by a unique EID (epigenome identifier), a standard name that identifies the cell line, and a class specifying the data associated and the project that produced them (i.e., REP, ENCODE). E060 and E064 were rejected due to data quality issues, so they were excluded from the realise.

EID	Standardized epigenome name	Class
E001	ES-I3 Cells	Class 4
E002	ES-WA7 Cells	Class 4
E003	H1 Cells	Class 1
E004	H1 BMP4 Derived Mesendoderm Cultured Cells	Class 1
E005	H1 BMP4 Derived Trophoblast Cultured Cells	Class 1
E006	H1 Derived Mesenchymal Stem Cells	Class 1
E007	H1 Derived Neuronal Progenitor Cultured Cells	Class 1
E008	H9 Cells	Class 1
E009	H9 Derived Neuronal Progenitor Cultured Cells	Class 5
E010	H9 Derived Neuron Cultured Cells	Class 5
E011	hESC Derived CD184+ Endoderm Cultured Cells	Class 2
E012	hESC Derived CD56+ Ectoderm Cultured Cells	Class 2
E013	hESC Derived CD56+ Mesoderm Cultured Cells	Class 2
E014	HUES48 Cells	Class 5
E015	HUES6 Cells	Class 5
E016	HUES64 Cells	Class 2
E017	IMR90 fetal lung fibroblasts Cell Line	Class 1
E018	iPS-15b Cells	Class 5
E019	iPS-18 Cells	Class 4
E020	iPS-20b Cells	Class 4
E021	iPS DF 6.9 Cells	Class 2
E022	iPS DF 19.11 Cells	Class 2
E023	Mesenchymal Stem Cell Derived Adipocyte Cultured Cells	Class 5
E024	ES-UCSF4 Cells	Class 2
E025	Adipose Derived Mesenchymal Stem Cell Cultured Cells	Class 5
E026	Bone Marrow Derived Cultured Mesenchymal Stem Cells	Class 5
E027	Breast Myoepithelial Primary Cells	Class 2
E028	Breast variant Human Mammary Epithelial Cells (vHMEC)	Class 3
E029	Primary monocytes from peripheral blood	Class 5
E030	Primary neutrophils from peripheral blood	Class 5
E031	Primary B cells from cord blood	Class 4
E032	Primary B cells from peripheral blood	Class 5
E033	Primary T cells from cord blood	Class 3
E034	Primary T cells from peripheral blood	Class 5
E035	Primary hematopoietic stem cells	Class 4
E036	Primary hematopoietic stem cells short term culture	Class 5
E037	Primary T helper memory cells from peripheral blood 2	Class 4
E038	Primary T helper naive cells from peripheral blood	Class 4
E039	Primary T helper naive cells from peripheral blood	Class 5
E040	Primary T helper memory cells from peripheral blood 1	Class 5
E041	Primary T helper cells PMA-I stimulated	Class 5
E042	Primary T helper 17 cells PMA-I stimulated	Class 5

List of the epigenomes provided by REP, release 9. Each epigenome is characterised by a unique EID (epigenome identifier), a standard name that identifies the cell line, and a class specifying the data associated and the project that produced them (i.e., REP, ENCODE). E060 and E064 were rejected due to data quality issues, so they were excluded from the realise.

EID	Standardized epigenome name	Class
E043	Primary T helper cells from peripheral blood	Class 5
E044	Primary T regulatory cells from peripheral blood	Class 5
E045	Primary T cells effector/memory enriched from peripheral blood	Class 5
E046	Primary Natural Killer cells from peripheral blood	Class 5
E047	Primary T CD8+ naive cells from peripheral blood	Class 4
E048	Primary T CD8+ memory cells from peripheral blood	Class 5
E049	Mesenchymal Stem Cell Derived Chondrocyte Cultured Cells	Class 5
E050	Primary hematopoietic stem cells G-CSF-mobilized Female	Class 2
E051	Primary hematopoietic stem cells G-CSF-mobilized Male	Class 3
E052	Muscle Satellite Cultured Cells	Class 5
E053	Cortex derived primary cultured neurospheres	Class 4
E054	Ganglion Eminence derived primary cultured neurospheres	Class 2
E055	Foreskin Fibroblast Primary Cells skin01	Class 3
E056	Foreskin Fibroblast Primary Cells skin02	Class 3
E057	Foreskin Keratinocyte Primary Cells skin02	Class 2
E058	Foreskin Keratinocyte Primary Cells skin03	Class 2
E059	Foreskin Melanocyte Primary Cells skin01	Class 3
E061	Foreskin Melanocyte Primary Cells skin03	Class 4
E062	Primary mononuclear cells from peripheral blood	Class 4
E063	Adipose Nuclei	Class 5
E065	Aorta	Class 2
E066	Liver	Class 2
E067	Brain Angular Gyrus	Class 5
E068	Brain Anterior Caudate	Class 5
E069	Brain Cingulate Gyrus	Class 5
E070	Brain Germinal Matrix	Class 2
E071	Brain Hippocampus Middle	Class 2
E072	Brain Inferior Temporal Lobe	Class 5
E073	Brain_Dorsolateral_Prefrontal_Cortex	Class 5
E074	Brain Substantia Nigra	Class 5
E075	Colonic Mucosa	Class 4
E076	Colon Smooth Muscle	Class 5
E077	Duodenum Mucosa	Class 4
E078	Duodenum Smooth Muscle	Class 5
E079	Esophagus	Class 2
E080	Fetal Adrenal Gland	Class 5
E081	Fetal Brain Male	Class 3
E082	Fetal Brain Female	Class 4
E083	Fetal Heart	Class 3
E084	Fetal Intestine Large	Class 2
E085	Fetal Intestine Small	Class 2

List of the epigenomes provided by REP, release 9. Each epigenome is characterised by a unique EID (epigenome identifier), a standard name that identifies the cell line, and a class specifying the data associated and the project that produced them (i.e., REP, ENCODE). E060 and E064 were rejected due to data quality issues, so they were excluded from the realise.

EID	Standardized epigenome name	Class
E086	Fetal Kidney	Class 3
E087	Pancreatic Islets	Class 4
E088	Fetal Lung	Class 3
E089	Fetal Muscle Trunk	Class 5
E090	Fetal Muscle Leg	Class 5
E091	Placenta	Class 5
E092	Fetal Stomach	Class 5
E093	Fetal Thymus	Class 5
E094	Gastric	Class 2
E095	Left Ventricle	Class 2
E096	Lung	Class 2
E097	Ovary	Class 2
E098	Pancreas	Class 2
E099	Placenta Amnion	Class 5
E100	Psoas Muscle	Class 2
E101	Rectal Mucosa Donor 29	Class 4
E102	Rectal Mucosa Donor 31	Class 4
E103	Rectal Smooth Muscle	Class 5
E104	Right Atrium	Class 2
E105	Right Ventricle	Class 2
E106	Sigmoid Colon	Class 2
E107	Skeletal Muscle Male	Class 4
E108	Skeletal Muscle Female	Class 4
E109	Small Intestine	Class 2
E110	Stomach Mucosa	Class 5
E111	Stomach Smooth Muscle	Class 4
E112	Thymus	Class 2
E113	Spleen	Class 2
E114	A549 EtOH 0.02pct Lung Carcinoma Cell Line	ENCODE
E115	Dnd41 TCell Leukemia Cell Line	ENCODE
E116	GM12878 Lymphoblastoid Cells	ENCODE
E117	HeLa-S3 Cervical Carcinoma Cell Line	ENCODE
E118	HepG2 Hepatocellular Carcinoma Cell Line	ENCODE
E119	HMEC Mammary Epithelial Primary Cells	ENCODE
E120	HSMM Skeletal Muscle Myoblasts Cells	ENCODE
E121	HSMM cell derived Skeletal Muscle Myotubes Cells	ENCODE
E122	HUVEC Umbilical Vein Endothelial Primary Cells	ENCODE
E123	K562 Leukemia Cells	ENCODE
E124	Monocytes-CD14+ RO01746 Primary Cells	ENCODE
E125	NH-A Astrocytes Primary Cells	ENCODE
E126	NHDF-Ad Adult Dermal Fibroblast Primary Cells	ENCODE
E127	NHEK-Epidermal Keratinocyte Primary Cells	ENCODE

List of the epigenomes provided by REP, release 9. Each epigenome is characterised by a unique EID (epigenome identifier), a standard name that identifies the cell line, and a class specifying the data associated and the project that produced them (i.e., REP, ENCODE). E060 and E064 were rejected due to data quality issues, so they were excluded from the realise.

EID	**Standardized epigenome name**	**Class**
E128	NHLF Lung Fibroblast Primary Cells	ENCODE
E129	Osteoblast Primary Cells	ENCODE

4.1 Data sources

All genome-wide maps of histone modifications, DNA accessibility, DNA methylation and RNA expression contained in the Release 9 of the compendium are freely available online. Raw sequencing data deposited at the Sequence Read Archive[10] [38] or dbGAP[11] [39] are linked from:

- *NIH Roadmap Epigenomics - GEO - NCBI* [30] (HTTP)
- *NCBI - GEO FTP server for Roadmap Epigenomic data* [29] (FTP)

All primary processed data[12] (including mapped reads) for profiling experiments are contained within Release 9 of the Human Epigenome Atlas at:

- *Epigenome Atlas FTP server for Roadmap data* [28] (FTP)
- *BCM Roadmap Epigenomics EDACC*[13] [27] (HTTP)

Complete metadata associated with each dataset in this collection, are archived at GEO and describe samples, assays, data processing details and quality metrics collected for each profiling experiment (Section 4.2.2). Data generated by the integrative analysis described in the article "Integrative analysis of 111 reference human epigenomes" [35] are available at *Supplementary website for the 2015 Consortium paper* [34] (HTTP), as the metadata associated with consolidated epigenomes (Section 4.2.1). In the following subsections, data provided by each source and the way in which they are organized and described.

[10] The Sequence Read Archive (SRA, previously known as the Short Read Archive) is a bioinformatics database that provides a public repository for raw sequencing data and alignment information from high-throughput sequencing platforms with the aim of enhancing reproducibility and allow for new discoveries by comparing data sets [37].

[11] The database of Genotypes and Phenotypes (dbGaP) was developed to archive and distribute the data and results from studies that have investigated the interaction of genotype and phenotype in Humans.

[12] bed, wig and tab files are considered primary processed data by NIH Roadmap Epigenomics Mapping Consortium; while txt, bam and cel files are raw data. All those data are produced by REMCs and included in the Epigenome Atlas release to be used as starting point in further elaborations, so they must not be confused with the processed data produced in the context of the "Integrative analysis of 111 reference human epigenomes" [35] available exclusively through *Supplementary website for the 2015 Consortium paper* [34] (Section 4.1.4).

[13] BMC is the acronym of BioMed Central. EDACC stands for Epigenomics Data Analysis and Coordination Center, which provides data analysis and coordination for all of the Reference Epigenome Mapping Centers, as well as imports all other Roadmap Epigenomics Program data generated outside of the mapping centers [40].

4.1.1 NIH Roadmap Epigenomics - GEO - NCBI

This source contains links for raw sequencing data deposited at the Sequence Read Archive [38] or dbGAP [39]. On this web page there are available three different ways to get the data, that are accessible through tabs that organize the data by samples, by studies[14], and as double entry matrix where the cell lines are represented by rows and the experiments[15] are the columns. The Samples tab includes:

- A complete list of each sample and experiment.
- Possibility to display selected tracks on NCBI Sequence Viewer [41] or UCSC Genome Browser [42] using check boxes to select which tracks to view.
- Search feature to filter the sample table.
- A function that allows samples table exporting.
- Download links for individual sra, bam, cell, txt, and wig files.
- Links to the original GEO records which contain a complete description of the sample and experiment parameters.

The Studies tab includes:

- An overview of each study.
- Batch download links for sra, bam, cell, txt and wig files in a study.
- Links to original GEO and SRA records.
- Links to corresponding samples in the Samples tab.

The Matrix tab includes:

- A matrix that depicts samples (rows) and experiments (columns), and the number of assays available for each.
- Possibility to display selected tracks on NCBI Sequence Viewer [41] or UCSC Genome Browser [42] using check boxes to select which tracks to view. All boxes in row or column can be selected clicking the box next to the sample or experiment.
- Filter by Production Center option.
- Links to corresponding samples in the Samples tab (clicking the sample or experiment name, or the number in each cell).

[14] In GEO terminology, a "study" is the project that provides one or more related datasets (e.g., UCSD Human Reference Epigenome Mapping Project, BI Human Reference Epigenome Mapping Project: ChIP-Seq in human subject, BI Human Reference Epigenome Mapping Project: Characterization of DNA methylation by RRBS). Studies (or series) are identified inside GEO by GSExxx code.

[15] GEO uses the term "experiment" to identify a particular technology used to obtain a dataset (e.g., Bisulfite-Seq, MeDIP-Seq, Whole genome sequencing) or, in ChIP-Seq case, the feature used (e.g., H3K79me2).

All the download links of bam, txt and cell files redirect to GEO FTP server [29], see Section 4.1.2 for details. All the download links of sra files redirect[16] to SRA FTP server [43].

4.1.2 NCBI - GEO FTP server for Roadmap Epigenomic data

This source is a directory based repository accessible directly or through links provided by *NIH Roadmap Epigenomics - GEO - NCBI* [30] (Section 4.1.1) that organises data in two main directories:

- "by_experiment": contains a directory for each experiment and each experiment directory contains all the analysed cell lines; in each cell line directory, data files from one or more samples of that cell line are present.
- "by_sample": samples are first grouped by cell line and then by experiment.

Depending on the experiments carried out on a sample and the provider, different data file formats can be available:

- cel: data file created by Affymetrix DNA microarray image analysis software (exon array experiment) that stores the results of the intensity calculations on the pixel values from probes on an Affymetrix GeneChip [44, 45].
- bam: a compressed binary version of the Sequence Alignment/Map (SAM) format, a compact and index-able representation of nucleotide sequence alignments [46, 47].
- txt: text file containing Affymetrix experiment custom information and annotation [48, Guidelines for Creating a Text File section].
- wig: old file format for display of dense, continuous data such as GC percent, probability scores, and transcriptome data [49].
- tab: textual file containing data produced by smRNA-Seq experiment.
- bed: textual file that provides a flexible way to define the data lines that are displayed in an annotation track; a line represents a feature, each containing 3-12 columns (attributes) of data, plus optional track definition lines [50, 51].

All the files are inside a gzip (.gz) compress archive. File names follow the pattern [GSMCode]_[provider].[cellLine].[experiment].[sampleID].[fileExtension].gz (e.g. GSM433177_BI.H1.H3K4me1.Solexa-10529.wig.gz). SampleID is optional and can be in different formats; it allows to disambiguate samples obtained by technical or biological replicates.

The primary processed data are in bed, wig and tab format, depending on the cell line and the experiments performed; all the other formats are raw data. In Table 4.4 the number of datasets with certain file formats associated is reported.

[16] Most of the sra links to single files do not work properly and return a "550: No such file or directory" error.

Table 4.4: The number of datasets with certain file formats available.

File formats	**N. of dataset**
bed, wig	954
cel	21
txt	19
wig	214
bam, bed, wig	56
bed, wig	1425
bam, bed, tab, wig	1
bed, tab, wig	42
cel, txt	95
Total	**2827**

4.1.3 BCM Roadmap Epigenomics EDACC and Epigenome Atlas FTP server for Roadmap data

Both sources are available through Genboree [52], a web-based platform for multi-omics[17] research and data analysis that can be used to upload data and perform various analyses. They contain all primary processed data[18] (including mapped reads) for profiling experiments. The data are organized in the same exact way in both sources; the only difference is the protocol used to get them. Both sources are simple directory based repository in which primary processed sample data are accessible through three different main directories [53].

The data in "study-sample-experiment" are organized hierarchically by:

- data-producing center (study)
- cell line/primary cell/primary tissue (sample)
- epigenomic mark (experiment)

The data in "sample-experiment" are organized hierarchically by:

- cell line/primary cell/primary tissue (sample)
- epigenomic mark (experiment)

The data in "experiment-sample" are organized hierarchically by:

- epigenomic mark (experiment)
- cell line/primary cell/primary tissue (sample)

Two[19] different data file types can be associated with each sample:

[17] Multi-omics is a biological analysis approach where the data sets are a combination of genome, proteome, transcriptome, epigenome, and microbiome data.

[18] See Footnote 12.

[19] In the *DATA RELEASE POLICY* [53] of the *Epigenome Atlas FTP server for Roadmap data* [28] they list thee types of primary processed data. The cell file format is not present though, so probably they have been removed in the most recent realise versions.

- .bed: with coordinates of DNA regions obtained by mappings of Illumina reads to GRCh37/hg19. Available for all experiments except Bisulfite-Seq and Reduced Representation Bisulfite-Seq.
- .wig: with read density scores based on read mappings. For Bisulfite-Seq and Reduced Representation Bisulfite-Seq, the score is calculated as methylated calls/(methylated + unmethylated calls). For other experiments, the score is the raw number of reads mapping in 20 bp windows.

All files are inside a gzip (.gz) compress archive. The name of a file follows the pattern [provider].[cellLine].[experiment].[sampleID].[fileExtension].gz (e.g. UCSD.H1_Derived_Mesenchymal_Stem_Cells.H2BK5ac.SK517.bed.gz). SampleID is optional and can be in different formats; it allows to disambiguate samples obtained by technical or biological replicates.

4.1.4 Supplementary website for the 2015 Consortium paper

This web portal serves as a supplementary data repository accompanying consortium paper “Integrative analysis of 111 reference human epigenomes” [35], and it provides uniformly processed datasets, integrative analysis products, and interactive genome browser sessions resulting from a joint analysis of 111 consolidated epigenomes from the Roadmap Epigenomics Project and 16 epigenomes from ENCODE project. It is divided into different sections and subsections that correspond to the various analyses performed starting from raw data, and the different types of data produced. Below a simplified list of the available data is reported.

1. Primary processed data and raw data: links to *BCM Roadmap Epigenomics EDACC* [27] and *NIH Roadmap Epigenomics - GEO - NCBI* [30] are reported.
2. ChIP-seq and DNase-seq uniformly processed data for consolidated epigenomes:
   (a) Mapped reads: links to TagAlign[20] files containing:
      i. unfiltered raw primary alignment files ((Unconsolidated Epigenomes, non-uniform mappability));
      ii. 36 bp mappability filtered primary alignment files (Unconsolidated Epigenomes, uniform mappability[21]);
      iii. 36 bp mappability filtered, pooled and subsampled read alignment files (consolidated epigenomes).
   (b) Peak regions: links to narrow contiguous regions of enrichment (peaks) for histone ChIP-seq and DNase-seq and broad domains (NarrowPeak [55] file format) of enrichment for histone ChIP-seq and DNase-seq (BroadPeak [56] and GappedPeak [57] file format); both for consolidated and unconsolidated epigenomes.

[20] Tag Alignment is a legacy bed-based format used in hg18 genomic mapping of short sequence tags [54].

[21] Only unique mapping reads were retained and duplicates were filtered out.

   (c) Genome-wide signal coverage tracks: links to bigwig files containing -log10(p-value) signal tracks and fold-enrichment signal tracks of consolidated and unconsolidated epigenomes.

   Each consolidated epigenome has a .tagAlign, a .narrowPeak, and a .bigwig file for each genome feature associated.

3. RNA-seq uniform processed data for consolidated epigenomes:
   (a) expression quantification data in tab-delimited matrix file;
   (b) bigWig file containing RNA-seq signal tracks;
   (c) RNA-seq intergenic contigs.

4. Methylation data for consolidated epigenomes:
   (a) fractional methylation and read coverage in bigWig format for whole-genome bisulphite methylation calls;
   (b) fractional methylation and read coverage in bigWig format for RRBS[22] methylation calls;
   (c) fractional methylation in bigWig format generated by combination of MeDIP[23] and MRE[24] techniques (mCRF[25]).

5. Differentially Methylated Regions (DMRs) calls across reference epigenomes: whole-genome bisulphite sequencing (WGBS) DMRs and reduced representation bisulfite sequencing (RRBS) DMRs matrix and bigWig file.

6. DMRs in Human Embryonic Stem Cells (hESC): DMRs defined across human ESCs and three derived cell types are presented in a xlsx file.

7. Additional DMR calls: a xlsx file containing DMRs defined for studying breast epithelia differentiation.

8. Imputed signal tracks[26] for histone ChIP-seq, DNase-seq, RNA-seq, and methylation: each directory corresponds to a marker and contains a set of bigWig files, one for each reference epigenome; the signal tracks for the Histone Modifications and DNase are based on the p-value of the predicted target mark position.

[22] Reduced representation bisulfite sequencing (RRBS) is an efficient and high-throughput technique used to analyse the genome-wide methylation profiles on a single nucleotide level [58].

[23] Methylated DNA immunoprecipitation (MeDIP) is a versatile, large-scale (chromosome- or genome-wide) immunocapturing approach for unbiased detection of methylated DNA [59].

[24] Methylation-sensitive Restriction Enzyme is used to generate DNA fragments required to perform further epigenetic analysis exploiting the fact that these enzymes leave the methylated DNA intact [60].

[25] methylCRF is an algorithm that integrates MeDIP and MRE sequencing data to predict DNA methylation levels at single CpG resolution genome wide [61]

[26] For imputing epigenomic datasets ChromImpute method is used. ChromImpute predicts a target mark in a target reference epigenome, combining information about other marks mapped in the target reference epigenome and the target mark at the same position in similar reference epigenomes. To make predictions about unobserved datasets, ChromImpute implements an ensemble of regression trees. Additionally, an imputed version of each observed dataset was generated without using the corresponding observed data.

9. Peak regions on imputed data: narrow contiguous regions of enrichment (peaks) for histone ChIP-seq and DNase-seq and broad domains of enrichment for histone ChIP-seq in bed format.

10. Chromatin state model based on imputed data.

11. Chromatin state learning data based on:

    (a) Core 15-state model: a ChromHMM[27] model applicable to all 127 epigenomes was learned by virtually concatenating consolidated data corresponding to the core set of 5 chromatin marks assayed in all epigenomes; the model was trained on 60 epigenomes with highest-quality data.

    (b) Expanded 18-state model: a second "expanded" model applicable to 98 epigenomes that also have an H3K27ac ChIP-seq dataset, was learned by virtually concatenating consolidated data corresponding to the core set of 5 chromatin marks and H3K27ac. The model was trained on 40 high quality epigenomes.

    (c) Expanded 50 chromatin state models using large numbers of histone marks for Class 1 epigenomes: for each of the seven deeply-profiled reference epigenomes, chromatin states were independently learned on observed data for all available histone modifications or variants, and DNase in the reference epigenome; the focus is put on a larger set of 50-states to capture the additional state distinctions afforded by using additional marks.

12. Common lineages and properties of epigenomes (obtained by clustering): correlation matrices (RData format) and Newick formatted optimally ordered hierarchical trees, annotated with bootstrap scores

13. DNaseI-accessible regulatory regions: BED files with coordinates for regions of each region type (promoter, enhancer, dyadic) for each epigenome separately and RData files containing matrices of chromatin state calls for the three region types

14. Regulatory modules (promoter, enhancer, dyadic) of coordinated activity: BED files with coordinates for regions in each module, high-resolution figures (PDF, PNG) of module heatmaps for each module and txt files containing the order in which modules are plotted in the heatmaps.

15. Predicting motifs and active regulators in each cell-type/tissue/lineage

16. DNA Motif Positional Bias in Digital Genomic Footprinting Sites

Each subsection provides links to *wustl roadmap data repository* [63] and, in some cases, also links to external resources are provided.

The *wustl roadmap data repository* [63] can be accessed directly through HTTP protocol and allows to access data sorted by data type (analysis performed) or by file type (data format).

[27] ChromHMM is a software for learning and characterizing chromatin states. It is based on a multivariate Hidden Markov Model that explicitly models the presence or absence of each chromatin mark. The resulting model can then be used to systematically annotate a genome in one or more cell types [62].

4.2 Metadata

Metadata for reprocessed data from 127 consolidated epigenomes (111 Roadmap + 16 ENCODE) and unconsolidated epigenomes are available through Google Doc Spreadsheet [64], accessible from the Roadmap web portal [34], and can be downloaded in various formats from there. Metadata about single samples can be obtained by querying the *Gene Expression Omnibus* database, using the *GEO accession number (GSMxxx)*, associated with a specific sample.

4.2.1 Metadata available through spreadsheet

The spreadsheet is organized in 3 sheets.

- Consolidated_EpigenomeIDs_summary_Table: main metadata table for 127 consolidated epigenomes.
- Consolidated_EpigenomeIDs_QC: quality control measures for sets of DNase-seq and Histone ChIP-seq processed data from all 127 consolidated epigenomes.
- Unconsolidated_Release9_QC: quality control measures for DNase-seq and Histone ChIP-seq datasets from all unconsolidated epigenomes.

