

Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying

Marco Masseroli[§], Abdulrahman Kaitoua, Pietro Pinoli, Stefano Ceri

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20133 Milano, Italy

[§] Corresponding author

Email addresses:

MM: marco.masseroli@polimi.it

AK: abduhrahman.kaitoua@polimi.it

PP: pietro.pinoli@polimi.it

SC: stefano.ceri@polimi.it

Abstract

While a huge amount of (epi)genomic data of multiple types is becoming available by using Next Generation Sequencing (NGS) technologies, the most important emerging problem is the so-called *tertiary analysis*, concerned with sense making, e.g., discovering how different (epi)genomic regions and their products interact and cooperate with each other. We propose a paradigm shift in tertiary analysis, based on the use of the Genomic Data Model (GDM), a simple data model which links genomic feature data to their associated experimental, biological and clinical metadata. GDM encompasses all the data formats which have been produced for feature extraction from (epi)genomic datasets. We specifically describe the mapping to GDM of SAM (Sequence Alignment/Map), VCF (Variant Call Format), NARROWPEAK (for called peaks produced by NGS ChIP-seq or DNase-seq methods), and BED (Browser Extensible Data) formats, but GDM supports as well all the formats describing experimental datasets (e.g., including copy number variations, DNA somatic mutations, or gene expressions) and annotations (e.g., regarding transcription start sites, genes, enhancers or CpG islands). We downloaded and integrated samples of all the above-mentioned data types and formats from multiple sources. The GDM is able to homogeneously describe semantically heterogeneous data and makes the ground for providing data interoperability, e.g., achieved through the GenoMetric Query Language (GMQL), a high-level, declarative query language for genomic big data. The combined use of the data model and the query language allows comprehensive processing of multiple heterogeneous data, and supports the development of domain-specific data-driven computations and bio-molecular knowledge discovery.

Keywords: Genomic data management; Data modeling; Data interoperability; Metadata management; Query languages; Operations for genomics

1. Introduction

Extraordinary advances in genomics are made possible by Next Generation Sequencing (NGS), a family of technologies that is progressively reducing the time and cost of reading an individual genome¹; therefore, huge amounts of sequencing data of many genomes, in multiple biological and clinical conditions, are continuously collected and made publicly available, often organized by worldwide consortia such as ENCODE [1], Roadmap Epigenomics [2], TCGA [3] and the 1000 Genomes Project [4].

So far, the bioinformatics research community has been mostly challenged by *primary analysis* (production of sequences in the form of short DNA or RNA segments, or "reads") and *secondary analysis* (alignment of reads to a reference genome and search for specific features of genome regions, such as variants and peaks of binding or expression intensities) [5]. The most important emerging problem is the so-called *tertiary analysis* [6], concerned with sense making, e.g., discovering how different (epi)genomic regions and their products interact and cooperate with each other. Tertiary analysis requires integrating heterogeneous DNA features, such as variations (e.g., a mutation in a given DNA position), or peaks of binding or expression (i.e., genomic regions with higher read density), or structural properties of

¹ Recently below the barrier of 1000 US\$ for a human genome

(<https://www.genome.gov/sequencingcosts/>).

the DNA (e.g., break points, where the DNA is damaged, or junctions, where the DNA creates loops). Such data are collected within numerous and heterogeneous files (both in formats and semantics); they are usually distributed within different repositories, and lack an attribute-based organization for systematically expressing features as high-level attributes. Furthermore, they lack a systematic description of their metadata, i.e., of their biological and clinical properties, which are greatly heterogeneous. Answers to crucial biomedical questions may be hidden within already existing open and public collections of heterogeneous data, but the methods and tools which are made available for knowledge extraction are still rather poor and specialized.

We propose a paradigm shift in tertiary genomic data management, based on the introduction of a simple data model which links genomic features to their associated metadata. This model is able to homogeneously describe semantically heterogeneous data and makes the ground for providing data interoperability, which can be achieved through a high-level, declarative query language for genomic big data. The combination of the data model and query language provides the right concepts for information extraction from genomic data repositories, and allows the development of domain-specific data-driven computations required by tertiary data analysis and bio-molecular knowledge discovery.

2. Genomic Data Model

The Genomic Data Model (GDM) that we propose is based on the notions of *datasets* and *samples*, and on two abstractions: one for *genomic regions*, which represent portions of the DNA and their features, and one for their *metadata*. Datasets are collections of samples, and each sample consists of two parts: the *region data*, which

describe the characteristics and DNA location of genomic features (e.g., called though the processing of raw NGS data after their alignment to a reference genome), and the *metadata*, which describe general properties of the sample.