The whole spreadsheet can be accessed through Google Spreadsheet API [65] or downloaded individually as .xlsx, .ods, .pdf, .html; each sheet can be downloaded as .csv and .tsv.

4.2.1.1 Consolidated_EpigenomeIDs_summary_Table

This sheet is organized in 130 rows and 90 columns (31 main attributes plus some sub-attributes). The first three rows are reserved for the header, where only the first row containing the name of the columns is mandatory while the other two are optional. The optional header fields situated in the second and third row can be a sub-attribute used to identify columns with the same primary attribute, to count attribute value (e.g., how many time the attribute "sex" is "male" and how many it is "female") or to store some kind of operation result (e.g., median, sum). Each row from 4th to 130th represents a *consolidated epigenome* and each column an attribute of it. In the following list all attributes[28], types[29], sub-attributes as sub-lists, and a brief explanation if available are reported (attributes composing the primary key are marked in bold).

1. Comments[String]
2. **Epigenome ID (EID)**[String]: E060 and E064 were rejected due to data quality issues.
3. ORDER[Unsigned_Integer]: epigenome ordering based on cluster analysis.
4. GROUP[String]: organ or tissue from which the epigenome originates.

[28] Attributes followed by (H) symbol mean that they are hidden in the Google Doc spreadsheet.

[29] Attribute types are shown in square brackets after their name.

5. COLOR[Hexadecimal_Color]: color assigned to epigenome.
6. Epigenome Mnemonic[String]
7. Under Seq[Integer]: -1: low sequencing depth for multiple marks, 0: moderate sequencing depth for some mark, 1: optimal sequencing depth for all marks
8. Quality Rating[Integer]: Quality rating based on NSC and RSC[30] (Manually curated), 1: High, 0: Moderate (at least 2 marks have suboptimal scores), -1: Low
9. Auto Use Train (Core)[Boolean]: Samples used for training ChromHMM based on automated selection, (UnderSeq >= 0) & (Quality >= 0) & (Replace >= 0)
10. Manual Use Train (Core)[Boolean]: Samples labeled 1 are used in training ChromHMM (manually selected from Auto column)
11. Has K27ac[Boolean](H)
12. Train Core + K27ac[Boolean]: Samples used for training chromHMM core + K27ac models
13. Core + Ac + DNase + 9 Class1 Marks[Boolean](H)
14. Core + Ac + DNase + 13 Class1 Marks[Boolean](H)
15. Standardized Epigenome name[String]
16. Epigenome name (from EDACC Release 9 directory)[String]
17. ANATOMY[String]: describes the anatomic region of the epigenome (e.g, MUSCLE, SKIN, BRAIN).
18. TYPE[String]: epigenome classification (e.g., PrimaryTissue, CellLine, PrimaryCulture).
19. LAB (Based on 5 core histone marks only)[String]: name of the lab that generates the core histone marks data, since the data used to obtain the information associated with the epigenomes can come from different provider.

[30] A measure of enrichment derived without dependence on prior determination of enriched regions. Forward and reverse strand read coverage signal tracks are computed (number of unique mapping read starts at each base in the genome on the + and - strand counted separately). The forward and reverse tracks are shifted towards and away from each other by incremental distances and for each shift, the Pearson correlation coefficient is computed. In this way, a cross-correlation profile is computed, representing the correlation between forward and reverse strand coverage at different shifts. The highest cross-correlation value is obtained at a strand shift equal to the predominant fragment length in the dataset as a result of clustering/enrichment of relative fixed-size fragments around the binding sites of the target factor or feature. The NSC (Normalized Strand Cross-correlation coefficient) is the maximal cross-correlation value (which occurs at strand shift equal to fragment length) divided by the background cross-correlation (minimum cross-correlation value over all possible strand shifts). The RSC (Relative Strand Cross-correlation coefficient) is the fragment-length cross-correlation value minus the background cross-correlation value, divided by the phantom-peak cross-correlation value minus the background cross-correlation value [66].

20. AGE (Post Birth in YEARS/ Fetal in GESTATIONAL WEEKS/CELL LINE CL)[String]

21. SEX (Male, Female, Mixed, Unknown)[String]

22. SOLID / LIQUID[String]: e.g., LIQUID, SOLID

23. ETHNICITY[String]

24. Single Donor (SD) /Composite (C)[String]

25. DONOR / SAMPLE ALIAS[String]: list of the identifiers of the samples from which data are extracted (e.g., skin03, RM035, RM066;RM080, HuFNSC02, STL001).

26. CLASS[String]: Epigenomes were grouped in specific classes based on the diversity of assays used to profile them. Class 1 epigenomes were subjected to a thorough set of assays, including DNA methylation (whole-genome bisulfite sequencing), mRNA expression (RNA-seq), chromatin accessibility (DNase-seq), and ChIP-seq for a large set of histone modifications. Class 2 epigenomes were used to generate datasets for core histone modifications (ChIP-seq), chromatin accessibility (DNase-seq), DNA methylation (WGBS), and mRNA expression (RNA-seq). Class 3 epigenomes were used to generate datasets for the core histone modifications (ChIP-seq), chromatin accessibility (DNase-seq), DNA methylation (RRBS or MeDIP/MRE assays), and mRNA expression (microarrays). Class 4 epigenomes were subjected to the same assays as Class 3 epigenomes except for chromatin accessibility data. Class 5 epigenomes were subjected to ChIP-seq assays for the minimum set of five core histone modifications.

27. Pool Filenames[String]: separated file names of the datasets from which mandatory data are taken.

    (a) Input

    (b) H3K9me3

    (c) H3K27me3

    (d) H3K4me3

    (e) H3K4me1

    (f) H3K36me3

    (g) H3K27ac

    (h) H3K9ac

    (i) DNase

    (j) RNA-seq

    (k) WGBS

    (l) RRBS

    (m) mCRF

28. Nreads (36 bp) 30 M[Unsigned_Integer]: Number of 36 bp[31] length reads subsampled with respect to a maximum depth of 30 million[32] reads.

    (a) Input
    (b) H3K9me3
    (c) H3K27me3
    (d) H3K4me3
    (e) H3K4me1
    (f) H3K36me3
    (g) H3K27ac
    (h) H3K9ac
    (i) DNase

29. NSC (Signal to noise)[Float]: Normalized strand cross-correlation coefficient. Genome-wide correlation between + and - strand read counts when shifted by fraglen/2 relative to background. It represents enrichment of clustered ChIP fragments around target sites. Input-DNA values are used as a reference for calibration. Diffused marks such as H3K9me3 inherently have lower signal to noise ratios, and hence NSC, compared to strong active marks such as H3K4me3. Samples with very low seq. depth can have abnormally high NSC since there is a significant depletion of 'background'. i.e. these samples tend to have higher specificity but very low sensitivity.

    (a) Input
    (b) H3K9me3
    (c) H3K27me3
    (d) H3K4me3
    (e) H3K4me1
    (f) H3K36me3
    (g) H3K27a
    (h) H3K9ac
    (i) DNase

[31] The raw Release 9 read alignment files contain reads that are pre-extended to 200 bp. However, there were significant differences in the original read lengths across the Release 9 raw datasets reflecting differences between centers and changes of sequencing technology during the course of the project (36 bp, 50 bp, 76 bp and 100 bp). To avoid artificial differences due to mappability, for each consolidated data set the raw mapped reads were uniformly truncated to 36 bp and then refiltered using a 36 bp custom mappability track to only retain reads that map to positions (taking strand into account) at which the corresponding 36-mers starting at those positions are unique in the genome [35].

[32] To avoid artificial differences in signal strength due to differences in sequencing depth, all consolidated histone mark datasets (except the additional histone marks of the seven deeply profiled epigenomes) were uniformly subsampled to a maximum depth of 30 million reads (the median read depth over all consolidated samples) [35].

30. RSC (Phantom Peak)[Float]: Relative strand cross-correlation coefficient. Relative enrichment of fragment-length cross-correlation to read-length cross-correlation (phantom peak). The read-length cross-correlation is a baseline correlation that is entirely due to an inherent mappability-bias for reads separated on + and - strand by read-length. The fragment length cross-correlation (which is due to clustering of relatively fixed sized fragment around target sites) should be able to beat the read-length correlation for highly enriched datasets with sufficient localized target sites. So a RSC value > 0.8 is desirable in general. Marks that tend to be enriched at repeat-like regions and those that have low signal to noise ratios with diffused genome-wide patterns can have stronger read-length peaks and RSC values < 0.8.

    (a) Input
    (b) H3K9me3
    (c) H3K27me3
    (d) H3K4me3
    (e) H3K4me1
    (f) H3K36me3
    (g) H3K27a
    (h) H3K9ac
    (i) DNase

31. Pool Filenames[String]: separated file names of the datasets from which optional data are taken (if any).

    (a) H3K4me2
    (b) H2AK5ac
    (c) H2BK120ac
    (d) H2BK5ac
    (e) H3K18ac
    (f) H3K23ac
    (g) H3K4ac
    (h) H3K79me1
    (i) H4K8ac
    (j) H2BK12ac
    (k) H3K14ac
    (l) H4K91ac
    (m) H2A.Z
    (n) H2BK15ac
    (o) H3K79me2
    (p) H2BK20ac

(q) H3K56ac

(r) H4K20me1

(s) H4K5ac

(t) H3K23me2

(u) H2AK9ac

(v) H3K9me1

(w) H3T11ph

(x) H4K12ac

4.2.1.2 Consolidated_EpigenomeIDs_QC

This section of the metadata spreadsheet is a table of 1160 rows per 21 attribute columns. The first row is dedicated to the header containing the attribute name. The remaining 1159 rows represent the *sets of ChIP-Seq and DNase-Seq data* associated with the consolidated epigenomes. Each set comes from a specific experiment and belongs to a unique consolidated epigenome, so it is identified by the couple of attributes <"MARK", EID">. Below the cell attributes, types and available explanations are listed (attributes composing the primary key are marked in bold).

1. **MARK**[String]: Histone mark used in the ChIP-Seq experiment that originates the data or "DNase" value if coming from DNase-Seq experiment.

2. **EID**[String]: The consolidated epigenome to which the data are referred.

3. E-Mnemonic[String]

4. Standardised epigenome name[String]

5. Epigenome name (from EDACC Release 9 directory)[String]

6. MARK CLASS[String]: "core" if the histone associated with the data is a mandatory one for consolidated epigenome definition or DNase data, "class1" if it is optional in the consolidated epigenome definition.

7. FNAME[String]: string that identifies files inside [63] containing the data (e.g., E100-H3K4me3.*).

8. NREADS[Unsigned_Integer]: Number of reads.

9. FRAGLEN[Unsigned_Integer]: Estimated fragment length parameter used during peak calling data process phase[33].

10. FCC[Float](H): parameter used in some calculations or algorithms.

[33] For the histone ChIP-seq data, the MACSv2.0.10 peak caller was used to compare ChIP-seq signal to a corresponding whole-cell extract (WCE) sequenced control to identify narrow regions of enrichment (peaks) that pass a Poisson P value threshold 0.01, broad domains that pass a broad-peak Poisson P value of 0.1 and gapped peaks which are broad domains ($P < 0.1$) that include at least one narrow peak ($P < 0.01$). Fragment lengths for each data set were pre-estimated using strand cross-correlation analysis and the SPP peak caller package and these fragment length estimates were explicitly used as parameters in the MACS2 program (–shift_size = fragment_length/2) [35].

11. RLEN[Unsigned_Integer](H): parameter used in some calculations or algorithms.

12. RCC[Float](H): parameter used in some calculations or algorithms.

13. MINLEN[Unsigned_Integer](H): parameter used in some calculations or algorithms.

14. MINCC[Float](H): parameter used in some calculations or algorithms.

15. NSC (Signal to noise)[Float]: Normalized strand cross-correlation coefficient. Genome-wide correlation between + and - strand read counts when shifted by fraglen/2 relative to background. It represents enrichment of clustered ChIP fragments around target sites. Input-DNA values are used as a reference for calibration. Diffused marks such as H3K9me3 inherently have lower signal to noise ratios, and hence NSC, compared to strong active marks such as H3K4me3. Samples with very low seq. depth can have abnormally high NSC since there is a significant depletion of 'background'. i.e. these samples tend to have higher specificity but very low sensitivity.

16. RSC (Phantom Peak)[Float]: Relative strand cross-correlation coefficient. Relative enrichment of fragment-length cross-correlation to read-length cross-correlation (phantom peak). The read-length cross-correlation is a baseline correlation that is entirely due to an inherent mappability-bias for reads separated on + and - strand by read-length. The fragment length cross-correlation (which is due to clustering of relatively fixed sized fragment around target sites) should be able to beat the read-length correlation for highly enriched datasets with sufficient localized target sites. So a RSC value > 0.8 is desirable in general. Marks that tend to be enriched at repeat-like regions and those that have low signal to noise ratios with diffused genome-wide patterns can have stronger read-length peaks and RSC values < 0.8.

17. SPOT[Float]: quality score used to select signal enrichment regions, computed based on regions identified with the HotSpot peak caller.

18. FindPeaks[Float]: quality score used to select signal enrichment regions, inferred based on peak calls made using the FindPeaks software.

19. Poisson[Float]: Metric used to select signal enrichment regions, derived by modelling the read distribution in genome-tiling 1,000-bp windows with a Poisson distribution and selecting as enriched regions windows with $P < 0.05$.

20. % of Imputed[34] Top 1% in Observed[35] Top 1%[Float]: Percentage of the 1% best quality score imputed regions that have a better quality score than the 1% best quality score observed regions

21. Correlation of Imputed with Observed[Float]

[34] In [35] the data composing the consolidated epigenomes are referred as imputed data (sets).

[35] In [35] the data composing the unconsolidated epigenomes are referred as observed data (sets).

The last two attributes values (*%of Imputed Top 1% in Observed Top 1%* and *Correlation of Imputed with Observed*) are missing from row 1134 to 1160.

This second table can be related to the first sheet through the EID attribute (in relational database terminology the second tables is in an n:1 relation with the first one through the external key EID).

4.2.1.3 Unconsolidated_Release9_QC

The last sheet is composed of a header row and 2181 *ChIP-Seq and DNase-Seq datasets* associated with unconsolidated epigenomes rows. To each dataset there are associated 17 attribute columns reported below with provided definition where necessary (primary key in bold).

1. LAB[String]
2. GROUP[Unknown]
3. E-Mnemonic[UnKnown]
4. **CELL TYPE / TISSUE**[String]
5. **MARK**[String]: Histone mark used in the ChIP-Seq experiment that originates the data or "DNase" value if coming from DNase-Seq experiment.
6. **DONOR**[String]: Identifier of the sample provider.
7. FILENAMES[String]
8. CONTROL FILE NAME[String]
9. NREADS (36 bp mappability filtered)[Unsigned_Integer]: Number of reads filtered using a 36 bp custom mappability track.
10. FRAGLEN[Unsigned_Integer]: Estimated fragment length parameter used during peak calling data processing phase.
11. FCC[Float](H): parameter used in some calculations or algorithms.
12. RLEN[Unsigned_Integer](H): parameter used in some calculations or algorithms.
13. RCC[Float](H): parameter used in some calculations or algorithms.
14. MINLEN[Unsigned_Integer](H): parameter used in some calculations or algorithms.
15. MINCC[Float](H): parameter used in some calculations or algorithms.
16. NSC (Signal to noise)[Float]: Normalized strand cross-correlation coefficient. Genome-wide correlation between + and - strand read counts when shifted by fraglen/2 relative to background. It represents enrichment of clustered ChIP fragments around target sites. Input-DNA values are used as a reference for calibration. Diffused marks such as H3K9me3 inherently have lower signal to noise ratios, and hence NSC, compared to strong active marks such as

H3K4me3. Samples with very low seq. depth can have abnormally high NSC since there is a significant depletion of 'background'. i.e. these samples tend to have higher specificity but very low sensitivity.

17. RSC (Phantom Peak)[Float]: Relative strand cross-correlation coefficient. Relative enrichment of fragment-length cross-correlation to read-length cross-correlation (phantom peak). The read-length cross-correlation is a baseline correlation that is entirely due to an inherent mappability-bias for reads separated on + and - strand by read-length. The fragment length cross-correlation (which is due to clustering of relatively fixed sized fragment around target sites) should be able to beat the read-length correlation for highly enriched datasets with sufficient localized target sites. So a RSC value > 0.8 is desirable in general. Marks that tend to be enriched at repeat-like regions and those that have low signal to noise ratios with diffused genome-wide patterns can have stronger read-length peaks and RSC values < 0.8.

Attributes "GROUP" and "E-Mnemonic" are missing for all the samples.

This table can be related to the second sheet using the "MARK" and "CELL TYPE/TISSUE" attribute in a quite direct way ("CELL TYPE/TISSUE" is called "epigenome name", some attribute values are different). Data in this sheet can be also related to those in the first one through the relationship with the second sheet.

4.2.2 Metadata available through GEO

The structure of the metadata provided by GEO is very flexible and can differ significantly according to the sample. This is because the GEO portal is meant to store data originated by a broad spectrum of experiments and coming from different sources; so the data publisher customizes the metadata in a way that best fits them accordingly to the *GEOarchive metadata guidelines table* [67]. REP metadata specifically differ between datasets obtained from sequencing and those obtained from microarrays. A list of the most typical metadata associated with the REP datasets coming from NGS experiments follow.

1. Status: access status and date.
2. Title: short description of the sample, experiment, and cell line.
3. Sample type: "SRA" for NGS experiment, "RNA" for array-based experiment.
4. Source name: sample identifier.
5. Organism
6. Characteristics: detailed description of the sample and experiment through custom tags (e.g., sex, biomaterial_type, chip_antibody).
7. Extracted molecule
8. Extraction protocol
9. Library strategy
10. Library source

11. Library selection
12. Instrument model
13. Description: additional informations about the dataset and external link.
14. Data processing: description of the processing performed to obtain the files associated with the dataset.
15. Submission date
16. Last update date
17. Contact name
18. Organization name
19. Street address
20. City
21. State/province
22. ZIP/Postal code
23. Country
24. Platform ID: link to the associated GEO platform.
25. Series: links to the associated GEO series
26. Relations: links to additional resources associated

A representative list of the array-based datasets metadata is below:

1. Status: access status and date.
2. Title: short description of the sample, experiment, and cell line.
3. Sample type: "SRA" for NGS experiment, "RNA" for array-based experiment.
4. Source name: sample identifier.
5. Organism
6. Characteristics: detailed description of the sample and experiment through custom tags (e.g., sex, biomaterial_type, chip_antibody).
7. Treatment protocol
8. Growth protocol
9. Extracted molecule
10. Extraction protocol
11. Label

12. Label protocol
13. Hybridization protocol
14. Scan protocol
15. Description: additional informations about the dataset and external link.
16. Data processing: description of the processing performed to obtain the files associated with the dataset.
17. Submission date
18. Last update date
19. Contact name
20. Organization name
21. Street address
22. City
23. State/province
24. ZIP/Postal code
25. Country
26. Platform ID: link to the associated GEO platform.
27. Series: links to the associated GEO series
28. Relations: links to additional resources associated

"Characteristics", "Description", "Data processing", and "Relations" attributes have some sub-attribute that depends on the sample. Description and data processing attributes are also unstructured, so their representation depend on the file format. Also, the position inside the file and name of the attributes change slightly depending on the file format.

Metadata can be downloaded using HTTP protocol by constructing a URL to retrieve data. URLs are formatted as in the following examples:

Example 4.2.1 URL that retrieves a text file containing the 'brief' view of accession GPL96

```
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=gpl96&targ=
self&view=brief&form=text
```

Example 4.2.2 URL that retrieves a xml containing the 'brief' view of accession GSM428289

```
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM42829&targ=
self&view=brief&form=xml
```

There are four parameter to set:

- acc: a valid GEO accession identifier (gplxxx, gsmxxx or gsexxx)
- targ: the metadata to retrieve (self, gsm, gpl, gse or all)[36]
- view: amount of data to show (brief, quick, data or full)
- form: format file to get (text, html or xml)

[36] In GEO there are three kinds of entity: samples, identified by GSMxxx IDs; Platform, identified by GPLxxx IDs and series, identified by GSExxx IDs. Those objects are related together (e.g., a series is a set of samples, a platform is used to produce different series), so the targ parameter can be used to get the associated object ("gsm", "gpl", or "gse") to that specified by acc parameter, instead of getting that one ("self"). "all" value retrieves all the metadata associated with the ID specified in acc.

Chapter 5

Datasets selection

From the analysis of the REP source described in the previous chapter, we know that it provides a lot of different data, at different stages of the data analysis pipeline. Since we want to integrate data to make them available to be queried using GMQL and perform tertiary analysis over them, we focus our attention on secondary analysed data. More precisely we are interested in:

- broad and narrow regions associated to protein binding sites identified using ChIP and DNase experiments;
- methylated regions;
- expression quantification of regions associated to genes, exons, and other significant known regions of interest.

In fact, this are the data usually used to identify DNA regularization functions, genome interaction mechanisms, new genes or relevant DNA regions, and anomalous behaviours at molecular level that lead to pathological states (such as cancer, for example). Once it is clear what are the data of interest, we try to group that data into datasets of semantically homogeneous data (but still heterogeneous in terms of format). For each identified dataset we perform a more deep inspection respect of that performed during the source analysis to detect if they can be imported in a GDM repository or not[1]. For data that can be imported we define the transformation required to convert them in GDM format and the transformation to perform over the associated metadata as well.

5.1 Consolidated epigenomes enriched region peaks

The processed peaks data associated with the Roadmap consolidated epigenomes can be downloaded from *wustl roadmap data repository* [63] through HTTP protocol starting from the base directory *byFileType/peaks/consolidated/*. The directory structure is reported in Figure 5.1. Three types of peaks are available (i.e., broad, narrow and gapped) collected in three distinct directories that contain one or more files

[1] All the textual metadata can be imported after some kind of transformation, but data publish in binary format can't be manipulated, so they are discarded.

for each experiment performed over a consolidated epigenome. These subdirectories contain also two folders dedicated to the hammock (see Sec. 5.1.1.4) and UCSC compatible version of the files. The UCSC compatible files differ from the standard ones only for the score[2] region attribute. Only for broadPeaks, it is also available a folder containing the most recent version of the DNase Hotspot peak call datasets and associated index files. All the consolidated peaks files are obtained starting from raw ChIP-seq or DNase BED files.

The BED standard demands 0-based coordinates and also the formats of the peaks files, so the 0-based coordinate system is maintained through the elaboration required to produce the peaks files.

[2] The score is used only to define how much dark is a region when represented in a genome browser. The UCSC genome browser requires a score ranging from 0-1000. The peak regions produced by REP can go beyond the upper range (this happens in the example 5.1.5), so in the UCSC compatible version, the scores are scaled to fit in the range.

- `byFileType/peaks/consolidated/` .... Base directory, no files here.
  - `broadPeak/` .... It contains a broadPeak file for all the experiments performed on each epigenome, depending on epigenome class. It also contains DNase broad region files in BED format.
    - `DNase/` .... It contains all the DNase broad region files in BED format and their index files (.tbi); each broad region file has a tbi associated. The BED files in this directory are the same as those present in the parent directory but updated more recently.
    - `hammock/` .... It contains the hammock version of the ChIP-seq broadPeak files.
    - `ucsc_compatible/` .... It contains the UCSC-compatible version of the ChIP-seq broadPeak files.
  - `gappedPeak/` .... It contains a gappedPeak file for all the experiments performed on each epigenome, depending on epigenome class.
    - `hammock/` .... It contains the hammock version of the ChIP-seq gappedPeak files.
    - `ucsc_compatible/` .... It contains the UCSC-compatible version of the ChIP-seq gappedPeak files.
  - `narrowPeak/` .... It contains a narrowPeak file for all the experiments performed on each epigenome, depending on epigenome class. It also contains DNase narrow region files in BED format.
    - `hammock/` .... It contains the hammock version of the ChIP-seq narrowPeak files.
    - `ucsc_compatible/` .... It contains the UCSC-compatible version of the ChIP-seq narrowPeak files.

Figure 5.1: Directory structure containing the peak data of the consolidated epigenomes.

5.1.1 ChIP-seq

For the histone ChIP-seq data, the MACSv2.0.10 peak caller was used to map ChIP-seq signal to a corresponding cell line control sequence to identify narrow regions of enrichment (peaks) that pass a Poisson P value threshold 0.01, broad domains that pass a broad-peak Poisson P value of 0.1, and gapped peaks which are broad domains (P < 0.1) that include at least one narrow peak (P < 0.01) [35].

All the files are inside a .gz archive and follow a standard ENCODE format. Each file name matches the pattern [EpigenomeID]-[Marker].[Format].gz. In all the three formats the 4th region attribute (name) is a unique region name that matches the pattern Rank_[RowNumber]. An exception to these conventions is represented by the hammock files because they are directly derived by standard ENCODE files and they match a slight different naming pattern (see Section 5.1.1.4).

5.1.1.1 broadPeak

This format is used to provide called regions of signal enrichment based on pooled, normalized (interpreted) data. It is a BED 6+3 format [56].

Example 5.1.3 First line of the E001-H3K4me1.broadPeak file [68].

$$\underbrace{chr3}_{\text{chrom}}\ \underbrace{195423086}_{\text{chromStart}}\ \underbrace{195426700}_{\text{chromEnd}}\ \underbrace{Rank\_1}_{\text{name}}\ \underbrace{78}_{\text{score}}\ \underbrace{.}_{\text{strand}}\ \underbrace{3.23874}_{\text{signal}}\ \underbrace{10.67458}_{\text{pValue}}\ \underbrace{7.87721}_{\text{qValue}}$$

Files in broadPeak format are imported in GMQL using the schema reported below.

```
<?xml version="1.0" encoding="UTF-8"?>
<gmqlSchemaCollection name="broadPeak"
    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="STRING">name</field>
<field type="INTEGER">score</field>
<field type="CHAR">strand</field>
<field type="DOUBLE">signal</field>
<field type="DOUBLE">pvalue</field>
<field type="DOUBLE">qvalue</field>
</gmqlSchema>
</gmqlSchemaCollection>
```

5.1.1.2 gappedPeak

This format is used to provide called regions of signal enrichment based on pooled, normalized (interpreted) data where the regions may be spliced or incorporate gaps in the genomic sequence. Gapped/chained regions of enrichment are defined as broadPeaks that contain at least one strong narrowPeak. It is a BED12+3 format [57].

Example 5.1.4 First line of the E010-H2A.Z.gappedPeak file [69].

.

$$\underbrace{chr6}_{\text{chrom}}\,\underbrace{58773503}_{\text{chromStart}}\,\underbrace{58779548}_{\text{chromEnd}}\,\underbrace{Rank\_1}_{\text{name}}\,\underbrace{350}_{\text{score}}\,\underbrace{\quad.\quad}_{\text{strand}}\,\underbrace{58773564}_{\text{thickStart}}\,\underbrace{58779480}_{\text{thickEnd}}\,\underbrace{\quad 0\quad}_{\text{itemRgb}}$$

$$\underbrace{\quad 2\quad}_{\text{blockCount}}\,\underbrace{522,3244}_{\text{blockSizes}}\,\underbrace{61,2733}_{\text{blockStarts}}\,\underbrace{2.42899}_{\text{signal}}\,\underbrace{39.41305}_{\text{pValue}}\,\underbrace{35.07461}_{\text{qValue}}$$

Each line represents a broad peak region identified by *chromStart* and *chromEnd* attributes. Inside each broad peak, there are one or more narrow peaks (blocks), also non-contiguous (gapped). The broad peak region is usually represented graphically with a thin line, whereas the narrow peak regions are represented by a thick line. The number of narrow peaks is indicated by *blockCount*. *thickStart* and *thickEnd* identify the start bp of the first narrow region and the stop bp of the last narrow region, respectively. The position of the narrow peaks inside the broad one can be calculated using *blockSizes* (it is a list of the sizes of the narrow regions in bp, separated by commas) and *blockStart* (it is a list of comma-separated relative starting positions of the narrow regions, calculated starting from *chromStart*). The GMQL schema is:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gmqlSchemaCollection name="gappedPeak"
   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="STRING">name</field>
<field type="INTEGER">score</field>
<field type="CHAR">strand</field>
<field type="LONG">thickstart</field>
<field type="LONG">thickend</field>
<field type="STRING">itemrgb</field>
<field type="INTEGER">blockcount</field>
<field type="STRING">blocksizes</field>
<field type="STRING">blockstarts</field>
<field type="DOUBLE">signal</field>
<field type="DOUBLE">pvalue</field>
<field type="DOUBLE">qvalue</field>
</gmqlSchema>
</gmqlSchemaCollection>
```

5.1.1.3 narrowPeak

This format is used to provide called peaks of signal enrichment based on pooled, normalized (interpreted) data. It is a BED6+4 format. [55].

Example 5.1.5 First line of the E115-H3K27me3.narrowPeak file [70].

$$\underbrace{chr10}_{\text{chrom}}\underbrace{22610916}_{\text{chromStart}}\underbrace{22612605}_{\text{chromEnd}}\underbrace{Rank\_1}_{\text{name}}\underbrace{1207}_{\text{score}}\underbrace{\ .\ }_{\text{strand}}\underbrace{28.81104}_{\text{signalValue}}\underbrace{120.76161}_{\text{pValue}}\underbrace{111.27103}_{\text{qValue}}\underbrace{268}_{\text{peak}}$$

The GMQL schema is:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gmqlSchemaCollection name="narrowPeak"
    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="STRING">name</field>
<field type="INTEGER">score</field>
<field type="CHAR">strand</field>
<field type="DOUBLE">signal</field>
<field type="DOUBLE">pvalue</field>
<field type="DOUBLE">qvalue</field>
<field type="INTEGER">peak</field>
</gmqlSchema>
</gmqlSchemaCollection>
```

5.1.1.4 hammock

New file format for encoding richly annotated genomics features and used to display epigenome tracks through browser [71]. A hammock file is compressed and indexed so it is associated with a tbi[3] index file. A hammock-format file is a line-oriented, tabular text file. Each line is one genomic feature, and must have 4 fields:

1. chromosome name
2. start coordinate (0 offset)
3. stop coordinate
4. JSON string

The 4th field is used to encode optional annotation information as a JSON string. This string is a JSON object without the outmost curly brackets, in the form of "key":"value" pairs. Hammock files will not be imported in GMQL.

Example 5.1.6 Row 222486 in E001-H3K4me1.broadPeak.hammock file [73].

[3] File produced by Tabix software tool and used to index position sorted files in TAB-delimited formats with the purpose to quickly retrieve features overlapping specified regions [72].

$$\underbrace{chr3}_{\text{chrom}}\underbrace{195423086}_{\text{chromStart}}\underbrace{195426700}_{\text{ChromStop}}$$
$$\underbrace{scorelst : [3.23874, 10.67458], id : 1, name : ''Rank\_1'',}_{\text{JSON string}}$$

The name attribute is no longer corresponding to the line number, but it refers to the line in which the region is placed in the original ENCODE format file. This example reports the same region shown in the example 5.1.5, but now it is no more in the first line. In the standard ENCODE file, rows are sorted by score whereas in hammock format the rows are sorted by position inside the genome.

Hammock files are obtained by converting ENCODE standard files (broad, narrow or gapped peak). Since they contain the same data contained in the standard ENCODE files, they are not imported in GMQL.

5.1.2 DNase

For DNase data, narrow peaks are called using two different peak callers on all datasets, MACS2 and HOTSPOT. MACS2 was used with a p-value threshold of 0.01. The files are in the standard narrowPeak format and they are placed in the source directory *byFileType/peaks/consolidated/narrowPeak/*. The HOTSPOT peak caller was also used to call broad domains of chromatin accessibility, both with a FDR of 1% and without applying a thresholds. Files generated by HOTSPOT are in BED format. Narrow region BED files are placed in the source directory *byFileType/peaks/consolidated/narrowPeak/*. Broad region BED files are placed in the source directory *byFileType/peaks/consolidated/broadPeak/* and *byFileType/-peaks/consolidated/broadPeak/DNase*[4]. Three different types of DNase peak call files can be associated with narrow regions of contiguous chromatin accessibility:

- [EpigenomeID]-[Marker].macs2.narrowPeak.gz, obtained by using MACS2 peak caller with a p-value threshold of 0.01.

Example 5.1.7 First line of the E003-DNase.macs2.narrowPeak file [74]. Rows are in standard narrowPeak encode format.

$$\underbrace{chr21}_{\text{chrom}}\underbrace{9825416}_{\text{chromStart}}\underbrace{9827604}_{\text{chromEnd}}\underbrace{Rank\_1}_{\text{name}}\underbrace{2022}_{\text{score}}\underbrace{\quad . \quad}_{\text{strand}}\underbrace{61.01925}_{\text{signalValue}}\underbrace{202.23363}_{\text{pValue}}$$
$$\underbrace{192.74306}_{\text{qValue}}\underbrace{1759}_{\text{peak}}$$

- [EpigenomeID]-[Marker].hotspot.fdr0.01.peaks.bed.gz, contains all the narrow peaks in FDR 1% obtained using Hotspot peak caller.

[4] Since files inside *byFileType/peaks/consolidated/broadPeak/DNase* have been updated more recently, we choose to import them inside GMQL.

Example 5.1.8 First line of the E003-DNase.fdr0.01.peaks.bed[5] file [75]. 5th column value represents the peak tag density, 6th column is the z-score.

$$\underbrace{chr1}_{\text{chrom}}\ \underbrace{10140}_{\text{chromStart}}\ \underbrace{10290}_{\text{chromEnd}}\ \underbrace{.}_{\text{strand}}\ \underbrace{26}_{\text{peak}}\ \underbrace{15.1272}_{\text{z-score}}$$

- [EpigenomeID]-[Marker].hotspot.all.peaks.bed.gz, genome-wide tag density Hotspot unthresholded peak calls.

Example 5.1.9 First line of the E003-DNase.all.peaks.bed file[5] [76]. Peak value represents the peak tag density.

$$\underbrace{chr1}_{\text{chrom}}\ \underbrace{10140}_{\text{chromStart}}\ \underbrace{10290}_{\text{chromEnd}}\ \underbrace{.}_{\text{strand}}\ \underbrace{26.000000}_{\text{peak}}$$

Broad domains of chromatin accessibility are provided in two different files:

- [EpigenomeID]-[Marker].hotspot.fdr0.01.broad.bed.gz, contains all the FDR 1% broad domains.

Example 5.1.10 First line of the E003-DNase.fdr0.01.hot.bed[5] file [77]. 5th column represents the z-score.

$$\underbrace{chr1}_{\text{chrom}}\ \underbrace{10147}_{\text{chromStart}}\ \underbrace{10277}_{\text{chromEnd}}\ \underbrace{.}_{\text{strand}}\ \underbrace{15.127200}_{\text{z-score}}$$

- [EpigenomeID]-[Marker].hotspot.broad.bed.gz, contains genome-wide tag density HOTSPOT peak calls.

Example 5.1.11 First line of the E003-DNase.hot.bed[5] file [78]. 5th column represents the z-score.

$$\underbrace{chr1}_{\text{chrom}}\ \underbrace{10147}_{\text{chromStart}}\ \underbrace{10277}_{\text{chromEnd}}\ \underbrace{.}_{\text{strand}}\ \underbrace{15.127200}_{\text{z-score}}$$

No DNase gappedPeak calls are available.

The DNase narrow regions files in standard narrowPeak format are imported in GMQL inside the same dataset of the ChIP-seq narrowPeak files since they have the same schema. All the .bed file are imported in the same dataset using the following GMQL schema:

[5] The name of the files inside the GZip archives follow different naming conventions from those used to naming the archives. While importing in GMQL, the naming conventions used for the archives is adopted.

```
<?xml version="1.0" encoding="UTF-8"?>
<gmqlSchemaCollection name="HOTSPOT_BED"
   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="INTEGER">peak</field>
<field type="DOUBLE">zscore</field>
</gmqlSchema>
</gmqlSchemaCollection>
```

Since in the narrow regions called without FDR threshold lack the zscore attribute and the broad regions lack the peak attribute, the missing attribute values are set to NULL value.

5.2 Consolidated epigenomes RNA-seq data

mRNA-seq datasets from 56 reference epigenomes that had RNA-seq data have been processed by aligning 75 bp or 100 bp long reads using the BWA aligner, and read coverage profiles have been generated separately for positive and negative strand strand-specific libraries. The set of these bigWig files represents RNA-seq signal tracks. Starting from the RNA-seq signal, exon and gene expressions are calculated using a modified RPKM (Reads Per Kilobase Million) measure[6].

5.2.1 RNA-seq signal tracks

At *wustl roadmap data repository* [63], starting from the base directory */byDataType-/rna/signal/normalized_bigwig/stranded/* for each of the 56 epigenomes, there are available two files in bigWig binary format [79], one for the positive strand and one for the negative one. In the folder it is included also a sample (identified by E000 epigenomeID) representing a Universal Human Reference RNA sample (HUR). Each file name matches the pattern [EpigenomeID].[Cell_line]. norm.[Strand].bw, where the [Strand] part can assume value "neg" or "pos". Because the files are binary, they can not be imported in GMQL without being converted.

Each file can be associated with metadata available through *Google spreadsheet: Metadata and quality control*, Consolidated_EpigenomeIDs_ summary_ Table sheet [64] by matching the [EpigenomeID] in the file name and the *EpigenomeID (EID)* attribute of the table in the sheet.

[6] The total number of reads aligned into coding exons is used as normalization factor; reads from the mitochondrial genome, reads falling into genes coding for ribosomal proteins, and reads falling into top 0.5% expressed exons have been excluded. RPKM for a gene is calculated using the total number of reads aligned into all merged exons for a gene normalized by total exonic length.

5.2.2 RNA-seq expression quantifications

The data regarding gene expression are collected in tab-delimited matrix files compressed in gz archives and available in the directory *byDataType/rna/expression/* at *wustl roadmap data repository* [63]. The files are:

- 57epigenomes.RPKM.pc.gz: RPKM expression matrix for protein coding genes;
- 57epigenomes.N.pc.gz: RNA-seq read counts matrix for protein coding genes;
- 57epigenomes.RPKM.nc.gz: RPKM expression matrix for non-coding RNAs;
- 57epigenomes.N.nc.gz: RNA-seq read counts matrix for non-coding RNAs;
- 57epigenomes.exon.RPKM.pc.gz: RPKM expression matrix for protein coding exons;
- 57epigenomes.exon.N.pc.gz: RNA-seq read counts matrix for protein coding exons;
- 57epigenomes.exon.RPKM.nc.gz: RPKM expression matrix for protein non-coding exons;
- 57epigenomes.exon.N.nc.gz: RNA-seq read counts matrix for protein non-coding exons;
- 57epigenomes.exn.RPKM.rb.gz: RPKM expression matrix for ribosomal gene exons;
- 57epigenomes.exn.N.rb.gz: RNA-seq read counts matrix for ribosomal gene exons;
- 57epigenomes.RPKM.intronic.pc.gz: RPKM expression matrix for intronic protein coding RNA elements;
- 57epigenomes.N.intronic.pc.gz: RNA-seq read counts matrix for intronic protein coding RNA elements;
- 57epigenomes.RPKM.rb.gz: RPKM expression matrix for ribosomal genes;
- 57epigenomes.N.rb.gz: RNA-seq read counts matrix for ribosomal genes.

Each file is a matrix M, where the value M_{ij} is an expression level value for the gene i in the epigenome j. In the first row are reported the EpigenomeIDs and in the first column the genes identifiers (ENSEMBL IDs).

Example 5.2.12 First two lines of the 57epigenomes.N.intronic.pc file [80].

```
gene_id E000 E003 E004 E005 E006 E007 E011 E012 E013 E016 E024
    E027 E028 E037 E038 E047 E050 E053 E054 E055 E056 E057
    E058 E059 E061 E062 E065 E066 E070 E071 E079 E082 E084
    E085 E087 E094 E095 E096 E097 E098 E100 E104 E105 E106
    E109 E112 E113 E114 E116 E117 E118 E119 E120 E122 E123
    E127 E128
```

```
ENSG00000000003 23 254  229  160  108  174  141  559  110  107
    304  58 31 24 47 27 2303 32 59 12 78 10 51 15 87 9 2 13 104 27
    216 34 1449 924  71 3 45 37 892  56 106  148  25 374  70 92 25
    271 0 749  373  22 113  467  45 97 131
```

Example 5.2.13 First two lines of the 57epigenomes.exon.N.nc file [81].

```
exon_location gene_id E000 E003 E004 E005 E006 E007 E011 E012
    E013 E016 E024 E027 E028 E037 E038 E047 E050 E053 E054
    E055 E056 E057 E058 E059 E061 E062 E065 E066 E070 E071
    E079 E082 E084 E085 E087 E094 E095 E096 E097 E098 E100
    E104 E105 E106 E109 E112 E113 E114 E116 E117 E118 E119
    E120 E122 E123 E127 E128
chr10:100011780-100011959<1 ENSG00000230928 124 0 0 0 0 33 0
    200  0 0 194  224  148  542  647  669  1793 152  281
    130  68 0 0 0 608  307  0 0 0 0 448  13 357  118  110
    0 347  192  272  300  478  472  0 851  268  0 0 431
    150  150  450  0 80 254  0 390  0
```

In the same directory there is also the Ensembl_v65.Gencode_v10.ENSG.gene_info textual file containing the coordinates and other annotations for the genes that are the rows in the expression matrices.

Example 5.2.14 The Ensembl_v65.Gencode_v10.ENSG.gene_info first two rows [82].

$$\underbrace{ENSG00000000003}_{\text{geneID}}\ \underbrace{X}_{\text{chrom}}\ \underbrace{99883667}_{\text{chromStart}}\underbrace{99894988}_{\text{chromEnd}}\ \underbrace{-1}_{\text{strand}}\ \underbrace{protein\_coding}_{\text{gene type}}\underbrace{TSPAN6}_{\text{symbol}}$$

$$\underbrace{tetraspanin\_6\_[Source : HGNC\_Symbol; Acc : 11858]}_{\text{name\_and\_source}}$$

The *Acc* field of the name_and_source attribute identifies the accession number of the gene and it is used by the specified data source (in this example the HGNC, acronym of Human Gene Nomenclature Committee) to uniquely identify the gene.

The expression data in the format provided by REP can not be directly imported in GMQL. They must be before transformed in valid regions with attributes in tab-separated format. The expression data are originally provided inside 14 matrix files containing two types of expression quantifications (read count and RPKM) for 7 types of data: genes (i.e., protein coding genes, ribosomal genes), exons (protein coding exons, protein non coding exons, ribosomial gene exons), intronic protein coding RNA elements and non-coding RNAs. For each type of data and epigenome (column of a matrix file), a new file with a row for each row in the matrix file can be created by relating the expression quantification values with the associated genetic coordinates. 7 (data types) multiplied by 57 (epigenomes) files are generated. A row of a generated file represents a gene, exon or intronic region and its expression value (both read count and RPKM) inside a particular epigenome. The GMQL schema of the generated files is:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gmqlSchemaCollection name="RNA_expression"
   xmlns="http://genomic.elet.polimi.it/entities">
<gmqlSchema type="TAB" coordinate_system="1-based">
<field type="STRING">chrom</field>
<field type="LONG">start</field>
<field type="LONG">end</field>
<field type="CHAR">strand</field>
<field type="INTEGER">read_count</field>
<field type="DOUBLE">RPKM</field>
<field type="STRING">geneID</field>
<field type="STRING">gene_type</field>
<field type="STRING">gene_symbol</field>
<field type="STRING">gene_name</field>
<field type="STRING">gene_name_source</field>
<field type="STRING">gene_name_accession</field>
</gmqlSchema>
</gmqlSchemaCollection>
```

The coordinates can be obtained from the Ensembl_v65.Gencode_v10.ENSG .gene_info file by matching the gene identifier contained in the first column in the matrix file to the geneID attribute of the gene_info file or, in the case of exons, directly by parsing the exon_location field in the matrix file. Since the coordinates of the genes are taken from the Ensembl [83] GRCh37 genes archive, the coordinates are in 1-based coordinate system. The expression values are taken from the correspond matrix file. The remaining attributes are taken from the gene_info file (the name_and_source gene_info attribute is parsed and spliced in three different region attributes; i.e., gene_name, gene_name_source and gene_name_accession). The strand attribute must be converted in a GMQL readable format before adding it to the new file (from "-1" to "-" and from "1" to "+"). The name of the new files matches the pattern [EpigenomeID]_[DataType].gdm, where [DataType] is one of the 7 data types for which the gene expression is available. [DataType] is obtained by removing the "57epigenome" prefix and the indication of the type of expression quantification since the new files contain both (e.g., [DataType] of 57epigenomes.exon.N.nc is exon.nc). It is worth to notice that [DataType] and the region attribute *gene type* are not the same, even if both of them describe the type of regions to which they relate. They can be the same, but in general *gene type* is a more fine-grained classification of the regions function and role inside the genome.

5.2.3 RNA-seq intergenic contigs

In the directory */byDataType/rna/intergenic_contigs/* of *wustl roadmap data repository* [63] also the coordinates of all significant intergenic RNA-seq contigs[7] not overlapping the annotated genes are reported. The intergenic contigs are grouped inside RNAseq_intergenic.tar.gz, a compressed archive containing a bedGraph file [86] for each epigenome subject to RNA-seq experiment. Each bedGraph file lists the intergenic contigs for an epigenome. This data are not imported in GMQL.

Example 5.2.15 First line of the E000.intergenic.bedGraph file, the first file inside RNAseq_intergenic.tar.gz file [87]. The value attribute report the normalized RPKM.

$$\underbrace{chr1}_{\text{chrom}}\ \underbrace{87601}_{\text{chromStart}}\ \underbrace{87800}_{\text{chromEnd}}\ \underbrace{0.514625}_{\text{value}}$$

The RNAseq_intergenic_summary.xls file reports some summary statistics of intergenic contigs.

Example 5.2.16 The RNAseq_intergenic_summary.xls file header and its first line of summary statistics [88].

```
EG Name Number_of_expressed_clusters
    Total_genomic_length_of_clusters Average_length_of_the_cluster
    Average_RPKM_of_the_cluster
E000 Universal_Human_Reference 28216 14760800 523.136 0.895187
```

Also for intergenic contigs data, metadata associated with the epigenome can be obtained from *Google spreadsheet: Metadata and quality control*, Consolidated_EpigenomeIDs_ summary_Table sheet [64] by matching the epigenome identifier extracted from the file name and the sheet attribute *Epigenome ID (EID)*.

5.3 Consolidated epigenomes methylation data

The processed data about methylated region associated with the Roadmap epigenomes can be downloaded from *wustl roadmap data repository* [63] starting from the base directory */byDataType/dnamethylation/*. The directory structure is reported in Figure 5.2. Both whole genome bisulfite sequencing (WGBS) and reduced representation bisulfite sequencing (RRBS) data are placed in an homonymous directory, where a dedicated subdirectory is reserved for the bigWig files containing fractional methylation data and one for total coverage tracks. Only fractional methylation tracks are available for mCRF (Conditional Random Field methylation). The total coverage and fractional methylation were obtained from the number of converted and unconverted reads at each individual CpG site.

[7] In general, a contig is a set of overlapping DNA segments that together represent a consensus region of DNA [84]. In this specific case, the procedure used to obtain contigs is reported at [85]

```
byDataType/dnamethylation/ .................. It is the base directory.
 |__RRBS/ .................................... It contains all the RRBS
 |   |                                         methylation files.
 |   |__FractionalMethylation_bigwig/ ........ It contains all the RRBS
 |   |                                         fractional methylation
 |   |                                         files.
 |   |__ReadCoverage_bigwig/ ................. It contains all the RRBS
 |                                             read coverage files.
 |__WGBS/ .................................... It contains all the WGBS
 |   |                                         methylation files.
 |   |__FractionalMethylation_bigwig/ ........ It contains all the WGBS
 |   |                                         fractional methylation
 |   |                                         files.
 |   |__ReadCoverage_bigwig/ ................. It contains all the WGBS
 |                                             read coverage files.
 |__mCRF/ .................................... It contains all the mCRF
     |                                         methylation files.
     |__FractionalMethylation_bigwig/ ........ It contains all the mCRF
                                               fractional methylation
                                               files.
```

Figure 5.2: Directory structure containing the methylation data of the consolidated epigenomes.

All the files containing methylation data are in bigWig format and match the pattern [EpigenomeID]_[Methylation_type]_[Track_type].bigwig, where [Methylation_type] can assume value *WGBS*, *RRBS* or *mCRF* and [Track_type] value *ReadCoverage* or *FractionalMethylation*. Since the bigWig is a binary format the files can not be directly imported in GMQL.

Each file is associated with an epigenome from which data are taken. Just a small set of epigenomes has methylation data attached. A methylation file can be associated with the corresponding epigenome metadata by matching the [EpigenomeID] in the file name and the *Epigenome ID (EID)* attribute available through *Google spreadsheet: Metadata and quality control*, Consolidated_EpigenomeIDs_ summary_Table sheet [64].

5.4 Differentially Methylated Regions (DMRs)

The processed data about DMRs associated with the Roadmap epigenomes can be downloaded from *wustl roadmap data repository* [63] starting from the base directory */byDataType/dnamethylation/DMRs/*. The DMRs are obtained by identifying regions where the methylation level is differentially expressed across different epigenomes. The directory structure containing DRMs calculated starting from WGBS and RRBS data, both in tab-delimited format and bigWig format, is reported in Figure 5.3. Only the tab-delimited version can be imported directly into GMQL. Each tab-delimited compressed file matches the name pattern [EpigenomeID]_[Methylation_type]_DMRs_v2.bed.gz and is paired with the corresponding index file. The bigWig files match the name pattern [EpigenomeID]_[Methylation_type]_DMRs_-v2.bigWig.