2.1 Motivation

Genomic region/feature data are very valuable for molecular investigation and precision medicine; they describe a broad variety of molecular aspects, which are individually measured, and provide single views on biomolecular phenomena. Their integrated evaluation would provide a systemic view on how they interact and cooperate towards the triggering and regulation of biological functions. Yet, they are available in a variety of formats which hamper their integration and comprehensive assessment.

GDM provides a schema to genomic feature data of DNA regions; thus, it makes such heterogeneous data self-describing, as advocated by Jim Gray [7], and interoperable. This is obtained by simple mapping of the data from data files in their original format into the GDM format when they are used, without including them into a database, so as to preserve the possibility for biologists to work with their usual file-based tools. The provided data schema has a fixed part, which guarantees the comparability of regions produced by different kinds of processing, and a variable part reflecting the “feature calling process” that produced the regions and describing the region features determined through various processing types. DNA regions are sequences of nucleotides², usually represented by strings of letters³; GDM identifies

² Nucleotides are the individual molecular components of the DNA macromolecule, and are of four different types (Adenine, Cytosine, Guanine, and Thymine).

them through their genomic coordinates and associates them with a list of one or more features (e.g., produced by NGS data secondary analysis).

Metadata are paramount to characterize the high heterogeneity of genomic feature data and guide their correct processing; however, they are collected in a broad variety of data structures and formats that constitute barriers to their use and comparison. To cope with the lack of agreed standards for metadata, GDM models metadata simply as free arbitrary semi-structured attribute-value pairs, where attributes may have multiple values (e.g., the *Disease* attribute can have both “*Cancer*” and “*Diabetes*” values). We expect metadata to include at least the considered organism, tissue, cell line, experimental condition (e.g., antibody target – in the case of NGS ChIP-seq experiments, treatment, etc.), experiment type, data processing performed, feature calling and analysis method used for the production of the related data; in the case of clinical studies, individual's descriptions including phenotypes.

2.2 Definitions

A *genomic region* r is a well-defined portion of the genome identified by the quadruple of values $\langle chr, left, right, strand \rangle$, called *region coordinates*, where chr represents the DNA chromosome where the region is located, $left$ and $right$ are the positions of the two ends of the region along the DNA coordinates⁴; $strand$ indicates

³ DNA can be abstracted as a string of billions of four different letters (A, C, G, T), each representing a nucleotide molecule, subdivided in chromosomes (23 in humans), which are disconnected intervals of the string.

⁴ Species are associated with their *reference genome*. DNA samples are aligned to these references, hence referred to the same system of coordinates; for humans, several reference genomes were progressively defined, the latest is *hg20* (also known as *GRCh38* or *hg38*).

the DNA strand on which the region is read, as well as the direction of DNA reading⁵ (encoded as either '+' or '-'), and can be missing (encoded as '*') when the region is not assigned to a specific strand, e.g., in the case of DNA binding regions identified through NGS ChIP-seq experiments).

A *sample s* is formally modeled as a triple $\langle id; R; M \rangle$ where:

- *id* is the sample *identifier* of type *long*
- *R* is the set of *regions* of the sample, built as pairs $\langle c; f \rangle$ of *coordinates c* and *features f*. Coordinates are composed of four fixed attributes *chr*, *left*, *right*, *strand* which are respectively typed *string*, *long*, *long*, *char*. Features are made of typed attributes; we assume attribute names of features to be different, and their types to be any of *Boolean*, *char*, *string*, *int*, *long*, *double* (GDM types are available in several programming languages, including Java and Scala, and frameworks for cloud computing, such as Apache Pig⁶, Apache Flink⁷ and Apache Spark⁸). The *region schema* of *s* is the list of attribute names used for the identifier, the coordinates and the features.

According to the University of California at Santa Cruz (UCSC) notation, we use *0-based, half-open inter-base coordinates*, i.e., the considered genomic sequence is [*left*; *right*). In this coordinate system, left and right ends can be identical (e.g., when they represent a *splicing junction*), or consecutive (e.g., when the region represents a *single nucleotide polymorphism*).

⁵ DNA is made of two strands rolled-up together in anti-parallel directions, i.e., they are read in opposite directions by the biomolecular machinery of the cell.