```
byDataType/dnamethylation/DMRs/ ........... It is the base directory.
 ├─RRBS_bed/ ............................... It contains all the DMRs bed
 │                                           files in compressed archives
 │                                           and the associated tbi index
 │                                           files obtained from RRBS
 │                                           data.
 ├─RRBS_bigwig/ ............................ It contains all the DMRs
 │                                           bigWig files obtained from
 │                                           RRBS data.
 ├─WGBS_bed/ ............................... It contains all the DMRs bed
 │                                           files in compressed archives
 │                                           and the associated tbi index
 │                                           files obtained from WGBS
 │                                           data.
 └─WGBS_bigwig/ ............................ It contains all the DMRs
                                             bigWig files obtained from
                                             WGBS data.
```

Figure 5.3: Directory structure containing the DMRs data of the consolidated epigenomes.

Example 5.4.17 First line of the E001_RRBS_DMRs_v2.bed file [90].

$$\underbrace{chr1}_{\text{chrom}}\ \underbrace{10589}_{\text{chromStart}}\ \underbrace{10620}_{\text{chromEnd}}\ \underbrace{0.842105263158}_{\text{score}}\underbrace{0.842105263158}_{\text{score}}$$

A DMR is obtained by combining all differentially methylated sites (DMSs) within a maximum distance of 250 bp from one another into a single DMR (DMRs with less than 3 DMSs have been excluded) [89]. For each DMR in each sample, the score attribute is computed by averaging the methylation level weighted by the number of reads overlapping the DMR.

The DMRs attribute score is duplicated. Before importing the DMRs files in GMQL the duplicated attribute is removed. Instead the region attribute strand is added with value "." for each region since required by GMQL.

The GMQL schema is:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gmqlSchemaCollection name="DMR"
    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">stand</field>
<field type="DOUBLE">score</field>
</gmqlSchema>
</gmqlSchemaCollection>
```

The row data used to calculate the DMRs are submitted to the Roadmap Consortium in BED or WIG format and their coordinates are in 0-based coordinate system.

5.5 Samples name convention

The Roadmap Epigenomic source provides files with a naming convention that differs between datasets. In particular, in some datasets, different parts of the file names are separated with the character ".", but in other datasets they are separated by the character "_". The use of "." inside the file names clashes with the requirements of GMQL that needs sample names without ".". The only "." character in a file path must be the one separating the file name from the file extension. So, with the purpose to uniform the names of the samples across the datasets and to obtain GMQL-valid sample names, during the transformation process all the "." in the file names are replaced by the character "_", with the exception of the "." before the file extension. Another difference in file names among different datasets is the use of "-" to separate the epigenomeID prefix from the rest of the file name in some cases, while in others cases it is used the "_". Again, to uniform the name convention across all the datasets, we replace also the character "-" with an underscore during the transformation process.

5.6 Metadata

All the Roadmap Epigenomics region files are related to a consolidated epigenome specified in the name of the files. The metadata associated with an epigenome can be found in *Google spreadsheet: Metadata and quality control*, Consolidated_EpigenomeIDs_-summary_Table sheet [64]. The epigenome metadata of a file can be obtained by matching the [EpigenomeID] in the file name with the *EpigenomeID (EID)* attribute of the table in the sheet. Some DMR files are associated with the epigenome E064, missing from the datasheet since it has been excluded from the release due to data quality issues. For this reason the epigenome metadata associated with these files are unavailable, with exception of the *comments* and *Epigenome ID (EID)* attributes. Some RNA expression files are associated with the epigenome E000. It is not an actual epigenome, but a reference sample designed to be used as a reference for expression profiling experiments. For this reason, the epigenome is not present inside the datasheet and the available metadata associated with the epigenome are a subset of those available for the actual epigenomes.

The peaks files are also related to an experiment, identified by an epigenome identifier and the feature (mark) involved in the experiment. Experiment metadata can be downloaded from *Google spreadsheet: Metadata and quality control*, Consolidated_EpigenomeIDs_QC sheet [64]. They can be associated with a file by matching both [EpigenomeID] and [Marker] in the file name with *EID* and *MARK* sheet table attributes, respectively. It is important to note that nor the epigenome data alone nor the union between epigenome and experiment metadata, in the case of peaks data, are unique for files belonging to different dataset. This implies that two files referring to the same EID and marker but different format have the same metadata associated. Similarly, two files referring to the the same epigenome but inside the same dataset have equal metadata. To disambiguate the files referring to the same epigenome and marker, but with different formats, some manually curated attributes, depending from the dataset and file, are added to all the metadata files during their importing in GMQL (for details see Section 5.6.3). In addition, also some manually curated metadata in common to all files are added.

The three different types of metadata (referring to epigenome, associated with experiment and manually curated) are distinguished and grouped by adding the prefix "epi___" if they are epigenome metadata, "exp___" if they are experiment metadata or "manually_curated___" if they are manually curated. Inside of each group they are sorted in alphanumerical order, and groups are alphabetically sorted too.

5.6.1 Epigenome metadata

A complete description of all the epigenome metadata available through the Consolidated_EpigenomeIDs_summary_Table is reported in the Section 4.2.1.1. From all the metadata available, only the attributes with a value for the corresponding epigenome of the region file are reported in the metadata files. Also metadata referring to a specific marker different from the marker of the associated region file are omitted. If the region file is associated with no marker, then none of the marker-specific metadata is reported in the corresponding metadata file.

GMQL requires alphanumeric strings as name of metadata attributes. Since the name of the attributes provided through the datasheet contains also special characters such as spaces, parenthesis and mathematical symbols, the name of the metadata attribute is transformed in an equivalent alphanumeric string by applying the following substitutions:

1. "%" character is replaced by "_perc_";
2. "+" character is replaced by "_plus_";
3. "/" character is replaced by "_or_";
4. "(" and ")" characters are replaced by an underscore;
5. all spaces are replaced by underscores;
6. multiple consecutive underscores are replaced by a single underscore (with the exception of those separating our added prefixes).
7. all the remaining non-alphanumeric character are eliminated;
8. underscores at the start or at the end of the string are eliminated.

After that an alphanumeric metadata attribute name is obtained.

Example 5.6.18 An epigenome attribute name before and after the transformation in a alphanumeric string.

"AGE (Post Birth in YEARS/ Fetal in GESTATIONAL WEEKS/CELL LINE CL) " becomes "AGE_Post_Birth_in_YEARS_or_Fetal_in_GESTATIONAL__WEEKS_or_CELL_LINE_CL".

The metadata "AGE" value is also modified before importing in GMQL. This attribute value can originally be "CL", "unknown", a number expressed in years, or a number expressed in months. If the original value is "CL", the metadata is omitted, and a new attribute named *manually_curated__life_stage* with value "CL" is added; conversely, if the original value is "unknown", it is reported as it is, while if it is a number, it is always expressed in months.

Some additional transformations are applied to the group of metadata associated with the donors of biological samples used to build a particular epigenome. The donor metadata provided by the datasheet, after the preliminary transformations previously described, are:

- epi__ethnicity
- epi__age_weeks
- epi__single_donor_or_composite
- epi__donor_or_sample_alias

epi__single_donor_or_composite can be "SD" or "C". If the value of *epi__single_donor_or_composite* is "SD", then the attribute *epi__age_weeks* and *epi__ethnicity* have a single value, otherwise they have a list of semicolon-separated values. *epi__donor_or_sample_alias* can be a single value or a list of values if the value of *epi__single_donor_or_composite* is "SD", since more than one sample can be withdraw by the same donor. Instead, *epi__donor_or_sample_alias* can only be a list of values if *epi__single_donor_or_composite* is "C". So, we have 3 different transformations possible, depending on the value of *epi__single_donor_or_composite* and *epi__donor_or_sample_alias*:

1. If the *epi__single_donor_or_composite* value is equal to "SD" and *epi__donor_or_sample_alias* has a single value, then the new attributes *epi__donor_id* and *epi__sample_alias* are added, both with the same value as *epi__donor_or_sample_alias*, and *epi__donor_or_sample_alias* is deleted.

Example 5.6.19 Transformation performed when the *epi__single_donor_or_composite* value is equal to "SD" and *epi__donor_or_sample_alias* has a single value.

```
epi__single_donor_or_composite SD
epi__donor_or_sample_alias 113
epi__ethnicity Caucasian
epi__age_weeks 20
```

becomes:

```
epi__donor_id 113
epi__sample_alias 113
epi__single_donor_or_composite SD
epi__ethnicity Caucasian
```

```
epi__age_weeks 20
```

2. If the *epi__single_donor_or_composite* value is "SD" and *epi__donor_or_sample_alias* has a multiple value, then a new *epi__sample_alias__X* attribute for each element in the *epi__donor_or_sample_alias* list is added with value equal to the value of element in position X of the *epi__donor_or_sample_alias* list. The X corresponds to the order in the list. The *epi__donor_id* attribute with the same value as the first sample_alias (epi__sample_alias__1) is added. In case *epi__age_weeks* has multiple values, only the first value is retained. The old attribute *epi__donor_or_sample_alias* is deleted.

Example 5.6.20 Transformation performed when *epi__single_donor_or_composite* value is "SD" and *epi__donor_or_sample_alias* has a multiple value.

```
epi__single_donor_or_composite SD
epi__donor_or_sample_alias NPC-03;NPC-04;NPC-05;NPC-06
epi__ethnicity Caucasian
epi__age_weeks 20
```

becomes:

```
epi__single_donor_or_composite SD
epi__sample_alias__1 NPC-03
epi__sample_alias__2 NPC-04
epi__sample_alias__3 NPC-05
epi__sample_alias__4 NPC-06
epi__donor_id NPC-03
epi__ethnicity Caucasian
epi__age_weeks 20
```

3. When the *epi__single_donor_or_composite* value is "C", an *epi__donor_id__X* attribute and an *epi__sample_alias__X* attribute for each element of the *epi__single_donor_or_composite* list with value equal to the value of the element in position X in the list are added. Also multiple *epi__ethnicity__X* attributes are added, one for each element in *epi__ethnicity* attribute. If there is only one value in *epi__ethnicity*, then that value is replicated in each of the *epi__ethnicity__X* new attributes. Multiple *epi__age_weeks__X* attributes are added, one for each element in the *epi__age_weeks* attribute. A *manually_curated__life_stage__X* attribute for each element in the *manually_curated__life_stage* attribute is added. The X corresponds to the order in the *epi__donor_or_sample_alias* list. The old attributes *epi__donor_or_sample_alias*, *epi__ethnicity*, *epi__age_weeks*, and *manually_curated__life_stage* are deleted.

Example 5.6.21 Transformation performed when *epi___single__donor__or__composite* value is "C".

```
epi__single_donor_or_composite C
epi__donor_or_sample_alias STL001;STL003
epi__ethnicity Caucasian/African American, Caucasian
epi__age_weeks 156, 1768
manually_curated__life_stage born
```

becomes:

```
epi__single_donor_or_composite C
epi__donor_id__1 STL001
epi__donor_id__2 STL003
epi__sample_alias__1 STL001
epi__sample_alias__2 STL003
epi__ethnicity__1 Caucasian/African American
epi__ethnicity__2 Caucasian
epi__age_weeks__1 156
epi__age_weeks__2 1768
manually_curated__life_stage__1 born
manually_curated__life_stage__2 born
```

5.6.2 Experiment metadata

All the experiment metadata available through Consolidated_EpigenomeIDs_QC sheet are listed in Section 4.2.1.2. As it happens for the epigenome metadata, also the experiment metadata are reported in the metadata file associated with a particular epigenome and marker imported in GMQL only if their value is not empty for that epigenome and marker. Some experiment metadata are a duplicate of epigenome metadata (i.e., "EID", "E-Mnemonic", "Standardised epigenome name", "Epigenome name (from EDACC Release 9 directory)", "NSC (Signal to noise)", "RSC (Phantom Peak)", "NREADS"). This metadata are reported only among the epigenome metadata group and omitted in the experiment group of metadata. Experiment metadata name must be transformed in alphanumeric string too, so the same transformations used for epigenome metadata name (see Section 5.6.1) are applied also to experiment metadata name. No value modification is required instead.

5.6.3 Manually curated metadata

The following manually curated metadata are added to all the metadata files:

- *manually_curated___assembly*: it is the version of the human genome to which the associated genomic regions are aligned (hg19 for all the Roadmap Epigenomics region files);
- *manually_curated___data_type*: it describes the type of data contained in the

associated region file (i.e., ChIP-seq, DNase, RNA-seq, DMR);

- *manually_curated__data_url*: it is a list of semicolon-separated addresses of the original files form which the region file referred by the metadata is generated;

- *manually_curated__feature*: it is the feature described in the associated region file (i.e., open chromatin, histone modification, gene expression, DNA methylation);

- *manually_curated__file_id*: it is the unique identifier of the region file referred by the metadata file;

- *manually_curated__file_size*: it is the size in byte of the region file referred by the metadata (after the transformation process);

- *manually_curated__format*: it is the format of the region file (i.e., narrowPeak, broadPeak, gappedPeak, BED);

- *manually_curated__metadata_url*: it is the address of the source from which the automatically generated metadata are taken (in the specific case it is the url of the datasheet).

To all the peak files metadata the peak_caller information is also added:

- *manually_curated__peak_caller*: it indicates the peak caller used to generate the data (i.e., MACS2, HOTSPOT).

In addition, to all the files generated using HOTSPOT peak caller, two additional metadata are added:

- *manually_curated__fdr_threshold*: it reports if some accuracy threshold is applied during peak call phase (i.e., none, 0.01);

- *manually_curated__region_type*: it indicates the type of regions inside the BED files (i.e., narrow, broad).

In order to disambiguate the RNA expression files, a new metadata attribute *manually_curated__rna_expression_region* is added to the corresponding metadata files to indicate the type of regions contained in the region file (i.e., protein coding genes, ribosomal genes, protein coding exons, protein non coding exons, ribosomial gene exons, intronic protein coding RNA elements, non-coding RNAs).

With the purpose to disambiguate the DMR files, instead, the attribute *manually_curated__methylation_technique* is added to the corresponding metadata files to indicate the technique used to identify the methylation regions contained in the file (i.e., RRBS, WGBS).

Finally, the *manually_curated__life_stage* metadata attribute is added to all the metadata files referring to a epigenome with an "AGE" metadata attribute. This attribute assumes value "cell line" if the "AGE" metadata attribute value is "CL", value "born" if the "AGE" was originally expressed in years (then converted and expressed in weeks in final metadata), and "fetal" if the "AGE" is expressed in gestational weeks.

5.6.4 Metadata post-processing modifications

After the generation of the metadata associated with each region file, they are subjected to a final modification process before being imported in the GDM repository. During this process some metadata attribute names are modified:

- AGE_Post_Birth_in_YEARS_or_Fetal_in_GESTATIONAL_WEEKS_or_CELL_LINE_CL → AGE_WEEKS;
- Auto_Use_Train_Core → Auto_Use_Train;
- Epigenome_ID_EID → Epigenome_ID;
- Epigenome_name_from_EDACC_Release_9_directory → Epigenome_name_EDACC_9;
- LAB_Based_on_5_core_histone_marks_only → LAB;
- Manual_Use_Train_Core → Manual_Use_Train;
- SEX_Male_Female_Mixed_Unknown → SEX;
- Single_Donor_SD_or_Composite_C → Single_Donor_or_Composite;
- Perc_of_Impute_top_1_perc_in_Observed_Top_1_perc → Perc_of_Imputed_top_1_perc_in_Observed_Top_1_perc.

After the replacement, the attribute name is converted in lower case letter. This occurs even if no replacement is performed, so at the end of the post-processing phase all the metadata attribute names are composed by lower case letters.

Chapter 6

Modules implementation

Once the datasets to import has been identified and the transformation to get each data and metadata files compatible with GDM has been defined, we implement the actual modules to download and transform the data. Since the data sources use different technologies and standards to share their data and the data available are extremely heterogeneous among sources (but also inside the same source), we can not use the standard downloader and transformer modules provided by GMQL-Importer, and an ad hoc downloader and transformer module must be implemented. So, we developed the *RoadmapDownloader* module to get the Roadmap Epigenomics data required by the user through the configuration file from *wustl roadmap data repository* [63]. The downloaded files are placed in a local repository managed through a local database by GMQL-Importer. The database keeps track of each file, its status, and its stage of elaboration. The files are organized by sources and datasets. Then, we developed the *RoadmapTransformer* module to uniform the data to the format required by the GDM and the GMQL server, the target server on which we want to import the data. The upload phase is quite standard, once the data are in the required format, and depends on the data destination. To upload the Roadmap Epigenomics Project (REP) data into GMQL server we used the already available GMQLLoader module. We pursue the overall implementation with the objective of keeping the code as general and configurable as possible to allow future expansions and to ensure code reusability, along with providing the most configuration options possible to the user through XML configuration file.

6.1 Remote files downloading: RoadmapDownloader

The `RoadmapDownloader` is a Scala class that extends the class `GMQLDownloader` overriding the methods `download` and `downloadFailedFiles`. These two methods are used by the GMQL-Importer main program to interact with the `RoadmapDownloader` module. The `download` method is called by the `runGMQLImporter` method in all the standard executions where the download is enabled by the user; and it downloads region and associated metadata files. `downloadFailedFiles` is executed when the user launches the GMQL-Importer with input parameter "-retry" and at the end of the `download` method execution. `downloadFailedFiles` retries to download all the remote files that previously failed to be downloaded. Figure 6.1 graphically represents the execution of the `RoadmapDownloader` module and the interaction between the main methods involved.

The `download` method creates a folder, if not already existing, for each dataset specified in the configuration file, in which the downloaded files are placed. After that, the status of all the prior files present in each dataset is set to "COMPARE". The "COMPARE" status implies that the module must check if the version of the file in the GMQL-Importer database is the same as the version of the file present on the remote source. At this point, the `recursiveDownload` and `downloadMetadata` methods are called. The first is in charge to download all the files matching the name and directory pattern in the *wustl roadmap data repository* [63] HTTP directory and place them in the local directory corresponding to the dataset to which the files belong. The latter one downloads the metadata associated with the REP data. After the execution of `recursiveDownload` and `downloadMetadata`, also the `downloadInfo` method is executed. It downloads additional files specified in the configuration file by user. These files contain informations used during the transformation phase of the files downloaded by `recursiveDownload` method. If no additional information is required to carry out the transformation process, then no additional file is downloaded by `downloadInfo`. During the downloading phase, all the files in the source missing in the local database or modified since the last download are added to the local database. If the download succeeds, the file in the local database is marked as "SUCCESS", otherwise as "FAILED". At the end of the downloading phase, the status of all files still marked as "COMPARE" is changed to "OUTDATED". This allows to detect files in the local database, but removed in the source.

For each dataset, the method `downloadFailedFiles` detects the local files in the database with status "FAILED". For each failed file, its URL is retrieved from the database and a new attempt to download it from the source using the URL in the database is made. If the new attempt to download the file ends with success, then the file is marked as UPDATED, else its status is left as FAILED. `downloadFailedFiles` supports also parallel execution. So, if parallel execution is enabled by the user, for each dataset a thread performing the operation described above is started. The pool of threads is synchronized before the end of the method to grant that all the threads have terminated the execution before new operations are started.

6.1.1 Region files downloading: recursiveDownload

As stated by the method name, `recursiveDownload` is an indirectly recursive function. Since the REP source [63] is a HTTP-based remote directory server, each directory is an HTTP page containing links to files or subdirectory. `recursiveDownload` receives in input an URL corresponding to a remote directory and produces a Jsoup[1] `Document` object from the HTML directory page thanks to Jsoup HTML pages parser [91]. Though a call to `checkFolderForDownloads` method, the obtained Jsoup `Document` is used to check if the remote directory contains some files that match the criteria specified by the user in the configuration file and download them. The following call to the method `downloadSubFolders` detects all the subdirectories in the directory received as input, exploiting the information present in the Jsoup `Document` and, for each of them, the method `recursiveDownload` is called again.

[1] Jsoup is a Java library for working with HTML. It provides a very convenient API for extracting and manipulating data [91].

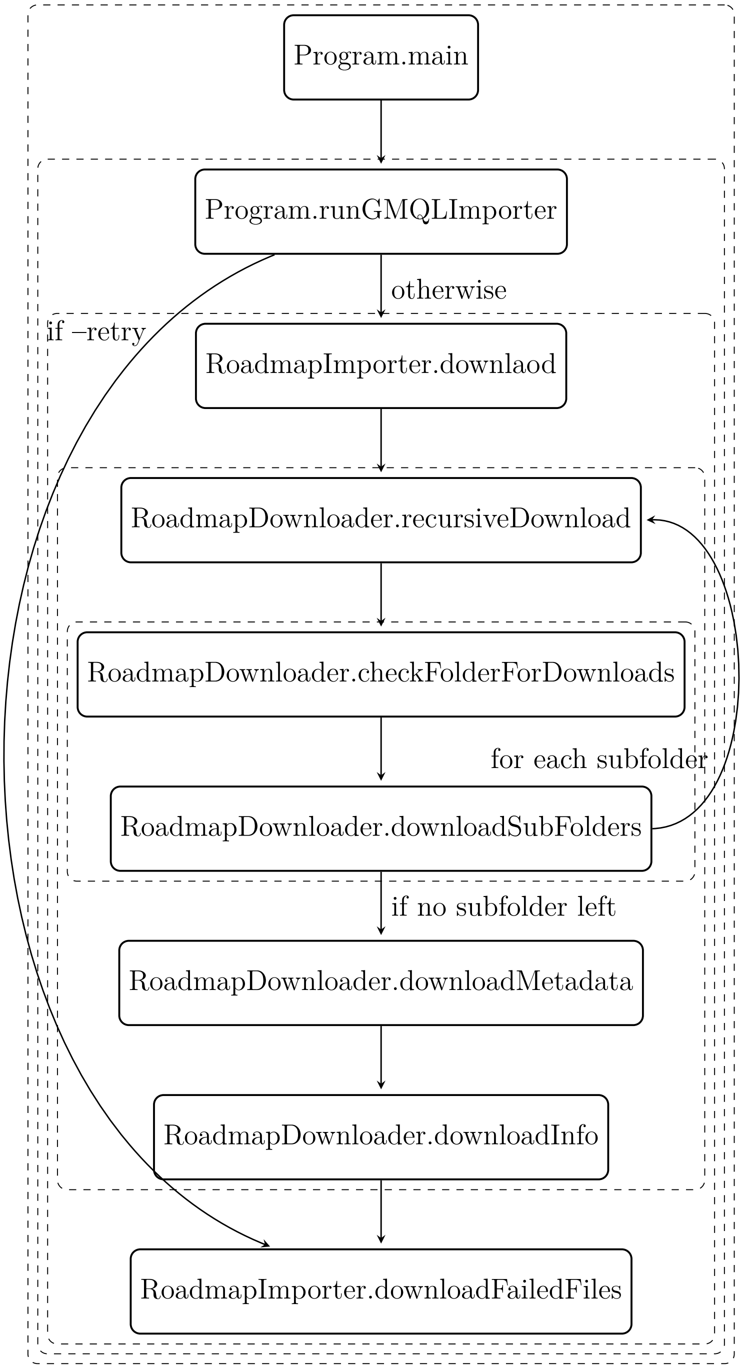

Program.main
Program.runGMQLImporter
otherwise
if –retry
RoadmapImporter.downlaod
RoadmapDownloader.recursiveDownload
RoadmapDownloader.checkFolderForDownloads
for each subfolder
RoadmapDownloader.downloadSubFolders
if no subfolder left
RoadmapDownloader.downloadMetadata
RoadmapDownloader.downloadInfo
RoadmapImporter.downloadFailedFiles

Figure 6.1: Order and conditions (if any) of execution of the main methods involved in the *RoadmapDownloader* module. Each rectangle represents a method and each dashed box represents the context in which the methods are executed. Methods that share a context are executed in sequence by the same parent method. Methods connected by an arrow but placed inside different contexts are related by a caller-called relationship. When a context ends, the caller method continues its execution until a new context starts (a new method is called) or the function terminates. Also between methods in the same context, the caller methods can execute code. The graph is designed to provide an intuitive and general idea of the `RoadmapDownload` work-flow, not an exhaustive description of the GMQL-Importer code.