⁶ <https://pig.apache.org/>

⁷ <http://flink.apache.org/>

⁸ <http://spark.apache.org/>

- M is the set of *metadata* of the sample, built as *attribute-value* pairs $\langle a; v \rangle$, where we assume the type of each value v to be *string* (numerical values can then be casted to a numerical type, such as *int*, *long*, or *double*, when used). The same attribute name a can appear in multiple pairs of the same sample (in which case we say that a is multi-valued).

A *dataset* is a collection of samples with the same region schema and with features having the same types; sample identifiers are unique within each dataset. Each dataset can be thought as grouping related data samples, in case produced within the same project (either at a genomic research center or within an international consortium) by using the same or equivalent technology and tools, but with different experimental conditions, described by metadata.

2.3 Implementation example

According to GDM, each dataset can be stored using two data structures (e.g., two tables), one for regions and one for metadata. An example of two tables for representing a particular experiment, called *ChIP-seq*, is shown in Fig. 1, where two small samples are represented. Sample 1 has 3 regions and 4 metadata attributes, sample 2 has 2 regions and 3 metadata attributes; the regions of the two samples are within chromosomes 1 and 2 of the DNA, and both are not stranded. The region features have an attribute *p_value* of type *double*, representing how significant is the calling of that genomic region in the *ChIP-seq* experiment. Note that the *id* attribute is present in both tables; it provides a many-to-many connection between regions and metadata of a sample. Note also that the quintuple (*id*, *chr*, *left*, *right*, *strand*) is not a key of the region table, since a sample can have multiple regions with the same coordinates; similarly the pair (*id*, *attribute*) is not a key of the metadata table, since metadata attributes can be multi-valued.

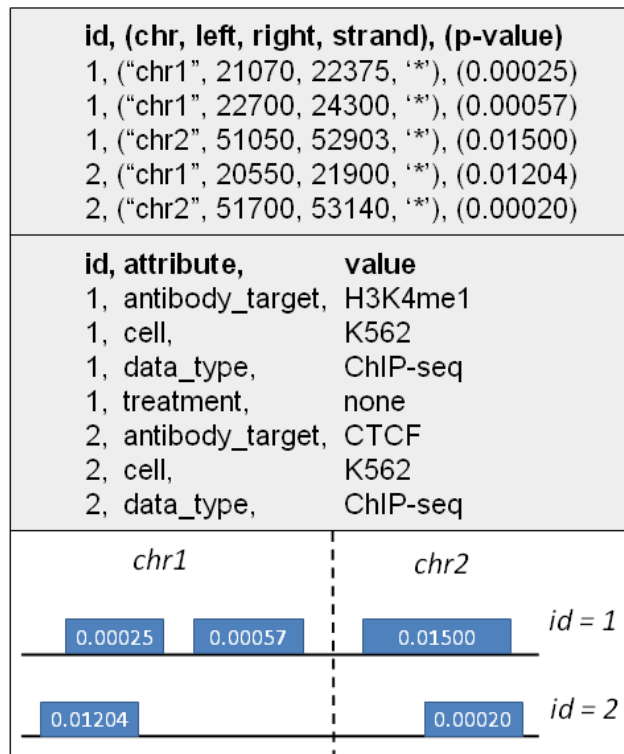


Fig. 1. Regions (top part) and metadata (central part) of a dataset consisting of two ChIP-seq samples (bottom part), respectively having three and two regions, and four and three metadata.

While the above example is simple, GDM supports the schema encoding of any processed data type (e.g., of NGS DNA-seq, RNA-seq, ChIP-seq, ChIA-PET, and VCF data file formats), and of any aligned genomic data in general (e.g., in Sequence Alignment/Map - SAM format), since all of them share the genomic region concept. Note that GDM can also model structural and functional annotations, i.e., regions of the genome with known properties (such as genes, with their exons and introns⁹ as well as functions). Examples of GDM modeling of different types of aligned genomic

⁹ Genes of eukaryotic organisms are mainly composed of two parts: *exons*, which encode gene transcripts (RNA) and proteins, and *introns*, which are noncoding sections of a gene.

data and their formats, together with exemplar instances of these data types, are described in the following Section 2.4.

2.4 Modeling aligned genomic data in GDM

In order to ease modeling any aligned genomic data in GDM and mapping the variety of their multiple formats to GDM, we defined how to simply describe them in terms of GDM data schema by using the Extensible Markup Language (XML). Figs. 2-5 show paradigmatic examples regarding four different genomic data types of high relevance, in four typical and highly used tab-delimited text formats. They include:

- **Aligned sequence data** (e.g., NGS nucleotide reads aligned to a reference genome) in **SAM** (Sequence Alignment/Map) format [8], as usually outputted from aligners (e.g., BWA [9] or Bowtie [10]) that read nucleotide sequences in FASTQ files [11] generated by NGS machines and assign the read sequences to a position with respect to a known reference genome
- **DNA variation data** (e.g., single nucleotide variants, insertions/deletions, copy number variants and structural variants), such as those generated through the NGS DNA-seq technique, in **VCF** (Variant Call Format) format [12], as typically provided by the 1000 Genomes Project [4]
- **Called peaks** (i.e., genomic regions of biomolecular signal enrichment called through multiple specific methods and tools, such as MACS [13] or ZIMBA [14]) representing genomic features (e.g., DNA hypersensitive sites in open chromatin regions, or histone modifications and transcription factor binding sites, determined through the NGS techniques of DNase I sequencing (DNase-seq) [15] or Chromatin Immunoprecipitation followed by sequencing (ChIP-seq) [16], respectively) in **NARROWPEAK** format, as typically provided by the ENCODE [1] and Roadmap Epigenomics [2] projects

- **CpG Island annotations** (i.e., known DNA regions where a cytosine nucleotide is followed by a guanine nucleotide in the linear sequence of bases along the 5' -> 3' direction) in a kind of **BED** (Browser Extensible Data) format [17], as provided by the world-wide recognized Genome Informatics Group of the University of California at Santa Cruz (UCSC) [18].

The GDM encoding of these data formats is next progressively discussed in Figs. 2-5; each of them includes (from top to bottom) a brief description of the represented data/format, the structure and fields of the original data format together with two or three exemplar data lines, the corresponding XML description of the data in terms of GDM data schema, and the related GDM structure and attributes together with two or three exemplar lines of the same data. Note that the XML description of the data includes, in its *gdmSchema* tag, the attribute *type*, which specifies the handle label for the specific data format; it allows the binding to a software loader which can be used to automatically map the data in their original format (detailed in the XML description) to the GDM format when they are used. The high flexibility provided by the defined XML description, and the associated loader, can manage and accommodate multiple different situations occurring in the heterogeneous data structures of the variety of data formats currently used in genomics. For example, Fig. 2 shows that the region coordinate attributes required in GDM can be mapped to any fields (with any names) in the structure of the original data format. Any field in the original data format is described by a *field* XML tag whose position in the XML description is equal to the position of the field in the original data format, and the value included in the *field* tag equals the name of the matching attribute in GDM (the *type* attribute of the *field* tag specifies the data type of the matching attribute in

GDM). Thus, it is straightforward to identify any original data field with a normalized name (and position, if required) in GDM, which open the way to data interoperability.

When the original data do not include all required GDM region coordinate attributes, but their values can be derived from other original data fields (as usual in aligned genomic data), the loader associated with the XML description of the data can provide them at data usage time. For example, in the original data in SAM format described in Fig. 2, a matching for the GDM *right* attribute is missing; however, for any described genomic region (i.e., data line) the associated loader derives the value of the *right* attribute as the sum of the value of the *left* attribute (i.e., of the *POS* original data field) and of the length of the sequence string in the *SEQ* attribute / original data field (i.e., as $left + length(SEQ)$) of the genomic region. (Note that the *strand* attribute missing in the original data is encoded as '*' in GDM, as defined in Section 2.2). At data usage time, if required, the loader can also convert values in the original data to normalized values in GDM in order to support seamless integration of data in different datasets (e.g., see Fig. 4 where the original value '.' for an undefined strand is converted to '*'; a more significant example, in data from NGS RNA-seq techniques, is the conversion of gene expression values from TPM (Transcripts Per Million) to FPKM (Fragments Per Kilobase of transcript per Million mapped reads) units).

SAM format: It is a text format for storing sequence data in a series of tab delimited ASCII columns.