6.1.1.1 checkFolderForDownloads

`checkFolderForDownloads` is a method that receives in input a remote directory (URL and Jsoup `Document`) and downloads all the files that match the pattern specified by the user contained in the remote directory. First, for each dataset, it checks if the remote directory path matches the regular expression provided in the configuration file through the *folder_regex* parameter. If not, nothing is done. If it matches, the Download folder inside the dataset folder is created, if it does not already exist, and all the link tags present in the directory HTML page are extracted using the Jsoup `Document` object. The value of the href attribute of each link element is matched against the pattern defined in the configuration file through the *files_regex* attribute to check if it is a file that belong to the dataset. If the link matches the pattern, then the date of upload and the size of the file are extracted from the HTML page of the directory. Date and size of the file are used to check if the local file is updated. If the local file is outdated, then the file is downloaded. If the downloading process succeeds, then the file is marked as "UPDATE", else it is marked as "FAILED".

6.1.1.2 downloadSubFolders

`downloadSubFolders` is a method that receives in input a remote directory (URL and Jsoup `Document`) and detects all the subdirectories contained in the input directory by extracting each of the link elements from the HTML directory page using the Jsoup `Document` object and checking that it actually corresponds to a subdirectory. Links to the parent directory and home directory are excluded. For each directory identified, the method `recursiveDownload` is called (indirect recursive call). The recursive call chain stops when the bottom of a branch in the directory tree is reached because it contains only files (leafs) and no more directory (branch).

6.1.2 Metadata files downloading: downloadMetadata

The `downloadMetadata` method downloads the metadata available through Google Spreadsheet and puts them in each dataset directory. In order to achieve this, the Google Spreadsheet API must be used and an OAuth2[2] authentication is required. `downloadMetadata` performs the authentication and gains access to Google Spreadsheet API by taking advantage of the `OAuth` class. Once the access to API services has been granted, for each dataset:

1. the *spreadsheet_url* parameter is read from the configuration file and the identifier of the spreadsheet is extracted;
2. the connection to the spreadsheet is established through the API service using the spreadsheet ID;
3. the destination folder where to place the file is created if missing;

[2] OAuth 2.0 is the industry-standard protocol for authorization [92]. The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf [93]. Google APIs use the OAuth 2.0 protocol for authentication and authorization [94].

4. all the non-hidden sheets contained in the spreadsheet are downloaded in CSV format and placed in the dataset download local folder.

The download of a single sheet is managed by the class `CSVDownload`.

The CSV files are not registered in the GMQL-Importer database and no check is performed on version discrepancy between the remote file and a local previously downloaded one. This is because there is no hash, date or size associated with remote sheets to use in the version comparison. Instead, the files are downloaded each time and any local file already in the directory is overwritten. This is justified by the fact that the number of files and their size is small (some units of files large a few hundreds of Kbytes).

6.1.3 Additional files downloading: downloadInfo

In some cases, during the transformation of the Roadmap Epigenomics data, some additional information may be required to obtain region files in a format compatible with GMQL. These information can be potentially contained in files placed outside the source in which the other REP data are, or they could require a different download method. For such reasons, the `downloadInfo` method has been introduced. It is in charge to download all the additional files specified by the user in the configuration file, which are required to perform the transformation. For each dataset `downloadInfo` downloads the additional files for which the user has specified the URL through the parameters *info_url* in the configuration file.

6.2 Data transformation: RoadmapTransformer

`RoadmapTransformer` is a Scala class that extends the `GMQLTransformer` class overriding the methods `transform` and `getCandidateNames`. They are the methods used by GMQL-Importer to interact with the `RoadmapTransformer` module. The main methods involved in the execution of the `RoadmapImporter` module and their order and conditions of execution are schematically represented in Figure 6.2.

`getCandidateNames` receives in input the name of a downloaded file ready to be transformed and returns the list of names of the files that are generated during the transformation phase (from a single file obtained from the source more than one GMQL compliant region files can be generated). `getCandidateNames` detects the type of file to transform by looking the input file name pattern first, and then it generates the list of candidate names accordingly. For the most of the REP files only one file is obtained during transformation phase and the name of the transformed file is generated by simply removing the .gz extension at the end of the original file name. Moreover, if the input name contains the character "-", it is replaced by "_". This allows to obtain uniform candidate name across all the datasets. The files containing RNA expression quantification data, instead, are tabular files where each row corresponds to a genome region and each column is associated with an epigenome. Since GMQL requires files containing a set of regions associated with a single epigenome, from each tabular file a new file for each column must be generated during transformation. Plus, RPKM expression quantification data are merged with read counts (N) expression quantification data referring the same region. So, for each input name of N data tables (nothing is done for input name of RPKM files since their data are merged inside the file generated starting from N data file), the corresponding files (both N and RPKM) are unzipped and placed in the Transformations folder, so that the list of epigenomes can be extracted from the header of the file. For each epigenome the corresponding file candidate name is generated by adding the epigenome identifier to the input name without the indication about number of epigenome and type of expression qualification (the new file contains regions referring only one epigenome and both N and RPKM expression quantifications) and placed in the returned list of candidate names.

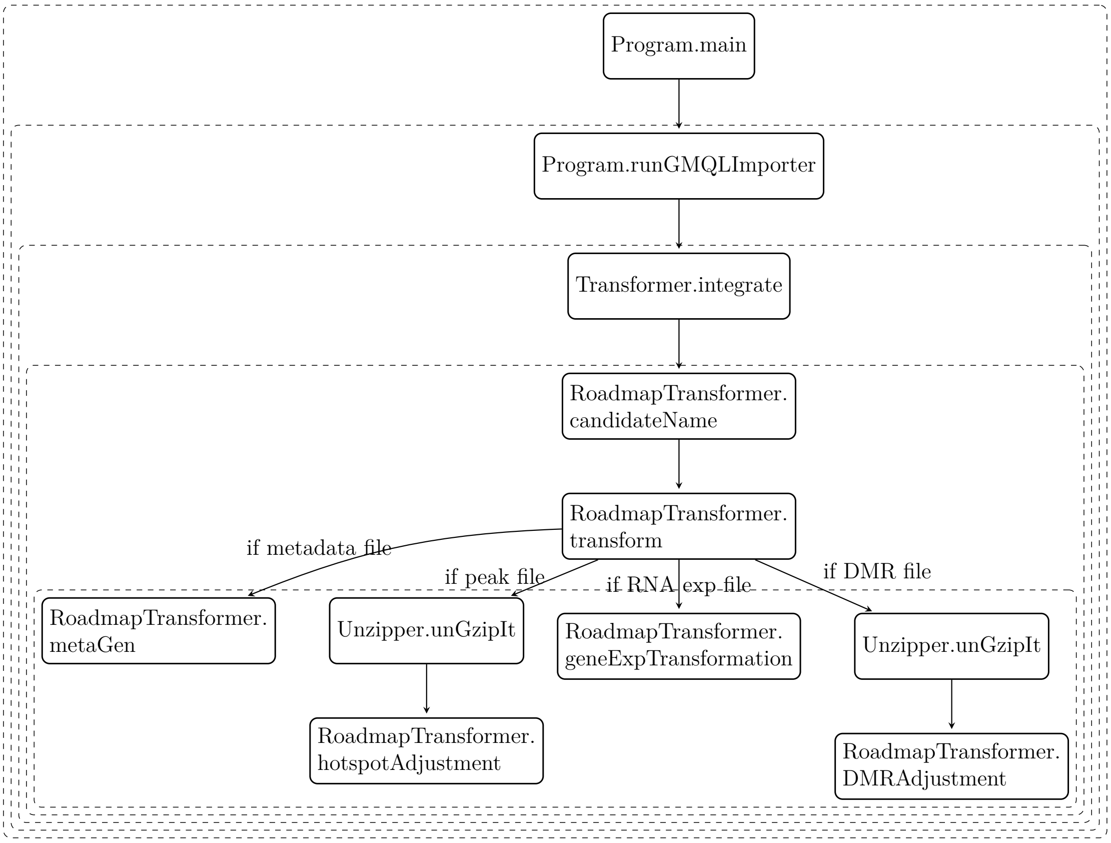

Program.main
Program.runGMQLImporter
Transformer.integrate
RoadmapTransformer. candidateName
RoadmapTransformer. transform
if metadata file
if peak file
if RNA exp file
if DMR file
RoadmapTransformer. metaGen
Unzipper.unGzipIt
RoadmapTransformer. geneExpTransformation
Unzipper.unGzipIt
RoadmapTransformer. hotspotAdjustment
RoadmapTransformer. DMRAdjustment

Figure 6.2: Order and conditions of execution of the main methods involved in the `RoadmapTransformer` module. Each rectangle represents a method and each dashed box represents the context in which the methods are executed. Methods that share a context are executed in sequence by the same parent method. Methods connected by an arrow but placed inside different contexts are related by a caller-called relationship. When a context ends, the caller method continues its execution until a new context starts (a new method is called) or the function terminates. Also between methods in the same context, the caller methods can execute code. The graph is designed to provide an intuitive and general idea of the `RoadmapTransformer` work-flow, not and exhaustive description of the GMQL-Importer code.

`transform` function receives in input the name and path of a file to transform and the name and path of the transformed file. `transform` performs the transformation process over the file to transform and save the result with name and path received as input. The transformation process depends both on the type of the file to transform and the file to obtain after the transformation, so the first thing to do is to detect them starting from their name. All the files downloaded from the REP repository are provided inside a GZip compressed archive (CSV and additional information files containing the metadata are not registered in the GMQL-Importer database, so the `Transform` object does not detect them when searches for files); so, if the name of the file to transform does not terminate with ".gz" a warning is shown and no transformation is performed. Instead, the file to obtain at the end of the transformation process can be either a metadata file, a peak file, a DMRs file, or a RNA expression quantification file. If the file to generate is a metadata file, then the `metaGen` method is called to generate the required metadata file. If the file to transform is a peak file, it is extracted from the GZip archive by calling the `unGzipIt` method provided by the `unzipper` object. Only if the file to transform contains regions spotted using HOTSPOT peak caller, the `hotspotAdjustment` is called with the purpose to align the regions to the same schema by adding any missing region attributes. If the file to transform is a peak file, it is extracted from the GZip archive and the methods `DMRAdjustment` is called to remove a duplicated attribute and add a missing one. Finally, if the file to transform is an RNA expression qualification file, it is already extracted from the archive by `getCandidateNames` method and the `geneExpTransformation` is directly called with the objective to generate a region file associated with an epigenome starting from two table files containing the read counting and RPKM RNA expression quantifications and the additional information file containing the coordinates of the regions.

6.2.1 .meta files generation: metaGen

Metadata are downloaded as CSV tables from *Google spreadsheet: Metadata and quality control* [64]. Since GMQL requires tab-separated key-value pairs files, the metadata associated with each file must be detected and extracted from the tables. To achieve this, a CSV parser is required. We choose to use the scala-csv package [95] available through Maven Repository, a simple but effective CSV parser for Scala. It provides also fast I/O and manipulation instructions to operate with the CSV files.

First, the `metaGen` method loads in memory the CSV files containing the useful metadata (jul2013.roadmapData.qc_Consolidated_EpigenomeIDs_summary_-Table.csv and jul2013.roadmapData.qc_Consolidated_EpigenomeIDs_QC.csv in the case of peak region data) using the reader object provided by the parser. jul2013.roadmapData.qc_Consolidated_EpigenomeIDs_summary_Table.csv has 3 rows as header, but only the first two are useful to our purpose. So, the first two lines are read and merged, concatenating the terms (using "___") coming from the first and second line of the same column. A new single line header containing the metadata attribute name (the key in the .meta files) is obtained and saved in a temporary file. The content of the roadmapData.qc_Consolidated_Epige-nomeIDs_QC.csv is copied in the temporary file. The new temporary CSV file now contains all the data coming from roadmapData.qc_Consolidated_Epigeno-meIDs_QC.csv plus the new header. Using the tools provided by scala-csv, the temporary file and qc_Consolidated_EpigenomeIDs_QC are placed in memory, each one inside a `List` of `Maps`. Each `Map` has the file first line (header) as keys and the content of a line (starting from the second) as a value. A value is bounded to a key if it is in the same column. In other words, each attribute name is associated with the attribute value. Each element of the list corresponds to a line of the CSV file, except the header. This data structure is very convenient for us because we have converted the metadata from tabular form to key-value pairs, that is what we need in the .meta file. Now, the marker used to obtain the region in the file, the epigenome identifier and the format are extracted from the file name received as input by `metaGen`. The first two are used to select only the related metadata, the third is used to add a manually curated additional metadata. From the first `List` (containing epigenome metadata from the temporary file) only the line with attribute Epigenome ID (EID) equal to the one extracted by the file name is written to the .meta file. From the second `List` (containing experiment metadata from qc_Consolidated_EpigenomeIDs_QC) only the line with attribute Epigenome ID (EID) and mark equal to the one extracted by the file name is added to the .meta file. Epigenome metadata can be associated with every REP file since any of them is related to an epigenome identified by the epigenomeID contained in the name file. Some files are related to epigenomes not listed in the CSV tables (i.e. E000, E060, E064). In these cases the greatest possible number of epigenome metadata are inferred from the file name and/or hard-coded. Experiment metadata, instead, are associated only with the ChIP-seq and DNase region files because they are the only ones realated to a mark listed in the CSV tables. The prefix "epi___" is added to the key of metadata coming from the first list, instead to those coming from the second list the prefix "exp___" is added. This allows to easily distinguish the origin of the metadata also in the final .meta file. The actual writing of the selected maps (rows) is performed by the method `mapToFile`. Before calling the method `mapToFile`, all the useless metadata are filtered out (metadata associated with a marker not related to the region file) and the duplicated metadata are removed from the experiment metadata (but maintained in the experiment metadata group).

Once the mandatory epigenome metadata and the optional experiment metadata are written on the new metadata, the manually curated metadata are generated as key-value tuples and added to a sorted data structure (`TreeSet`). Before all the file type specific metadata are added to the structure, and then all the metadata in common to all the files independently on their type are added, including the additional metadata received as input by `metaGen`. When all the manually curated data are inserted in the sorted data structure, they are written in the metadata file as tab-separated name-value pairs, adding the prefix "manually_curated___" to the attribute name.

At the end of the execution of the `metaGen` method, a .meta file is created inside the *Transformations* folder of the dataset directory with name equal to the associated regions file name plus the .meta attribute. Metadata are grouped by prefix and ordered by key alphabetical order inside each group. Groups are in alphabetical order too.

6.2.1.1 mapToFile

`mapToFile` is a simple method in charge to write, using a given `Writer` object, all the key-value pairs in a tab-separated fashion. `mapToFile` writes a key-value pair only if the value is not "NA" or "N/A" and if the key is not included in the skip list received in input. Before to be written, the key of an attribute is transformed in a valid alphanumeric key. If a prefix or a suffix is received as input, they are added to the key, separated by "___". In addition, some functions associated with a regular expression can be passed (inside a `List` of `Regex`-funtion tuples) to `mapToFile`. Before writing a key-value pair, the key is matched with the regular expressions and if a match is positive the related function is executed. These functions can modify both the value and the key before writing them on the metadata file. In the actual implementation this system is used to correct the "AGE" epigenome metadata attribute value by passing to `mapToFile` the `ageCorrector` methods, but this system is designed to allow easy future expansions. If it is required to modify any other metadata attribute value, it is sufficient to develop a specific method and pass it to `mapToFile` along with the regular expression to define on which key-value pairs applying the function. The introduction of new value-altering methods is fully modular and does not require to change the `mapToFile`. It also grants more generality and re-usability of the `mapToFile` methods, also in other GMQL-Importer modules, since the implementation is fully independent on the REP metadata.

6.2.2 RNA expression region files generation: geneExpTransformation

`geneExpTransformation` is a method that receives in input the name and path of the file to generate and the path of the information file. The information file is loaded in memory and an iterator over the lines (rows) of both the read count table and the associated RPKM table is generated. The paths of the table files are the same as that of the file to generate since they had been already extracted from the archive and placed in the *Transformations* folder of the dataset by `getCandidateNames`. The names of the table files are obtained from the name of the file to generate. The iterator is obtained by using the Scala-csv library properly configured to parse TSV files instead of CSV. The iterators return a `Map` object containing key-value tuples, where a key is the name of a column obtained from the header and the value is the content of the key column for the line indexed by the iterator. The epigenome associated with the data to generate is extracted from its name. The iterators are used to get, from the N and RPKM tables, the RNA expression quantification corresponding to the epigenome of the file to generate, by using the epigenome identifier as key to retrieve the corresponding value from the map of each line. The regions coordinates are obtained by extracting from the row maps the column *exon_location* if available, or using the *gene_id* to get the coordinates from the additional info file if *exon_location* is not available. The *gene_id* info is used also to retrieve from the additional information file also all the other region attributes. Coordinates of the region, N and RPKM RNA expression quantifications and all the other region attributes associated with a region are written as a line in the file to be generated.

Chapter 7

Utilities developed

Along with the two modules required to integrate the REP, some other functionalities for GMQL-Importer has been developed during this thesis project. Also some tools are added to the program for future use, some already existing methods are improved or rewritten, and some general code modifications have been made.

7.1 Unzipper

The `Unzipper` singleton object contains the method `unGzipIt` and allows to all the classes that need it to call the method. `unGzipIt` was previously repeated in multiple classes. It is the method used to extract the files inside a GZip archive and put the extracted files in a directory specified by the user. By adding the method to an object, the code replication is avoided. `Unzipper` is meant to allow future expansion of the code since it is possible to change the implementation of `unGzipIt` without changing the classes that use it. It also allows adding a function that extracts files from different types of archives and keeps them together in the same object.

7.2 OAuth

The *OAuth* is the class that allows authenticating an application with Google API through the OAuth2 protocol. It instantiates all the required objects to carry out with success the authentication process when instantiated. The OAuth object can receive the path where to store the acquired credential when instantiated, or the working directory used by default if no custom path is provided. The method `authorizes` returns the credential given the client secret[1] as input. The credential can be used to access to any service provided by Google API. The method `getSheetsService` uses the credential to obtain access to the Spreadsheet services. The class can be potentially expanded with a method to access to other services that require OAuth2 authentication.

[1] The client secret is a JSON file containing the information required to identify the application when it requires to access a remote service that use a OAuth2 protocol to authenticate the clients. It is obtained by registering the application.

7.3 csvDownload

The `csvDownload` is a class devoted to downloading a sheet from a Google spreadsheet, given in input the spreadsheet identifier and name. When instantiated the URL used to downloads a sheet in CSV format is generated. The method actually download the file with a name and location functional to what is required by RoadmapImporter. This class allows to generate a downloader for any spreadsheet and can be extended by adding new functions that download files in other formats, perform other downloading action (download all the sheets), or save them with different name and location.

7.4 checkRegionData

The `checkRegionData` method of the singleton object `Transformer` has been completely rewritten to effectively implement the required missing value policy[2] and to check the consistency of the region data to the schema provided by the user for the dataset to which the data belong. It allows to parse both GDM/TAB format and GTF format. All the region files are treated as a GDM file, unless the file schema has the tag *gmqlSchema* with attribute *type* set as "gtf". So, the first thing that the method does is to load the schema file to check if the GMQL schema type is TAB or GTF. If it is GTF, some additional checks are performed because it can be considered a particular case of the TAB format. Problems found during the checks can rise an action (e.g., delating a row, changing a value) and `checkRegionData` keeps track of the problem using a pool of counters. At the end of the method, for each file, the counters are looked up and a summary of the problems detected, if any, is reported in the log file.

If the file is in GTF format, the following checks are made and actions performed:

- if a region contains less then the nine mandatory attributes separated by tab, then the region is removed and the problem is registered;
- if the number of optional attributes contained in the ninth mandatory attribute do not correspond to that defined in the schema, then the issue is registered and region discarded;
- if the region optional attributes are in the wrong format, it is reported;
- if the region optional attribute names do not match the schema attribute names, the issue is reported.

Instead, if the file is in tab format:

- if the number of region attributes is not equal to the number of attributes specified in the schema, then the region is removed and the event registered.

Then, for all the attributes and formats, the following rules and actions have been implemented:

- if the attribute score is present without a numeric value ("", "NULL", "N/A", "."), its value is replaced with ".", and the event is reported;

[2] With missing value policy, we mean the set of rules and conventions applied by GMQL when dealing with missing value.

- if a numeric attribute different from score is present with a value "", "N/A", or "NA", then it is replaced with value "NULL"; if the attribute is a string or a character and it has value "NULL", "NA", or "N/A", then its value is replaced with value "" (void string) and the event is registered;
- if the type of the attribute does not correspond to the type defined for that attribute in the schema, no action is taken but the problem is registered;
- if the attribute strand does not have a valid value ("+", "-", ".", "*"), the invalid value is replaced with value '.' and the problem is reported.

At the end of the inspection of each file some data about the status and quality of the scanned file are reported in the log:

- the total number of regions deleted because of a wrong number of attributes;
- the total number of regions with an invalid strand value replaced;
- the total number of missing values for each attribute;
- the total number of type mismatches for each attribute.

Plus, if the file is in GTF format, the following log entries are added:

- the total number of lines with wrong numbers of mandatory attributes removed;
- the total number of lines with wrong numbers of optional attributes removed;
- the total number of wrong region attribute format issues for each optional attributes;
- the total number of wrong region attribute name issues for each optional attributes.

By looking at the final log, it is possible to deduce if a file has a lot of missing values or if the file is different from the schema defined by the user and take some action accordingly (e.g., change schema, discard the file). This method is totally general and is applied every time the transformation phase is executed, independently on the transformation module set by the user. It has been designed to be easy to add more rules or modify existing rules.

The method receives in input the region file to check and the schema file to use as reference. It works performing the following steps:

- the XML schema is read using the Scala XML parser;
- a temporary file is generated along with its writer;
- the file to check is read line by line;
- if the file is a GTF, the additional checks are performed on each line and related actions are taken;
- the rules and the actions in common to all the file types are applied to each line;

- if the line has not been deleted by an action triggered by a rule match, then the line is written in the temporary file;
- if the original file has been modified by some actions, the file is replaced by the temporary one, else the temporary file is deleted and the original file is kept.

At the end, two booleans are returned to indicate if the original file has been modified or not and if the file is correct with respect to the schema.

7.5 regionFileSort

The `checkRegionData` is a method of the singleton object `Transformer` in charge to receive a region file and to sort the regions of the file by their coordinates (i.e., the region attributes chrom, start, end). The file is read line by line and each line is inserted in an immutable Scala `TreeMap`, a sorted data structure capable of carry out all the basic operations in a time proportional to the logarithm of the collection size [96]. The `TreeMap` works with key-value pairs, where, in our case, the value is a line of the region file and the key is the corresponding regions file name. Since a region file can potentially contain two regions with the same coordinates, the lines associated with some coordinate are placed in a mutable Scala `Queue`, to avoid to lose lines with coordinates already inserted in the `TreeMap`. This is done by associating each coordinate with a region coordinate. Each element contained in the `TreeMap` is a coordinates-`Queue` pairs. Each `Queue` contains all the lines with the same region coordinates. The insertion of a line in the `Queue` is performed in constant time [96]. So, summing the time required to read all the lines, the time to insert all the coordinates in the `TreeMap` and the line in the `Queue` associated with region coordinate, the time complexity of the data structure building is $\mathcal{O}(n + n \log n + n) = \mathcal{O}(n \log n)$, where n is the number of regions contained in the region file. Now that all the regions are loaded and sorted in memory, the lines are rewritten overwriting the previous file. The `TreeMap` is traversed in $\mathcal{O}(n \log n)$ and each line to write is dequeued in constant time. The time complexity of writing the sorted regions on the file is $\mathcal{O}(n \log n)$, so we can conclude that the overall time complexity required to sort a region file is $\mathcal{O}(n \log n)$. In the limit case of all the n regions with the same coordinates the time complexity becomes linear w.r.t. n since they are queued and dequeued in a single `Queue`. Anyway, we can assume that regions with same coordinates are very rare, as it always happens in all the region files obtained from biological experiments.

Chapter 8

Results

During this project thesis we develop two modules, RoadmapDownloader and RoadmapTransformer, capable of integrate themselves in GMQL-Importer, if the configuration file is set properly by user, and perform the integration of the most relevant (w.r.t. the tertiary analysis that will be performed on them using GMQL once integrated in a GDM repository) datasets available in the REP remote repository. Along with the modules, some other functionality for GMQL-Importer are developed, the most relevant ones for the user are the regions sorting in coordinates order inside a samples and the regions consistency check with the associated schema. Using the developed code we successfully integrate six datasets:

- HG19_ROADMAP_EPIGENOMICS_NARROW, it contains ChIP-seq and DNase narrow regions called using MACS2 in narrowPeak format files;
- HG19_ROADMAP_EPIGENOMICS_BROAD, it contains broad ChIP-seq regions called using MACS2 in broadPeak format files;
- HG19_ROADMAP_EPIGENOMICS_GAPPED, it contains ChIP-seq gapped regions called using MACS2 in gappedPeak format files;
- HG19_ROADMAP_EPIGENOMICS_BED, it contains DNase regions (narrow and broad) called using HOTSPOT in BED format files;
- HG19_ROADMAP_EPIGENOMICS_RNA_expression, it contains DNase regions (narrow and broad) called using HOTSPOT in BED format files;
- HG19_ROADMAP_EPIGENOMICS_DMR, it contains the RNA expression quantification of different gene, exon and intron regions in BED format files.

8.1 Test performed

To check if the code developed works properly and meets the software requirements, we perform three different tests of increasing complexity:

1. the run the code over small subsets of REP data and metadata representative of each of the datasets, with the purpose to fast detect any bugs and problems;
2. the downloading and transforming of all the data and metadata inside the REP six datasets, to check that all the data are correctly integrated;

Table 8.1: Time required by GMQL-Importer to complete the tests.

Test	**Download time**	**Transform time**	**Total Time**	**Overhead**
RoadmapSmall	00:34:53	00:11:42	00:46:42	00:00:07
Roadmap	12:48:33	05:09:59	17:58:39	00:00:07
Roadmap + ENCODE + TCGA	14:07:33	05:50:41	19:58:27	00:00:13

3. the execution of the program using the complete REP datasets plus some representatives datasets from ENCODE and TCGA, to check the interoperability of the new code with the legacy code.

In Table 8.1, we report, for each test, the time required to download the datasets form the source, the time spent to transform data and metadata, and the total time to complete all the executions. We compute also the overhead as:

$$overhead = total\_time - (download\_time + transform\_time).$$

The overhead time represents the time spent in computations that are nor download phase nor transformation phase. This time is almost insignificant if competed with the time required to download or transform the data, especially when the data are a lot. It increase slightly with the number of sources, but it can be ignored. As Table 8.1 highlights, the most of the time, in the configuration used to perform the test, is required to complete the download phase. The network is the main bottleneck of the system, and, incising the download speed, we can improve significantly the total time required to perform the transformation. The test are executed on a server with 2 Intel Xeon E5-2650 CPUs (2.00GHz speed, 8 core each [97]) and 378GB of memory.

8.2 Statistics

Tables 8.2 and 8.3 report the number of region files and metadata files for each dataset before and after the data transformation phase respectively. As it can be seen, the number of region files remain the same for almost all the datasets, since the source files are in a format quite similar to that required by GDM, and the transformation process only extracts them from Gzip archives and perform minor modifications. In some cases, the name of the files is also changed, with the purpose to have a uniform naming convention across all the datasets. For certain HG19_ROADMAP_EPIGENOMICS_BED dataset files, some region attributes are added with null value to fit files containing similar data with originally different schemas into the same dataset. Instead, the files belonging to the HG19_ROADMAP_EPIGENOMICS_DMR dataset are modified by adding the missing strand genome coordinate, which in GDM is mandatory, setting it to value ".", and a duplicated region attribute is removed. Different it is the case of RNA_expression, where the source data are in a really different format from what required by GDM. For HG19_ROADMAP_EPIGENOMICS_RNA_expression, the number of files obtained from the transformation is much greater then the number of source files since, for each pair of source files, a new file is created for each column in the source files. For each of the new file generated, the mandatory GDM coordinates are added, along with other region attributes, taking them from a reference file (provided by Roadmap Epigenomics) with gene info and genomic locations. Regarding the metadata files, before data transformation all the datasets have the same number of files because all the datasets use the metadata available through the *Google spreadsheet: Metadata and quality control* [64]. During data transformation, the relevant metadata are extracted from the tables and associated with the region files so that each region file has a corresponding metadata file, as required by GDM.

Tables 8.2 and 8.3 report also the total region and metadata files size in each dataset. The size of metadata is negligible if compared with the size of the region files, and the overall size across all the datasets is nearly left unchanged by the transformation process. The size of the region data after the data transformation, instead, is greatly increased (percentage increase of 239,7%). This is caused mainly by the extraction from the compressed archives and partially by the afresh generated files.

Dataset	**N. of data files**	**Total size of data files**	**N. of metadata files**	**Total size of metadata files**
HG19_ROADMAP_EPIGENOMICS_NARROW	1032 .gz	2.629 GB	3 .csv	911.206 KB
HG19_ROADMAP_EPIGENOMICS_BROAD	979 .gz	7.685 GB	3 .csv	911.206 KB
HG19_ROADMAP_EPIGENOMICS_GAPPED	979 .gz	2.734 GB	3 .csv	911.206 KB
HG19_ROADMAP_EPIGENOMICS_BED	156 .bed	315.229 MB	3 .csv	911.206 KB
HG19_ROADMAP_EPIGENOMICS_RNA_expression	14 .gz + 1 .gene_info	85.722 MB + 5.131 MB	3 .csv	911.206 KB
HG19_ROADMAP_EPIGENOMICS_DMR	66 .gz	1.160 GB	3 .csv	911.206 KB
Total	**3227**	**14.613 GB**	**18**	**5.467 MB**

Table 8.2: Table containing the number of files and the total size of each dataset after the data download phase.

Dataset	**N. of data files**	**Total size of data files**	**N. of metadata files**	**Total size of metadata files**
HG19_ROADMAP_EPIGENOMICS_NARROW	1032 .narrowPeak	11.785 GB	1032 .meta	1.861 MB
HG19_ROADMAP_EPIGENOMICS_BROAD	979 .broadPeak	24.330 GB	979 .meta	1.760 MB
HG19_ROADMAP_EPIGENOMICS_GAPPED	979 .gappedPeak	6.873 GB	979 .meta	1.764 MB
HG19_ROADMAP_EPIGENOMICS_BED	156 .bed	1.106 GB	156 .meta	309.836 KB
HG19_ROADMAP_EPIGENOMICS_RNA_expression	399 .bed	2.488 GB	399 .meta	586.501 KB
HG19_ROADMAP_EPIGENOMICS_DMR	66 .bed	3.060 GB	66 .meta	84.435 KB
Total	**3611**	**49.642 GB**	**3611**	**6.365 MB**

Table 8.3: Table containing the number of files and the total size of each dataset after the data transformation phase.

Chapter 9

Conclusions

This thesis describes all the steps taken to develop an automated procedure able to integrate the Roadmap Epigenomics Project into a GDM repository. First, we successfully identify the most relevant data and metadata available through REP, and we manage to group them into six semantically homogeneous datasets. Second, we accomplish the data integration of the selected datasets into a GDM repository by implementing two new modules for GMQL-Importer. The modules are able to get all the data and metadata from the REP repository and group them in the six semantically homogeneous datasets individuated during the analysis phase, performing all the necessary transformation to make the data inside each dataset compatible with the GDM. At the end of the execution of the RoadmapImporter modules, we obtain the six datasets generated by assembling semantically homogeneous, but structurally heterogeneous, data. The data are organized in samples, consisting of set of genomic regions. The integrated samples inside the same dataset are both structurally and semantically homogeneous. Each sample has its metadata file associated. The metadata have been transformed to be human readable and enrich the information provided by a sample. The most relevant data are now available to be queried using GMQL, allowing everybody to perform tertiary analysis in an easy and efficient way, taking advantage of all the services provided by GMQL. The primary goals of the thesis have been reached and the results obtained during this project are now at disposal for future expansions, some of them proposed in Chapter 10.

Also the secondary objective of add new functionalities to GMQL-Importer to be used in combination to any module has been reached, mainly adding the possibility to sorting the regions, implementing the check of consistency between regions and schema and managing the missing attribute values.

This thesis is also an effective example of how it is possible to use GMQL-Importer to carry out, in a relatively easy way, the potentially complex task to integrate any type of data, coming from any source, into a target repository based on any data model. In fact, this thesis completely describes the process required to integrate heterogeneous datasets, from the source analysis to the modules testing, going through the homogeneous datasets design and transformation process development by means of GMQL-Importer modules; so it can be used as reference for future development of other GMQL-Importer modules for integrating data and metadata coming from new sources.

Chapter 10

Future expansions

Future expansions to this project mainly regard two aspects: the addition of further functionalities to GMQL-Importer and the use of the integrated data. As regards the further expansion of GMQL, new modules able to integrate new sources of genomic data and metadata can be added, but also the RoadmapImporter modules can be extended by adding new datasets and types of data that can be integrated. In both case the addition of new code is facilitated by the modular structure of the project and the code design of single methods, developed with expandability in mind. Instead, the possible expansion of the project through the use of data, the possibilities are enormous. GMQL allows to perform a vast range of tertiary analysis, but the integrated data and metadata can be used also for ad-hoc computations, using GMQL, or any of its related services, to select, filter and retrieve the required data to use in further elaboration exploiting any software able to manage GDM data (eventually ad-hoc developed). The advantage to retrieve the data from a GDM repository instead to get them directly from the original source is evident. First, the data are homogeneous and compliant to a well defined and structured data model, compatible and interoperable with popular bioinformatics tools, such us Bioconductor packages and GRanges data structure. Moreover, during the integration process, data have got through some data cleaning process too, where duplicates are removed and missing values are managed as required by GDM and GMQL well defined missing value policies. Second, taking the data from a GDM repository, the datasets can be customized, taking only some subsets of interest (reducing the time required for the following computations), or integrated with data from different original sources, but totally independent from them (expanding in this way the domain of the future analysis). With this in mind, the work done in this project thesis is only a step towards the wider plan to provide some tools to make tertiary analysis, and the use of genomic data in general, more effective and easy to perform from a technical point of view, allowing, in this way, the end-users of such tools to focus entirely on complex biological and medical problems.

Appendix A

Table cell line / tissue - experiment count

In this table all the cell lines/tissues from which samples are taken are reported. For each cell line/tissue, all the experiments performed are listed with the corresponding datasets produced by that experiment type for that particular cell line/tissue. The total number of datasets associated with the cell line/tissue is reported too.

Table A.1: Number of datasets available for each experiment and each cell lines.

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
adipose	**8**		
Bisulfite-Seq	3	ChIP-Seq input	1
H3K27ac	1	mRNA-Seq	3
adipose derived mesenchymal stem cells	**7**		
ChIP-Seq input	1		
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9ac	1	H3K9me3	1
adipose derived mesenchymal stem cells day20	**14**		
ChIP-Seq input	2		
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	2	H3K9me3	2
adipose derived mesenchymal stem cells, day0	**13**		
ChIP-Seq input	2		
H3K27me3	1	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	2	H3K9me3	2
adipose nuclei	**36**		
ChIP-Seq input	5		
H3K27ac	1	H3K27me3	5
H3K36me3	5	H3K4me1	5

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
H3K4me3	5	H3K9ac	5
H3K9me3	5		
adrenal gland	**14**		
Bisulfite-Seq	3	ChIP-Seq input	2
H3K27ac	2	H3K36me3	2
H3K4me1	2	H3K9me3	1
mRNA-Seq	2		
adrenal gland, fetal day101 M	**1**		
DNase hypersensitivity	1		
adrenal gland, fetal day108 F	**1**		
DNase hypersensitivity	1		
adrenal gland, fetal day108 M	**1**		
DNase hypersensitivity	1		
adrenal gland, fetal day113 F	**2**		
DNase hypersensitivity	1	Genotyping array	1
adrenal gland, fetal day85 F	**1**		
DNase hypersensitivity	1		
adrenal gland, fetal day85 M	**1**		
Exon array	1		
adrenal gland, fetal day96 U	**2**		
DNase hypersensitivity	1		
Exon array	1		
adrenal gland, fetal day97 M	**7**		
ChIP-Seq input	1	H3K27ac	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9me3	1		
bladder	**6**		
ChIP-Seq input	2	H3K27ac	1
H3K36me3	1	H3K4me1	1
mRNA-Seq	1		
bone marrow derived mesenchymal stem cells	**34**		
ChIP-Seq input	4	H3K27ac	4
H3K27me3	4	H3K36me3	4
H3K4me1	4	H3K4me3	4
H3K9ac	4	H3K9me3	4
RRBS	2		
brain, angular gyrus	**14**		
ChIP-Seq input	1	H3K27ac	2
H3K27me3	1	H3K36me3	2
H3K4me1	2	H3K4me3	2

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
H3K9ac	1	H3K9me3	2
RRBS	1		
brain, anterior caudate	**17**		
ChIP-Seq input	2	H3K27ac	2
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	1	H3K9me3	2
RRBS	2		
brain, cerebellum	**1**		
mRNA-Seq	1		
brain, cingulate gyrus	**15**		
ChIP-Seq input	2	H3K27ac	2
H3K27me3	1	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	1	H3K9me3	2
RRBS	1		
brain, dorsal neocortex, fetal week15 U	**3**		
H3K27me3	1	H3K4me3	1
H3K9ac	1		
brain, fetal day101 M	**1**		
DNase hypersensitivity	1		
brain, fetal day104 M	**1**		
DNase hypersensitivity	1		
brain, fetal day105 M	**1**		
DNase hypersensitivity	1		
brain, fetal day109 F	**1**		
DNase hypersensitivity	1		
brain, fetal day112 U	**2**		
DNase hypersensitivity	1	Exon array	1
brain, fetal day117 F	**2**		
DNase hypersensitivity	1		
Exon array	1		
brain, fetal day120 U	**4**		
ChIP-Seq input	1	H3K27me3	1
H3K4me1	1	H3K9me3	1
brain, fetal day122 M	**11**		
Digital genomic footprinting	1		
DNase hypersensitivity	2	Exon array	1
H3K27me3	1	H3K36me3	1
H3K4me3	1	H3K9me3	1
MeDIP-Seq	1	MRE-Seq	1

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
RRBS	1		
brain, fetal day142 F	**1**		
DNase hypersensitivity	1		
brain, fetal day85 F	**4**		
DNase hypersensitivity	2	Exon array	1
MeDIP-Seq	1		
brain, fetal day96 F	**6**		
ChIP-Seq input	1	DNase hypersensitivity	2
Exon array	1	MeDIP-Seq	1
RRBS	1		
brain, fetal week17 F	**20**		
ChIP-Seq input	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
MeDIP-Seq	2	MRE-Seq	2
mRNA-Seq	2	smRNA-Seq	2
brain, germinal matrix, fetal week20 M	**17**		
Bisulfite-Seq	1		
ChIP-Seq input	1	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
MeDIP-Seq	1	mRNA-Seq	1
smRNA-Seq	3		
brain, hippocampus middle	**26**		
Bisulfite-Seq	2	ChIP-Seq input	3
H3K27ac	3	H3K27me3	3
H3K36me3	3	H3K4me1	3
H3K4me3	3	H3K9ac	1
H3K9me3	3	mRNA-Seq	2
brain, inferior temporal lobe	**16**		
ChIP-Seq input	2		
H3K27ac	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9ac	1
H3K9me3	2	RRBS	1
brain, mid frontal, Brodmann area 9/46, dorsolateral prefrontal cortex	**15**		
ChIP-Seq input	2		
H3K27ac	2	H3K27me3	1
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9ac	1
H3K9me3	2	RRBS	1
brain, substantia nigra	**17**		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
ChIP-Seq input	2		
H3K27ac	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9ac	1
H3K9me3	2	RRBS	2
breast, fibroblast primary cells	**8**		
ChIP-Seq input	1		
H3K27ac	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1	mRNA-Seq	2
breast, luminal epithelial cells	**20**		
Bisulfite-Seq	1		
ChIP-Seq input	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K9me3	1	MeDIP-Seq	5
MRE-Seq	5	mRNA-Seq	3
smRNA-Seq	1		
breast, myoepithelial cells	**29**		
Bisulfite-Seq	3	ChIP-Seq input	2
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	2	H3K9me3	2
MeDIP-Seq	4	MRE-Seq	4
mRNA-Seq	3	smRNA-Seq	1
breast, stem cells	**12**		
ChIP-Seq input	1		
MeDIP-Seq	4	MRE-Seq	4
mRNA-Seq	2	smRNA-Seq	1
breast, vHMEC	**17**		
ChIP-Seq input	2		
Digital genomic footprinting	1	DNase hypersensitivity	2
Genotyping array	1	H3K27me3	1
H3K36me3	1	H3K4me1	2
H3K4me3	1	H3K9me3	1
MeDIP-Seq	1	MRE-Seq	1
mRNA-Seq	2	smRNA-Seq	1
CD14 primary cells	**14**		
ChIP-Seq input	1		
DNase hypersensitivity	2	Exon array	3
Genotyping array	2	H3K27ac	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9me3	1		
CD15 primary cells	**7**		
ChIP-Seq input	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1	H3K9me3	1

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
RRBS	1		
CD19 primary cells	**26**		
ChIP-Seq input	3	DNase hypersensitivity	3
Exon array	1	Expression array	1
H3K27ac	1	H3K27me3	4
H3K36me3	3	H3K4me1	2
H3K4me3	3	H3K9me3	4
RRBS	1		
CD20 primary cells	**1**		
DNase hypersensitivity	1		
CD3 cord blood primary cells	**5**		
DNase hypersensitivity	2	Exon array	2
Expression array	1		
CD3 mobilized primary cells	**3**		
DNase hypersensitivity	2	Exon array	1
CD3 primary cells	**25**		
ChIP-Seq input	3		
Digital genomic footprinting	1	DNase hypersensitivity	2
Exon array	2	H3K27ac	1
H3K27me3	3	H3K36me3	3
H3K4me1	3	H3K4me3	3
H3K9me3	3	RRBS	1
CD34 cultured cells	**5**		
H3K27me3	1		
H3K36me3	1	H3K4me1	1
H3K4me3	1	H3K9me3	1
CD34 mobilized primary cells	**78**		
ChIP-Seq input	12		
Digital genomic footprinting	1	DNase hypersensitivity	14
Exon array	4	H3K27ac	3
H3K27me3	8	H3K36me3	7
H3K4me1	6	H3K4me3	8
H3K9me3	7	mRNA-Seq	1
RRBS	7		
CD34 primary cells	**24**		
Bisulfite-Seq	1	ChIP-Seq input	1
DNase hypersensitivity	1	Exon array	8
Expression array	1	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
RRBS	2		
CD4 memory primary cells	**24**		
ChIP-Seq input	3	H3K27ac	2
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	3

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
H3K9me3	2	MeDIP-Seq	3
MRE-Seq	3	mRNA-Seq	1
smRNA-Seq	1		
CD4 mobilized primary cells	**4**		
ChIP-Seq input	1	DNase hypersensitivity	3
CD4 naive primary cells	**23**		
ChIP-Seq input	2		
H3K27ac	2	H3K27me3	2
H3K36me3	3	H3K4me1	2
H3K4me3	2	H3K9ac	1
H3K9me3	2	MeDIP-Seq	3
MRE-Seq	3	mRNA-Seq	1
CD4 primary cells	**9**		
ChIP-Seq input	1		
Digital genomic footprinting	1	DNase hypersensitivity	3
Exon array	1	Genotyping array	1
H3K36me3	1	H3K4me3	1
CD4+ CD25- CD45RA+ naive primary cells	**13**		
ChIP-Seq input	2		
H3K27ac	1	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
CD4+ CD25- CD45RO+ memory primary cells	**13**		
ChIP-Seq input	2		
H3K27ac	1	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
CD4+ CD25- IL17- PMA-ionomycin stimulated MACS purified Th primary cells	**14**		
ChIP-Seq input	2		
H3K27ac	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
CD4+ CD25- IL17+ PMA-ionomcyin stimulated Th17 primary cells	**13**		
ChIP-Seq input	2		
H3K27ac	1	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
CD4+ CD25- Th primary cells	**13**		
ChIP-Seq input	2		
H3K27ac	1	H3K27me3	2

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
CD4+ CD25+ CD127- Treg primary cells	**13**		
ChIP-Seq input	2		
H3K27ac	1	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
CD4+ CD25int CD127+ Tmem primary cells	**14**		
ChIP-Seq input	2		
H3K27ac	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
CD56 mobilized primary cells	**1**		
DNase hypersensitivity	1		
CD56 primary cells	**9**		
ChIP-Seq input	1		
Digital genomic footprinting	1	DNase hypersensitivity	2
H3K27ac	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1		
CD8 memory primary cells	**13**		
ChIP-Seq input	2	H3K27ac	1
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9me3	2		
CD8 mobilized primary cells	**1**		
DNase hypersensitivity	1		
CD8 naive primary cells	**31**		
ChIP-Seq input	4	H3K27ac	2
H3K27me3	3	H3K36me3	3
H3K4me1	3	H3K4me3	4
H3K9ac	1	H3K9me3	3
MeDIP-Seq	3	MRE-Seq	3
mRNA-Seq	1	smRNA-Seq	1
CD8 primary cells	**13**		
ChIP-Seq input	1		
Digital genomic footprinting	1	DNase hypersensitivity	4
Exon array	3	H3K27ac	1
H3K36me3	1	H3K4me3	1
H3K9me3	1		
chondrocytes from bone marrow derived mesenchymal stem cells	**36**		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
ChIP-Seq input	5	H3K27ac	4
H3K27me3	4	H3K36me3	4
H3K4me1	4	H3K4me3	4
H3K9ac	4	H3K9me3	4
RRBS	3		
colon smooth muscle	**16**		
ChIP-Seq input	2	Expression array	1
H3K27ac	1	H3K27me3	2
H3K36me3	3	H3K4me1	1
H3K4me3	2	H3K9ac	1
H3K9me3	2	RRBS	1
colonic mucosa	**17**		
ChIP-Seq input	2		
H3K27ac	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9ac	2
H3K9me3	2	RRBS	1
duodenum mucosa	**17**		
ChIP-Seq input	2		
Expression array	1	H3K27ac	1
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	2	H3K9me3	2
RRBS	1		
duodenum smooth muscle	**13**		
ChIP-Seq input	2	H3K27ac	1
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9me3	2		
ES-I3 cell line	**18**		
ChIP-Seq input	2	Expression array	2
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	2	H3K9me3	2
RRBS	2		
esophagus	**16**		
Bisulfite-Seq	1	ChIP-Seq input	3
H3K27ac	2	H3K27me3	1
H3K36me3	2	H3K4me1	2
H3K4me3	1	H3K9me3	2
mRNA-Seq	2		
ES-WA7 cell line	**10**		
ChIP-Seq input	1	Expression array	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9ac	1	H3K9me3	1
RRBS	2		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
fibroblast primary cell line	**6**		
RRBS	6		
fibroblasts, skin, abdomen, fetal day97 M	**2**		
mRNA-Seq	2		
fibroblasts, skin, back, fetal day96 M	**2**		
mRNA-Seq	2		
fibroblasts, skin, scalp, fetal day97 M	**2**		
mRNA-Seq	2		
gastric	**23**		
Bisulfite-Seq	3	ChIP-Seq input	3
DNase hypersensitivity	2	Genotyping array	1
H3K27ac	3	H3K27me3	1
H3K36me3	2	H3K4me1	2
H3K4me3	1	H3K9me3	2
mRNA-Seq	3		
H1 +BMP4 cell line	**8**		
Bisulfite-Seq	6	mRNA-Seq	2
H1 BMP4 derived mesendoderm cultured cells	**46**		
Bisulfite-Seq	4		
ChIP-Seq input	4	Digital genomic footprinting	1
DNase hypersensitivity	2	H2AK5ac	2
H2BK120ac	2	H2BK15ac	2
H2BK5ac	2	H3K18ac	2
H3K23ac	1	H3K27ac	4
H3K27me3	4	H3K36me3	1
H3K4ac	1	H3K4me1	2
H3K4me2	2	H3K4me3	2
H3K79me1	2	H3K79me2	1
H3K9ac	2	H4K8ac	1
mRNA-Seq	2		
H1 BMP4 derived trophoblast cultured cells	**59**		
ChIP-Seq input	3	Digital genomic footprinting	1
DNase hypersensitivity	2	H2A.Z	2
H2AK5ac	2	H2BK120ac	2
H2BK12ac	2	H2BK5ac	2
H3K14ac	2	H3K18ac	2
H3K23ac	2	H3K27ac	3
H3K27me3	3	H3K36me3	4
H3K4ac	2	H3K4me1	4
H3K4me2	2	H3K4me3	3
H3K79me1	2	H3K79me2	2
H3K9ac	2	H3K9me3	3

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
H4K12ac	2	H4K8ac	2
H4K91ac	1	mRNA-Seq	2
H1 cell line	**114**		
Bisulfite-Seq	5		
ChIP-Seq input	13	DNase hypersensitivity	2
Genotyping array	1	H2A.Z	1
H2AK5ac	2	H2BK120ac	3
H2BK12ac	2	H2BK15ac	2
H2BK20ac	2	H2BK5ac	2
H3K14ac	2	H3K18ac	2
H3K23ac	2	H3K23me2	2
H3K27ac	2	H3K27me3	6
H3K36me3	7	H3K4ac	2
H3K4me1	6	H3K4me2	2
H3K4me3	8	H3K56ac	2
H3K79me1	3	H3K79me2	2
H3K9ac	5	H3K9me3	7
H4K20me1	2	H4K5ac	2
H4K8ac	2	H4K91ac	2
MRE-Seq	3	mRNA-Seq	3
RRBS	4	smRNA-Seq	1
H1 derived embryoid body cultured cells	**1**		
RRBS	1		
H1 derived mesenchymal stem cells	**46**		
Bisulfite-Seq	2		
ChIP-Seq input	2	Digital genomic footprinting	1
DNase hypersensitivity	2	H2A.Z	1
H2AK5ac	2	H2BK120ac	1
H2BK12ac	2	H2BK5ac	2
H3K14ac	2	H3K18ac	2
H3K23ac	2	H3K27ac	2
H3K27me3	2	H3K36me3	2
H3K4ac	2	H3K4me1	2
H3K4me2	2	H3K4me3	2
H3K79me1	2	H3K9ac	2
H3K9me3	2	H4K8ac	2
H4K91ac	1	mRNA-Seq	2
H1 derived neuronal progenitor cultured cells	**56**		
Bisulfite-Seq	5		
ChIP-Seq input	4	DNase hypersensitivity	2
H2AK5ac	2	H2BK120ac	2
H2BK12ac	1	H2BK15ac	2
H2BK5ac	1	H3K14ac	1
H3K18ac	2	H3K23ac	2
H3K27ac	5	H3K27me3	3
H3K36me3	3	H3K4ac	2
H3K4me1	3	H3K4me2	1

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
H3K4me3	4	H3K79me1	1
H3K9ac	1	H3K9me3	3
H4K8ac	1	H4K91ac	1
mRNA-Seq	4		
H9 cell line	**63**		
Bisulfite-Seq	3	ChIP-Seq input	3
DNase hypersensitivity	2	Genotyping array	1
H2A.Z	1	H2AK5ac	2
H2BK120ac	1	H2BK12ac	2
H2BK15ac	3	H2BK20ac	1
H2BK5ac	2	H3K14ac	2
H3K18ac	2	H3K23ac	2
H3K23me2	2	H3K27ac	2
H3K27me3	2	H3K36me3	2
H3K4ac	2	H3K4me1	2
H3K4me2	2	H3K4me3	2
H3K56ac	2	H3K79me1	2
H3K79me2	2	H3K9ac	2
H3K9me3	3	H3T11ph	1
H4K20me1	1	H4K5ac	2
H4K8ac	2	H4K91ac	2
RRBS	1		
H9 derived embryoid body cultured cells	**1**		
RRBS	1		
H9 derived neuron cultured cells	**9**		
ChIP-Seq input	1	H2A.Z	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9me3	1	RRBS	2
H9 derived neuronal progenitor cultured cells	**8**		
ChIP-Seq input	1		
H2A.Z	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1	H3K9me3	1
RRBS	1		
heart	**1**		
DNase hypersensitivity	1		
heart, aorta	**16**		
Bisulfite-Seq	5	ChIP-Seq input	2
H3K27ac	1	H3K27me3	1
H3K36me3	2	H3K4me1	2
H3K9me3	1	mRNA-Seq	2
heart, fetal day101 U	**8**		
ChIP-Seq input	1		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
DNase hypersensitivity	1	Exon array	2
H3K27me3	1	H3K36me3	1
H3K9me3	1	RRBS	1
heart, fetal day103 F	**2**		
DNase hypersensitivity	1		
Exon array	1		
heart, fetal day105 F	**2**		
DNase hypersensitivity	1	Exon array	1
heart, fetal day105 M	**4**		
DNase hypersensitivity	1		
Exon array	1	H3K4me1	1
H3K9ac	1		
heart, fetal day110 F	**2**		
DNase hypersensitivity	1	Exon array	1
heart, fetal day110 M	**2**		
DNase hypersensitivity	1		
Exon array	1		
heart, fetal day117 F	**1**		
Exon array	1		
heart, fetal day120 M	**1**		
DNase hypersensitivity	1		
heart, fetal day147 F	**2**		
DNase hypersensitivity	1	Exon array	1
heart, fetal day91 F	**1**		
DNase hypersensitivity	1		
heart, fetal day96 M	**3**		
ChIP-Seq input	1		
DNase hypersensitivity	1	Exon array	1
heart, fetal day96 U	**3**		
Digital genomic footprinting	1		
DNase hypersensitivity	1	Exon array	1
heart, fetal M	**2**		
H3K4me1	1		
H3K4me3	1		
heart, left ventricle	**21**		
Bisulfite-Seq	4	ChIP-Seq input	2
H3K27ac	3	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9me3	2
mRNA-Seq	2		
heart, right atrium	**9**		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
Bisulfite-Seq	3	ChIP-Seq input	1
H3K27ac	1	H3K36me3	1
H3K4me1	1	H3K9me3	1
mRNA-Seq	1		
heart, right ventricle	**14**		
Bisulfite-Seq	4	ChIP-Seq input	2
H3K27ac	1	H3K36me3	2
H3K4me1	2	H3K9me3	1
mRNA-Seq	2		
hESC-derived CD184+ endoderm cultured cells	**14**		
Bisulfite-Seq	2	H3K27ac	2
H3K36me3	1	H3K4me1	2
H3K4me3	1	H3K9me3	1
mRNA-Seq	3	RRBS	2
hESC-derived CD56+ ectoderm cultured cells	**18**		
Bisulfite-Seq	4		
ChIP-Seq input	2	H3K27ac	2
H3K27me3	2	H3K36me3	1
H3K4me1	1	H3K4me3	2
H3K9me3	2	mRNA-Seq	2
hESC-derived CD56+ mesoderm cultured cells	**17**		
Bisulfite-Seq	2		
ChIP-Seq input	2	H3K27ac	2
H3K27me3	1	H3K36me3	2
H3K4me1	1	H3K4me3	2
H3K9me3	3	mRNA-Seq	2
hSKM cell line	**24**		
ChIP-Seq input	3		
H3K27me3	3	H3K36me3	3
H3K4me1	3	H3K4me2	3
H3K4me3	3	H3K9ac	3
H3K9me3	3		
HUES 28 cell line	**1**		
RRBS	1		
HUES1 cell line	**1**		
RRBS	1		
HUES1 derived embryoid body cultured cells	**1**		
RRBS	1		
HUES13 cell line	**1**		
RRBS	1		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
HUES3 cell line	**1**		
RRBS	1		
HUES3 derived embryoid body cultured cells	**1**		
RRBS	1		
HUES44 cell line	**1**		
RRBS	1		
HUES45 cell line	**1**		
RRBS	1		
HUES45 derived embryoid body cultured cells	**1**		
RRBS	1		
HUES48 cell line	**16**		
ChIP-Seq input	2	H3K27ac	1
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	2	H3K9me3	2
RRBS	1		
HUES49 cell line	**1**		
RRBS	1		
HUES53 cell line	**1**		
RRBS	1		
HUES6 cell line	**16**		
ChIP-Seq input	2	H3K27ac	2
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	1	H3K9me3	2
RRBS	1		
HUES6 derived embryoid body cultured cells	**1**		
RRBS	1		
HUES62 cell line	**1**		
RRBS	1		
HUES63 cell line	**1**		
RRBS	1		
HUES64 cell line	**31**		
Bisulfite-Seq	2	ChIP-Seq input	4
H3K27ac	2	H3K27me3	4
H3K36me3	3	H3K4me1	3
H3K4me3	3	H3K9ac	4
H3K9me3	3	mRNA-Seq	2
RRBS	1		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
HUES65 cell line	**1**		
RRBS	1		
HUES66 cell line	**2**		
RRBS	2		
HUES8 cell line	**1**		
RRBS	1		
HUES9 cell line	**1**		
RRBS	1		
IMR90 cell line	**92**		
Bisulfite-Seq	6	ChIP-Seq input	9
Digital genomic footprinting	1	DNase hypersensitivity	4
Exon array	4	Genotyping array	1
H2A.Z	2	H2AK5ac	2
H2AK9ac	2	H2BK120ac	2
H2BK12ac	3	H2BK15ac	3
H2BK20ac	2	H2BK5ac	3
H3K14ac	2	H3K18ac	2
H3K23ac	2	H3K27ac	3
H3K27me3	2	H3K36me3	2
H3K4ac	2	H3K4me1	3
H3K4me2	2	H3K4me3	2
H3K56ac	2	H3K79me1	4
H3K79me2	2	H3K9ac	2
H3K9me1	2	H3K9me3	3
H4K20me1	2	H4K5ac	2
H4K8ac	4	H4K91ac	2
mRNA-Seq	1		
iPS DF 19.11 cell line	**24**		
Bisulfite-Seq	4	ChIP-Seq input	2
Digital genomic footprinting	1	DNase hypersensitivity	1
Exon array	1	Genotyping array	1
H3K27ac	3	H3K27me3	2
H3K36me3	1	H3K4me1	2
H3K4me3	2	H3K9me3	2
mRNA-Seq	2		
iPS DF 19.7 cell line	**2**		
DNase hypersensitivity	1	Exon array	1
iPS DF 4.7 cell line	**1**		
DNase hypersensitivity	1		
iPS DF 6.9 cell line	**19**		
Bisulfite-Seq	2		
ChIP-Seq input	2	DNase hypersensitivity	1
Exon array	1	Genotyping array	1
H3K27ac	1	H3K27me3	2
H3K36me3	2	H3K4me1	2

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
H3K4me3	2	H3K9me3	2
mRNA-Seq	1		
iPS-11a cell line	**3**		
ChIP-Seq input	1	H3K4me3	1
RRBS	1		
iPS-11b cell line	**1**		
RRBS	1		
iPS-11c cell line	**1**		
RRBS	1		
iPS-15b cell line	**10**		
ChIP-Seq input	2	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1	H3K9ac	1
H3K9me3	1	RRBS	2
iPS-17a cell line	**1**		
RRBS	1		
iPS-17b cell line	**3**		
RRBS	3		
iPS-18a cell line	**9**		
ChIP-Seq input	1		
H3K27ac	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1	H3K9ac	1
H3K9me3	1	RRBS	1
iPS-18b cell line	**4**		
ChIP-Seq input	1		
Expression array	2	RRBS	1
iPS-18c cell line	**9**		
ChIP-Seq input	1		
H3K27me3	1	H3K36me3	1
H3K4me3	2	H3K9ac	1
H3K9me3	1	RRBS	2
iPS-20b cell line	**24**		
ChIP-Seq input	4		
Expression array	1	H3K27ac	1
H3K27me3	3	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9ac	2	H3K9me3	2
RRBS	5		
iPS-27b cell line	**1**		
RRBS	1		
iPS-27e cell line	**1**		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
RRBS	1		
kidney	**15**		
ChIP-Seq input	2	H3K27ac	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9ac	2
H3K9me3	2	RRBS	1
kidney renal cortex, fetal day103 F	**1**		
DNase hypersensitivity	1		
kidney renal cortex, fetal day108 M	**4**		
DNase hypersensitivity	2		
Exon array	2		
kidney renal cortex, fetal day113 M	**2**		
DNase hypersensitivity	1	Exon array	1
kidney renal cortex, fetal day120 F	**1**		
DNase hypersensitivity	1		
kidney renal cortex, fetal day127 M	**1**		
DNase hypersensitivity	1		
kidney renal cortex, fetal day89 F	**1**		
DNase hypersensitivity	1		
kidney renal cortex, fetal day91 M	**1**		
DNase hypersensitivity	1		
kidney renal cortex, fetal day96 F	**2**		
DNase hypersensitivity	2		
kidney renal cortex, fetal day97 M	**1**		
DNase hypersensitivity	1		
kidney renal cortex, left, fetal day105 M	**3**		
DNase hypersensitivity	3		
kidney renal cortex, left, fetal day120 M	**1**		
DNase hypersensitivity	1		
kidney renal cortex, right, fetal day105 M	**2**		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
DNase hypersensitivity	2		
kidney renal cortex, right, fetal day120 M	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, fetal day103 F	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, fetal day105 F	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, fetal day108 M	**2**		
DNase hypersensitivity Exon array	1 1		
kidney renal pelvis, fetal day113 M	**2**		
DNase hypersensitivity	1	Exon array	1
kidney renal pelvis, fetal day127 M	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, fetal day89 F	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, fetal day91 M	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, fetal day96 F	**2**		
DNase hypersensitivity	2		
kidney renal pelvis, fetal day97 M	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, left, fetal day 105M	**1**		
DNase hypersensitivity	1		
kidney renal pelvis, left, fetal day105 M	**2**		
DNase hypersensitivity	2		
kidney renal pelvis, left, fetal day120 M	**1**		
DNase hypersensitivity	1		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
kidney renal pelvis, right, fetal day105 M	**3**		
DNase hypersensitivity	3		
kidney renal pelvis, right, fetal day120 M	**1**		
DNase hypersensitivity	1		
kidney, fetal day105 F	**1**		
DNase hypersensitivity	1		
kidney, fetal day105 M	**3**		
DNase hypersensitivity	2		
Exon array	1		
kidney, fetal day108 F	**1**		
DNase hypersensitivity	1		
kidney, fetal day108 M	**1**		
Genotyping array	1		
kidney, fetal day113 F	**1**		
DNase hypersensitivity	1		
kidney, fetal day121 F	**1**		
DNase hypersensitivity	1		
kidney, fetal day122 U	**9**		
DNase hypersensitivity	1	Exon array	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9ac	1	H3K9me3	1
RRBS	1		
kidney, fetal day82 F	**1**		
DNase hypersensitivity	1		
kidney, fetal day85 F	**1**		
DNase hypersensitivity	1		
kidney, fetal day85 M	**1**		
DNase hypersensitivity	1		
kidney, fetal F	**1**		
ChIP-Seq input	1		
kidney, left, fetal day107 F	**2**		
DNase hypersensitivity	1	Exon array	1
kidney, left, fetal day110 F	**2**		
DNase hypersensitivity	1		
Exon array	1		
kidney, left, fetal day115 M	**2**		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
DNase hypersensitivity	1	Exon array	1
kidney, left, fetal day147 F	**2**		
DNase hypersensitivity Exon array	1 1		
kidney, left, fetal day87 F	**1**		
DNase hypersensitivity	1		
kidney, left, fetal day87 M	**1**		
DNase hypersensitivity	1		
kidney, left, fetal day96 M	**1**		
DNase hypersensitivity	1		
kidney, left, fetal day98 F	**1**		
DNase hypersensitivity	1		
kidney, right, fetal day 98 F	**1**		
DNase hypersensitivity	1		
kidney, right, fetal day107 F	**2**		
DNase hypersensitivity	1	Exon array	1
kidney, right, fetal day108 M	**2**		
DNase hypersensitivity Genotyping array	1 1		
kidney, right, fetal day115 M	**2**		
DNase hypersensitivity	1	Exon array	1
kidney, right, fetal day117 F	**2**		
DNase hypersensitivity Exon array	1 1		
kidney, right, fetal day147 F	**2**		
DNase hypersensitivity	1	Exon array	1
kidney, right, fetal day87 F	**1**		
DNase hypersensitivity	1		
kidney, right, fetal day87 M	**1**		
DNase hypersensitivity	1		
kidney, right, fetal day96 M	**1**		
DNase hypersensitivity	1		
large intestine, fetal day103 F	**2**		
DNase hypersensitivity Exon array	1 1		
large intestine, fetal day105 M	**4**		
DNase hypersensitivity	3	Exon array	1

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued* ...

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
large intestine, fetal day107 F	**2**		
DNase hypersensitivity	1		
Exon array	1		
large intestine, fetal day108 F	**1**		
DNase hypersensitivity	1		
large intestine, fetal day108 M	**9**		
ChIP-Seq input	1	Digital genomic footprinting	1
DNase hypersensitivity	1	Exon array	1
H3K27ac	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1		
large intestine, fetal day108, M	**3**		
H3K27me3	1	H3K36me3	1
H3K9me3	1		
large intestine, fetal day110 F	**2**		
DNase hypersensitivity	1	Exon array	1
large intestine, fetal day113 M	**1**		
DNase hypersensitivity	1		
large intestine, fetal day115 M	**1**		
DNase hypersensitivity	1		
large intestine, fetal day120 F	**1**		
DNase hypersensitivity	1		
large intestine, fetal day85 F	**1**		
DNase hypersensitivity	1		
large intestine, fetal day91 F	**1**		
DNase hypersensitivity	1		
large intestine, fetal day91 M	**1**		
DNase hypersensitivity	1		
large intestine, fetal day98 F	**1**		
DNase hypersensitivity	1		
liver	**29**		
Bisulfite-Seq	1		
ChIP-Seq input	4	H3K27ac	2
H3K27me3	3	H3K36me3	4
H3K4me1	4	H3K4me3	3
H3K9ac	2	H3K9me3	4
mRNA-Seq	2		
liver, fetal day110 F	**1**		
Genotyping array	1		
liver, fetal day96 U	**1**		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
Genotyping array	1		
liver, fetal day97 M	**1**		
Genotyping array	1		
lung	**15**		
Bisulfite-Seq	1	ChIP-Seq input	3
H3K27ac	2	H3K36me3	2
H3K4me1	2	H3K4me3	1
H3K9me3	2	mRNA-Seq	2
lung, fetal day101 U	**8**		
DNase hypersensitivity	1		
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9ac	1	H3K9me3	1
RRBS	1		
lung, fetal day103 M	**2**		
DNase hypersensitivity	1	Exon array	1
lung, fetal day108 F	**3**		
DNase hypersensitivity	2		
Exon array	1		
lung, fetal day108 M	**2**		
DNase hypersensitivity	1	Exon array	1
lung, fetal day110 F	**2**		
Exon array	2		
lung, fetal day112 U	**2**		
DNase hypersensitivity	1		
Exon array	1		
lung, fetal day113 M	**2**		
DNase hypersensitivity	1	Exon array	1
lung, fetal day117 F	**2**		
Exon array	2		
lung, fetal day122 U	**3**		
DNase hypersensitivity	1		
Exon array	1	H3K4me3	1
lung, fetal day67 M	**2**		
DNase hypersensitivity	1		
Exon array	1		
lung, fetal day82 F	**4**		
ChIP-Seq input	1	H3K27me3	1
H3K4me1	1	H3K9ac	1
lung, fetal day82 M	**2**		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued …*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
DNase hypersensitivity	1		
Exon array	1		
lung, fetal day85 F	**8**		
ChIP-Seq input	1	Digital genomic footprinting	1
DNase hypersensitivity	2	Exon array	1
H3K4me1	1	H3K9me3	1
RRBS	1		
lung, fetal day91 M	**1**		
DNase hypersensitivity	1		
lung, fetal day96 F	**5**		
ChIP-Seq input	1	DNase hypersensitivity	2
Exon array	1	H3K36me3	1
lung, fetal day98 F	**1**		
DNase hypersensitivity	1		
lung, left, fetal day105 F	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
lung, left, fetal day105 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
lung, left, fetal day107 F	**2**		
DNase hypersensitivity	1		
Exon array	1		
lung, left, fetal day108 F	**1**		
mRNA-Seq	1		
lung, left, fetal day110 F	**1**		
DNase hypersensitivity	1		
lung, left, fetal day115 M	**1**		
DNase hypersensitivity	1		
lung, left, fetal day117 F	**1**		
DNase hypersensitivity	1		
lung, left, fetal day87 M	**1**		
DNase hypersensitivity	1		
lung, left, fetal day91 F	**2**		
DNase hypersensitivity	1	Exon array	1
lung, left, fetal day91 M	**1**		
mRNA-Seq	1		
lung, left, fetal day96 M	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		

Continued …

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
lung, left, fetal day98 F	**1**		
mRNA-Seq	1		
lung, right, fetal day105 F	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
lung, right, fetal day105 M	**2**		
DNase hypersensitivity mRNA-Seq	1 1		
lung, right, fetal day107 F	**2**		
DNase hypersensitivity	1	Exon array	1
lung, right, fetal day108 F	**2**		
DNase hypersensitivity mRNA-Seq	1 1		
lung, right, fetal day110 F	**1**		
DNase hypersensitivity	1		
lung, right, fetal day115 M	**1**		
DNase hypersensitivity	1		
lung, right, fetal day117 F	**1**		
DNase hypersensitivity	1		
lung, right, fetal day87 M	**1**		
DNase hypersensitivity	1		
lung, right, fetal day91 F	**2**		
DNase hypersensitivity	1	Exon array	1
lung, right, fetal day96 M	**2**		
DNase hypersensitivity mRNA-Seq	1 1		
lung, right, fetal day98 F	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, arm, fetal day101	**1**		
mRNA-Seq	1		
muscle, arm, fetal day101 M	**1**		
DNase hypersensitivity	1		
muscle, arm, fetal day101 U	**1**		
DNase hypersensitivity	1		
muscle, arm, fetal day104 M	**1**		
DNase hypersensitivity	1		
muscle, arm, fetal day105 F	**2**		
DNase hypersensitivity mRNA-Seq	1 1		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
muscle, arm, fetal day105 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, arm, fetal day113 M	**1**		
DNase hypersensitivity	1		
muscle, arm, fetal day115 F	**1**		
DNase hypersensitivity	1		
muscle, arm, fetal day115 M	**2**		
DNase hypersensitivity Exon array	1 1		
muscle, arm, fetal day120 F	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, arm, fetal day120 M	**2**		
DNase hypersensitivity mRNA-Seq	1 1		
muscle, arm, fetal day127 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, arm, fetal day85 F	**2**		
DNase hypersensitivity mRNA-Seq	1 1		
muscle, arm, fetal day91 M	**2**		
Digital genomic footprinting	1	DNase hypersensitivity	1
muscle, arm, fetal day96 M	**5**		
DNase hypersensitivity mRNA-Seq	3 2		
muscle, arm, fetal day97 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, arm, fetal day98 F	**2**		
DNase hypersensitivity mRNA-Seq	1 1		
muscle, back, fetal day101 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, back, fetal day104 M	**1**		
DNase hypersensitivity	1		
muscle, back, fetal day105 F	**2**		
DNase hypersensitivity mRNA-Seq	1 1		
muscle, back, fetal day105 M	**3**		
DNase hypersensitivity	2	mRNA-Seq	1

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
muscle, back, fetal day108 M	**1**		
DNase hypersensitivity	1		
muscle, back, fetal day113 F	**1**		
DNase hypersensitivity	1		
muscle, back, fetal day115 F	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
muscle, back, fetal day127 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, back, fetal day85 F	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
muscle, back, fetal day91 M	**1**		
DNase hypersensitivity	1		
muscle, back, fetal day96 M	**4**		
DNase hypersensitivity	2	mRNA-Seq	2
muscle, back, fetal day98 F	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
muscle, leg, day110 F	**1**		
H3K9me3	1		
muscle, leg, fetal day101 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, leg, fetal day104 M	**1**		
DNase hypersensitivity	1		
muscle, leg, fetal day105 M	**3**		
DNase hypersensitivity	2		
mRNA-Seq	1		
muscle, leg, fetal day110 F	**7**		
ChIP-Seq input	1	H3K27ac	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9me3	1		
muscle, leg, fetal day113 F	**2**		
Bisulfite-Seq	1	DNase hypersensitivity	1
muscle, leg, fetal day113 M	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
muscle, leg, fetal day115 F	**1**		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
DNase hypersensitivity	1		
muscle, leg, fetal day115 M	**2**		
DNase hypersensitivity	1	Exon array	1
muscle, leg, fetal day127 M	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
muscle, leg, fetal day85 F	**1**		
DNase hypersensitivity	1		
muscle, leg, fetal day96 M	**6**		
DNase hypersensitivity	3	mRNA-Seq	3
muscle, leg, fetal day97 M	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
muscle, lower limb, fetal day120 M	**1**		
DNase hypersensitivity	1		
muscle, trunk, fetal day113 F	**1**		
DNase hypersensitivity	1		
muscle, trunk, fetal day115 F	**7**		
ChIP-Seq input	1	H3K27ac	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9me3	1		
muscle, trunk, fetal day120 F	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
muscle, trunk, fetal day121 F	**1**		
DNase hypersensitivity	1		
muscle, upper back, fetal day96 M	**1**		
DNase hypersensitivity	1		
muscle, upper limb, fetal day108 F	**1**		
DNase hypersensitivity	1		
muscle, upper trunk, fetal day108 F	**1**		
DNase hypersensitivity	1		
neurosphere cultured cells, cortex derived	**25**		
Bisulfite-Seq	2		
ChIP-Seq input	2	H3K27me3	2

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
H3K36me3	2	H3K4me1	2
H3K4me3	1	H3K9me3	2
MeDIP-Seq	2	MRE-Seq	2
mRNA-Seq	4	smRNA-Seq	4
neurosphere cultured cells, ganglionic eminence derived	**35**		
Bisulfite-Seq	3		
ChIP-Seq input	4	H3K27ac	1
H3K27me3	3	H3K36me3	3
H3K4me1	3	H3K4me3	2
H3K9me3	3	MeDIP-Seq	2
MRE-Seq	2	mRNA-Seq	4
smRNA-Seq	5		
ovary	**10**		
Bisulfite-Seq	2	ChIP-Seq input	1
DNase hypersensitivity	1	Genotyping array	1
H3K27ac	1	H3K36me3	1
H3K4me1	1	H3K9me3	1
mRNA-Seq	1		
ovary, fetal	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
pancreas	**17**		
Bisulfite-Seq	1		
ChIP-Seq input	2	DNase hypersensitivity	2
H3K27ac	2	H3K27me3	1
H3K36me3	2	H3K4me1	2
H3K4me3	1	H3K9me3	2
mRNA-Seq	2		
pancreatic islets	**18**		
ChIP-Seq input	2	H3K27ac	2
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	3
H3K9ac	1	H3K9me3	2
mRNA-Seq	1	RRBS	1
penis foreskin fibroblast primary cells	**37**		
Bisulfite-Seq	1		
ChIP-Seq input	3	Digital genomic footprinting	1
DNase hypersensitivity	4	H3K27ac	3
H3K27me3	3	H3K36me3	3
H3K4me1	3	H3K4me3	3
H3K9me3	2	MeDIP-Seq	3
MRE-Seq	3	mRNA-Seq	3
smRNA-Seq	2		
penis foreskin keratinocyte primary cells	**41**		
Bisulfite-Seq	2	ChIP-Seq input	3

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
Digital genomic footprinting	1	DNase hypersensitivity	4
Genotyping array	1	H3K27ac	2
H3K27me3	4	H3K36me3	3
H3K4me1	3	H3K4me3	3
H3K9ac	1	H3K9me3	2
MeDIP-Seq	3	MRE-Seq	3
mRNA-Seq	3	smRNA-Seq	3
penis foreskin melanocyte primary cells	**37**		
ChIP-Seq input	3		
Digital genomic footprinting	1	DNase hypersensitivity	4
H3K27ac	3	H3K27me3	3
H3K36me3	3	H3K4me1	3
H3K4me3	3	H3K9me3	2
MeDIP-Seq	3	MRE-Seq	3
mRNA-Seq	3	smRNA-Seq	3
peripheral blood mononuclear primary cells	**31**		
ChIP-Seq input	5		
H3K27ac	2	H3K27me3	3
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9ac	2
H3K9me3	4	MeDIP-Seq	3
MRE-Seq	3	mRNA-Seq	1
smRNA-Seq	2		
placenta, amnion	**10**		
ChIP-Seq input	1	H3K27ac	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9me3	1	mRNA-Seq	2
smRNA-Seq	1		
placenta, basal plate	**4**		
mRNA-Seq	3	smRNA-Seq	1
placenta, chorion smooth	**10**		
ChIP-Seq input	1		
H3K27ac	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1	mRNA-Seq	1
smRNA-Seq	3		
placenta, day 113	**3**		
H3K27ac	1	H3K36me3	1
H3K4me1	1		
placenta, day105	**1**		
DNase hypersensitivity	1		
placenta, day108	**1**		
DNase hypersensitivity	1		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued* ...

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
placenta, day113	**5**		
ChIP-Seq input	1	DNase hypersensitivity	1
H3K27me3	1	H3K4me3	1
H3K9me3	1		
placenta, day85	**2**		
DNase hypersensitivity	2		
placenta, day91	**1**		
DNase hypersensitivity	1		
placenta, trophoblast	**3**		
mRNA-Seq	1	smRNA-Seq	2
placenta, villi	**4**		
mRNA-Seq	1		
smRNA-Seq	3		
psoas muscle	**16**		
Bisulfite-Seq	1	ChIP-Seq input	3
DNase hypersensitivity	1	H3K27ac	3
H3K27me3	1	H3K36me3	1
H3K4me1	2	H3K9me3	1
mRNA-Seq	3		
rectal mucosa	**19**		
ChIP-Seq input	2	Expression array	2
H3K27ac	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	2	H3K9ac	2
H3K9me3	2	RRBS	1
rectal smooth muscle	**10**		
ChIP-Seq input	1		
Expression array	1	H3K27ac	1
H3K27me3	1	H3K36me3	1
H3K4me1	1	H3K4me3	1
H3K9ac	1	H3K9me3	1
RRBS	1		
sigmoid colon	**18**		
Bisulfite-Seq	2	ChIP-Seq input	3
H3K27ac	2	H3K27me3	2
H3K36me3	2	H3K4me1	2
H3K4me3	1	H3K9me3	2
mRNA-Seq	2		
skeletal muscle	**26**		
ChIP-Seq input	3	Expression array	2
H3K27ac	1	H3K27me3	3
H3K36me3	3	H3K4me1	3
H3K4me3	3	H3K9ac	3
H3K9me3	3	RRBS	2

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued ...*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
skeletal muscle, lower limb, fetal day120 M	**1**		
mRNA-Seq	1		
skeletal muscle, upper limb, fetal day108 F	**1**		
mRNA-Seq	1		
skin, abdomen, fetal day97 M	**2**		
DNase hypersensitivity	2		
skin, back, fetal day96 M	**2**		
DNase hypersensitivity	2		
skin, biceps left, fetal day97 M	**2**		
DNase hypersensitivity	2		
skin, biceps right, fetal day97 M	**2**		
DNase hypersensitivity	2		
skin, fetal day82 F	**1**		
DNase hypersensitivity	1		
skin, quadricips left, fetal day97 M	**2**		
DNase hypersensitivity	2		
skin, quadricips right, fetal day97 M	**2**		
DNase hypersensitivity	2		
skin, scalp, fetal day97 M	**2**		
DNase hypersensitivity	2		
skin, upper back, fetal day97 M	**2**		
DNase hypersensitivity	2		
small intestine	**19**		
Bisulfite-Seq	2		
ChIP-Seq input	2	DNase hypersensitivity	1
Genotyping array	1	H3K27ac	3
H3K27me3	1	H3K36me3	2
H3K4me1	2	H3K4me3	1
H3K9me3	2	mRNA-Seq	2
small intestine, fetal day 108 M	**1**		
H3K9me3	1		
small intestine, fetal day105 F	**2**		
DNase hypersensitivity	1		
Exon array	1		
small intestine, fetal day107 F	**1**		

Continued ...

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
Exon array	1		
small intestine, fetal day108 M	**9**		
ChIP-Seq input	1	Digital genomic footprinting	1
DNase hypersensitivity	1	Exon array	1
H3K27ac	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1		
small intestine, fetal day110 F	**2**		
DNase hypersensitivity	1	Exon array	1
small intestine, fetal day115 M	**3**		
DNase hypersensitivity	2		
Exon array	1		
small intestine, fetal day87 M	**1**		
Exon array	1		
small intestine, fetal day91 F	**1**		
Exon array	1		
small intestine, fetal day91 M	**1**		
DNase hypersensitivity	1		
small intestine, fetal, day105 M	**1**		
DNase hypersensitivity	1		
small intestine, fetal, day107 F	**1**		
DNase hypersensitivity	1		
small intestine, fetal, day108 F	**1**		
DNase hypersensitivity	1		
small intestine, fetal, day120 F	**1**		
DNase hypersensitivity	1		
small intestine, fetal, day87 M	**1**		
DNase hypersensitivity	1		
small intestine, fetal, day91 F	**1**		
DNase hypersensitivity	1		
small intestine, fetal, day98 F	**1**		
DNase hypersensitivity	1		
spinal cord, fetal day105 M	**2**		
DNase hypersensitivity	1	mRNA-Seq	1
spinal cord, fetal day113 F	**1**		
DNase hypersensitivity	1		
spinal cord, fetal day87 F	**1**		
DNase hypersensitivity	1		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
spinal cord, fetal day89 F	**1**		
DNase hypersensitivity	1		
spinal cord, fetal day96 M	**2**		
DNase hypersensitivity	1		
mRNA-Seq	1		
spleen	**17**		
Bisulfite-Seq	1	ChIP-Seq input	3
H3K27ac	3	H3K27me3	1
H3K36me3	2	H3K4me1	2
H3K4me3	1	H3K9me3	1
mRNA-Seq	3		
spleen, fetal day112 U	**1**		
DNase hypersensitivity	1		
stomach mucosa	**8**		
ChIP-Seq input	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1	H3K9ac	1
H3K9me3	1	RRBS	1
stomach smooth muscle	**16**		
ChIP-Seq input	2		
Expression array	1	H3K27ac	1
H3K27me3	2	H3K36me3	2
H3K4me1	1	H3K4me3	2
H3K9ac	2	H3K9me3	2
RRBS	1		
stomach, fetal day105 M	**1**		
DNase hypersensitivity	1		
stomach, fetal day108 F	**2**		
DNase hypersensitivity	2		
stomach, fetal day108 M	**1**		
DNase hypersensitivity	1		
stomach, fetal day121 F	**1**		
DNase hypersensitivity	1		
stomach, fetal day91 M	**1**		
DNase hypersensitivity	1		
stomach, fetal day96 F	**6**		
DNase hypersensitivity	2	H3K27ac	1
H3K27me3	1	H3K4me1	1
H3K9me3	1		
stomach, fetal day98 F	**4**		
ChIP-Seq input	1	DNase hypersensitivity	1
H3K36me3	1	H3K4me3	1

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued . . .*

Cell / Tissue - Experiment	Datasets	Cell / Tissue - Experiment	Datasets
stomach, fetal, day101 U	**1**		
DNase hypersensitivity	1		
stomach, fetal, day105 F	**1**		
DNase hypersensitivity	1		
stomach, fetal, day107 F	**1**		
DNase hypersensitivity	1		
stomach, fetal, day127 M	**1**		
DNase hypersensitivity	1		
stomach, fetal, day147 F	**1**		
DNase hypersensitivity	1		
testes, fetal	**2**		
DNase hypersensitivity	2		
testis, spermatozoa primary cells	**2**		
Bisulfite-Seq	2		
Th17 primary cells	**2**		
ChIP-Seq input	1		
H3K9me3	1		
thymus	**8**		
Bisulfite-Seq	2	ChIP-Seq input	1
H3K27ac	1	H3K36me3	1
H3K4me1	1	H3K9me3	1
mRNA-Seq	1		
thymus, fetal day 110 F	**1**		
H3K9me3	1		
thymus, fetal day104 M	**1**	DNase hypersensitivity	1
thymus, fetal day105 F	**2**		
DNase hypersensitivity	1	Exon array	1
thymus, fetal day108 M	**2**		
DNase hypersensitivity	2		
thymus, fetal day110 F	**6**		
ChIP-Seq input	1		
H3K27ac	1	H3K27me3	1
H3K36me3	1	H3K4me1	1
H3K4me3	1		
thymus, fetal day113 F	**2**		
Digital genomic footprinting	1	DNase hypersensitivity	1
thymus, fetal day113 M	**1**		
DNase hypersensitivity	1		

Continued . . .

Number of datasets available for each experiment and each cell lines. *Continued* ...

Cell / Tissue - Experiment	**Datasets**	**Cell / Tissue - Experiment**	**Datasets**
thymus, fetal day115 F	**1**		
Bisulfite-Seq	1		
thymus, fetal day127 M	**1**		
DNase hypersensitivity	1		
thymus, fetal day147 F	**2**		
DNase hypersensitivity	1		
Exon array	1		
thymus, fetal day97 M	**1**		
DNase hypersensitivity	1		
thymus, fetal day98 F	**2**		
DNase hypersensitivity	2		
Treg primary cells	**5**		
ChIP-Seq input	1	H3K27me3	1
H3K36me3	1	H3K4me3	1
H3K9me3	1		
UCSF-4 cell line	**18**		
Bisulfite-Seq	2	ChIP-Seq input	2
H3K27me3	2	H3K36me3	2
H3K4me1	2	H3K4me3	2
H3K9me3	2	mRNA-Seq	2
smRNA-Seq	2		

The End

Bibliography

1. Gligorijević V, Pržulj N. Methods for biological data integration: perspectives and challenges. J R Soc Interface 2015;12(112):20150571.
2. Hawkins RD, Hon GC, Ren B. Next-generation genomics: an integrative approach. Nat Rev Genet 2010;11(7):476–486.
3. Hamid JS, Hu P, Roslin NM, Ling V, Greenwood CMT, Beyene J. Data integration in genetics and genomics: methods and challenges. Hum Genomics Proteomics 2009:869093.
4. Schmidt CW. Data explosion: bringing order to chaos with bioinformatics. Environ Health Perspect 2003;111(6):A340–5.
5. Masseroli M, Kaitoua A, Pinoli P, Ceri S. Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying. Methods 2016;111:3–11.
6. Congressional Justification FY 2017. URL: `https://www.nlm.nih.gov/about/2017CJ.html`.
7. Cook CE, Bergman MT, Finn RD, Cochrane G, Birney E, Apweiler R. The european bioinformatics institute in 2016: data growth and integration. Nucleic Acids Res 2016;44:D20–D26.
8. DNA Sequencing Costs: Data. URL: `https://www.genome.gov/27541954/dna-sequencing-costs-data/`.
9. Data-driven genomic computing (GeCo). URL: `http://www.bioinformatics.deib.polimi.it/geco/?home`.
10. Stanford University. ENCODE website. URL: `https://www.encodeproject.org/`.
11. About TCGA. URL: `https://cancergenome.nih.gov/abouttcga`.
12. Data-driven genomic computing: approach. URL: `http://www.bioinformatics.deib.polimi.it/geco/?approach`.
13. GMQL Introduction to the language. URL: `http://www.bioinformatics.deib.polimi.it/geco/documentation/GMQL_Introduction_to_the_language.pdf`.
14. Masseroli M, Pinoli P, Venco F, Kaitoua A, Jalili V, Palluzzi F, et al. GenoMetric Query Language: a novel approach to large-scale genomic data management. Bioinformatics 2015;31(12):1881–1888.
15. Kaitoua A, Pinoli P, Bertoni M, Ceri S. Framework for supporting genomic operations. IEEE-TC 2012:2603980.

16. NIH Roadmap Epigenomics Mapping Consortium. Roadmap Epigenomics Project website. URL: http://www.roadmapepigenomics.org/.

17. Wikipedia DNA methylation. URL: https://en.wikipedia.org/wiki/DNA_methylation.

18. Jinbb B, Li Y, Robertson K. DNA Methylation. Superior or subordinate in the epigenetic hierarchy? Genes Cancer 2011:607–617.

19. Histone Modifications. URL: https://www.whatisepigenetics.com/histone-modifications/.

20. Zhang C. Novel functions for small RNA molecules. Curr Opin Mol Ther 2009:641–651.

21. GMQL-Importer. URL: https://github.com/DEIB-GECO/GMQL-Importer.

22. Pena JIV. Automation of retrieval, transformation and uploading of genomic data and their metadata for their integration into a GDM repository. Master thesis. Politecnico di Milano, 2017.

23. Coarfa C, Yu F, Miller C, Chen Z, Harris R, Milosavljevic A. Pash 3.0: A versatile software package for read mapping and integrative analysis of genomic and epigenomic variation using massively parallel DNA sequencing. BMC Bioinformatics 2010:572.

24. Chromatin immunoprecipitation Wikipedia web page. URL: https://en.wikipedia.org/wiki/Chromatin_immunoprecipitation.

25. DNase I Demystified. URL: http://www.thermofisher.com/it/en/home/references/ambion-tech-support/nuclease-enzymes/general-articles/dnase-i-demystified.html.

26. Bisulfite sequencing Wikipedia web page. URL: https://en.wikipedia.org/wiki/Bisulfite_sequencing.

27. BCM Roadmap Epigenomics EDACC. URL: http://genboree.org/EdaccData/Release-9/.

28. Epigenome Atlas FTP server for Roadmap data. URL: ftp://ftp.genboree.org/EpigenomeAtlas/Current-Release/.

29. NCBI - GEO FTP server for Roadmap Epigenomic data. URL: ftp://ftp.ncbi.nlm.nih.gov/pub/geo/DATA/roadmapepigenomics/.

30. NIH Roadmap Epigenomics - GEO - NCBI. URL: https://www.ncbi.nlm.nih.gov/geo/roadmap/epigenomics/.

31. Simon J, Kingston R. Mechanisms of Polycomb gene silencing: knowns and unknowns. Nat Rev Mol Cell Biol 2009;10:697–708.

32. Heterochromatin Wikipedia web page. URL: https://en.wikipedia.org/wiki/Heterochromatin.

33. NIH Roadmap Epigenomics Mapping Consortium. Criterion For Complete Epigenome Classification. URL: http://www.roadmapepigenomics.org/complete_epigenomes/.

34. Supplementary website for the 2015 Consortium paper. URL: http://egg2.wustl.edu/roadmap/web_portal/.

35. Kundaje A, Meuleman W, Ernst J, Bilenky M, Yen A, Heravi-Moussavi A, et al. Integrative analysis of 111 reference human epigenomes. Nature 2015;(518):317–349.

36. Bioinformatics Research Laboratory of Baylor College of Medicine. Epigenome Atlas website. URL: http://www.genboree.org/epigenomeatlas/index.rhtml.

37. SRA Wikipedia web page. URL: https://en.wikipedia.org/wiki/Sequence_Read_Archive.

38. NCBI. SRA website. URL: https://www.ncbi.nlm.nih.gov/sra.

39. NCBI. dbGaP website. URL: https://www.ncbi.nlm.nih.gov/gap/.

40. NIH Roadmap Epigenomics Mapping Consortium. Epigenomics Data Analysis and Coordination Center (EDAAC). URL: http://www.roadmapepigenomics.org/overview/edaac.

41. NCBI. Sequence Viewer. Version 3.23.0. URL: https://www.ncbi.nlm.nih.gov/projects/sviewer/.

42. UCSC. Genome Browser. URL: https://genome.ucsc.edu/.

43. NCBI. SRA FTP server. URL: ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/.

44. Affymetrix CEL Data File Format. URL: http://dept.stat.lsa.umich.edu/~kshedden/Courses/Stat545/Notes/AffxFileFormats/cel.html.

45. fileinfo .cel file extension. URL: https://fileinfo.com/extension/cel.

46. UCSC. BAM format. URL: https://genome.ucsc.edu/FAQ/FAQformat.html#format5.1.

47. Sequence Alignment/Map Format Specification. URL: http://samtools.github.io/hts-specs/SAMv1.pdf.

48. Affymetrix Annotation Converter 1.0. URL: https://tools.thermofisher.com/content/sfs/manuals/annotation_converter_1_0_user_manual.pdf.

49. UCSC. Wiggle Track Format (WIG). URL: https://genome.ucsc.edu/goldenPath/help/wiggle.html.

50. UCSC. BED format. URL: http://genome.ucsc.edu/FAQ/FAQformat#format1.

51. EMBL-EBI. BED File Format - Definition and supported options. 2017. URL: https://www.ensembl.org/info/website/upload/bed.html.

52. Baylor College of Medicine. Genboree web site. URL: http://genboree.org/site/.

53. Admins EF. DATA RELEASE POLICY. 2014. URL: ftp://ftp.genboree.org/EpigenomeAtlas/Current-Release/-DATA%20RELEASE%20POLICY.

54. ENCODE tagAlign: BED3+3 format (historical). URL: https://genome.ucsc.edu/FAQ/FAQformat.html#format15.

55. ENCODE narrowPeak: Narrow (or Point-Source) Peaks format. URL: http://genome.ucsc.edu/FAQ/FAQformat.html#format12.

56. ENCODE broadPeak: Broad Peaks (or Regions) format. URL: http://genome.ucsc.edu/FAQ/FAQformat.html#format13.

57. ENCODE gappedPeak: Gapped Peaks (or Regions) format. URL: http://genome.ucsc.edu/FAQ/FAQformat.html#format14.

58. Reduced representation bisulfite sequencing. URL: https://en.wikipedia.org/wiki/Reduced_representation_bisulfite_sequencing.

59. Mohn F, Weber M, Schübeler D, Roloff T. Methylated DNA immunoprecipitation (MeDIP). Methods Mol Biol 2009;507:55–64.

60. Methylation Sensitive Restriction Enzymes for Epigenetics. URL: http://www.clontech.com/CA/Products/Cell_Biology_and_Epigenetics/Epigenetics/DNA_Preparation/MSRE_Overview.

61. methylCRF. Combining MeDIP-seq and MRE-seq to estimate single CpG methylation genome wide. URL: http://methylcrf.wustl.edu/.

62. ChromHMM: Chromatin state discovery and characterization. URL: http://compbio.mit.edu/ChromHMM/.

63. wustl roadmap data repository. URL: http://egg2.wustl.edu/roadmap/data/.

64. Google spreadsheet: Metadata and quality control. URL: https://docs.google.com/spreadsheets/d/1yikGx4MsO9Ei36b64yOy9Vb6oPC5IBGlFbYEt-N6gOM/edit?usp=sharing.

65. Introduction to the Google Sheets API. 2017. URL: https://developers.google.com/sheets/api/guides/concepts?authuser=1.

66. Quality Metrics. URL: https://genome.ucsc.edu/ENCODE/qualityMetrics.html.

67. GEOarchive metadata guidelines table. 2017. URL: https://www.ncbi.nlm.nih.gov/geo/info/spreadsheet.html#GAmeta.

68. E001-H3K4me1.broadPeak.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/broadPeak/E001-H3K4me1.broadPeak.gz.

69. E010-H2A.Z.gappedPeak.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/gappedPeak/E010-H2A.Z.gappedPeak.gz.

70. E115-H3K27me3.narrowPeak.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/narrowPeak/E115-H3K27me3.narrowPeak.gz.

71. Hammock. URL: http://wiki.wubrowse.org/Hammock#gappedPeak.

72. Li H. Tabix: fast retrieval of sequence features from generic TAB-delimited files. Bioinformatics 2011;(27):718–719.

73. E001-H3K4me1.broadPeak.hammock.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/broadPeak/hammock/E001-H3K4me1.broadPeak.hammock.gz.

74. E003-DNase.macs2.narrowPeak.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/narrowPeak/E003-DNase.macs2.narrowPeak.gz.

75. E003-DNase.hotspot.fdr0.01.peaks.bed.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/narrowPeak/E003-DNase.hotspot.fdr0.01.peaks.bed.gz.

76. E003-DNase.hotspot.all.peaks.bed.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/narrowPeak/E003-DNase.hotspot.all.peaks.bed.gz.

77. E003-DNase.hotspot.fdr0.01.broad.bed.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/broadPeak/E003-DNase.hotspot.fdr0.01.broad.bed.gz.

78. E003-DNase.hotspot.broad.bed.gz. URL: http://egg2.wustl.edu/roadmap/data/byFileType/peaks/consolidated/broadPeak/E003-DNase.hotspot.broad.bed.gz.

79. bigWig Track Format. URL: https://genome.ucsc.edu/goldenpath/help/bigWig.html.

80. 57epigenomes.N.nc.gz. URL: http://egg2.wustl.edu/roadmap/data/byDataType/rna/expression/57epigenomes.N.nc.gz.

81. 57epigenomes.exon.N.nc.gz. URL: http://egg2.wustl.edu/roadmap/data/byDataType/rna/expression/57epigenomes.exon.N.nc.gz.

82. Ensembl_v65.Gencode_v10.ENSG.gene_info. URL: http://egg2.wustl.edu/roadmap/data/byDataType/rna/expression/Ensembl_v65.Gencode_v10.ENSG.gene_info.

83. Ensembl. URL: http://www.ensembl.org/index.html.

84. Contig. URL: https://en.wikipedia.org/wiki/Contig#cite_ref-contig_assembly_1-0.

85. README. URL: http://egg2.wustl.edu/roadmap/data/byDataType/rna/intergenic_contigs/README.

86. BedGraph Track Format. URL: https://genome.ucsc.edu/goldenpath/help/bedgraph.html.

87. RNAseq_intergenic.tar.gz. URL: http://egg2.wustl.edu/roadmap/data/byDataType/rna/intergenic_contigs/RNAseq_intergenic.tar.gz.

88. RNAseq_intergenic_summary.xls. URL: http://egg2.wustl.edu/roadmap/data/byDataType/rna/intergenic_contigs/RNAseq_intergenic_summary.xls.

89. DMR finding. URL: http://egg2.wustl.edu/roadmap/data/byDataType/dnamethylation/DMRs/DMR%20Finding.pdf.

90. E001_RRBS_DMRs_v2.bed.gz. URL: http://egg2.wustl.edu/roadmap/data/byDataType/dnamethylation/DMRs/RRBS_bed/E001_RRBS_DMRs_v2.bed.gz.

91. jsoup: Java HTML Parser. URL: https://jsoup.org/.

92. OAuth 2.0. URL: https://oauth.net/2/.

93. Hardt D, ed. The OAuth 2.0 Authorization Framework. 2012. URL: https://tools.ietf.org/html/rfc6749.

94. Using OAuth 2.0 to Access Google APIs. URL: https://developers.google.com/identity/protocols/OAuth2.

95. scala-csv. URL: https://github.com/tototoshi/scala-csv.

96. Scala docs: collections performance characteristics. URL: https://docs.scala-lang.org/overviews/collections/performance-characteristics.html.

97. Intel® Xeon® Processor E5-2650. URL: https://ark.intel.com/products/64590/Intel-Xeon-Processor-E5-2650-20M-Cache-2_00-GHz-8_00-GTs-Intel-QPI.