QNAME	FLAG	RNAME	POS	MAPQ	CIGAR	RNEXT	PNEXT	TLEN	SEQ	QUAL
r001	99	ref	7	30	8M2I4M1D3M	=	37	39	TTAGATAAAGGATACTG	*
r002	0	ref	9	30	3S6M1P1I4M	*	0	0	AAAAGATAAGGATA	*
r001	147	ref	37	30	9M	=	7	-39	CAGCGGCAT	* NM:i:1

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<gdmSchemaCollection xmlns = "http://www.bioinformatics.deib.polimi.it/GDM/"
  name = "global_schemas">
  <gdmSchema type = "SAM">
    <field type = "string">QNAME</field> // Query template NAME
    <field type = "string">FLAG</field> // Bitwise FLAG
    <field type = "string">chr</field> // RNAME: Reference sequence NAME
    <field type = "long">left</field> // POS: 1-based leftmost mapping POSITION
    <field type = "int">MAPQ </field> // MAPPING Quality
    <field type = "string">CIGAR</field> // CIGAR string
    <field type = "string">RNEXT</field> // Reference name of the mate/next read
    <field type = "long">PNEXT</field> // Position of the mate/next read
    <field type = "long">TLEN</field> // Observed template LENGTH
    <field type = "string">SEQ</field> // Segment SEQUENCE
    <field type = "string">QUAL</field> // ASCII of Phred-scaled base QUALity+33
  </gdmSchema>
</gdmSchemaCollection>

(id, (chr, left, right, strand), (QNAME, FLAG, MAPQ, CIGAR, RNEXT, PNEXT, TLEN, SEQ, QUAL))
(1, ("ref", 7, 24, '*'), ("r001", "99", 30, "8M2I4M1D3M", "=", 37, 39, "TTAGATAAAGGATACTG", "**"))
(1, ("ref", 9, 23, '*'), ("r002", "0", 30, "3S6M1P1I4M", "*", 0, 0, "AAAAGATAAGGATA", "**"))
(1, ("ref", 37, 46, '*'), ("r001", "147", 30, "9M", "=", 7, -39, "CAGCGGCAT", "* NM:i:1"))
```

Fig. 2. Aligned sequence data (e.g., NGS nucleotide reads aligned to a reference genome) in SAM (Sequence Alignment/Map) format and their GDM description and schema.

VCF (Variant Call Format) format: It is a flexible and extendable line-oriented text format developed by the 1000 Genomes Project for releases of single nucleotide variants, indels, copy number variants and structural variants.

CHROM	POS	ID	REF	ALT	QUAL	FILTER
22	16050075	.	A	G	100	PASS
22	16050678	rs139377059	C	T	100	PASS

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<gdmSchemaCollection xmlns = "http://www.bioinformatics.deib.polimi.it/GDM/"
  name = "global_schemas">
  <gdmSchema type = "VCF">
    <field type = "string">chr</field> // Name of reference sequence (e.g., a chromosome) on which
      the variation is being called
    <field type = "long">left</field> // Starting position of the variation on the given reference sequence
    <field type = "string">id</field> // The identifier of the variation (e.g., a dbSNP rs identifier or "." if
      unknown)
    <field type = "string">ref</field> // The reference base (or bases in the case of an InDel) at the
      given position on the given reference sequence
    <field type = "string">alt</field> // The list of alternative alleles at the given position
    <field type = "int">qual</field> // A quality score associated with the inference of the given alleles
    <field type = "string">filter</field> // A flag indicating which of a given set of filters the variation has
      passed
  </gdmSchema>
</gdmSchemaCollection>

(id, (chr, left, right, strand), (id, ref, alt, qual, filter))
(1, ("22", 16050075, 16050075, '*'), (".", "A", "G", 100, "PASS"))
(1, ("22", 16050678, 16050678, '*'), ("rs139377059", "C", "T", 100, "PASS"))
```

Fig. 3. DNA variation data (e.g., single nucleotide variants, insertions/deletions, copy number variants and structural variants) in VCF (Variant Call Format) format and their GDM description and schema.

ENCODE NARROW (or point source) **PEAK** format: It is used for called regions of signal enrichment based on pooled, normalized (interpreted) data, which usually represent genomic features.

chrom	chromStart	chromEnd	name	score	strand	signalValue	pValue	qValue	peak
chr1	9356548	9356648	.	0	.	182	5.0945	-1	50
chr1	9358722	9358822	.	0	.	91	4.6052	-1	40

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<gdmSchemaCollection xmlns = "http://www.bioinformatics.deib.polimi.it/GDM/"
  name = "global_schemas">
  <gdmSchema type = "NARROWPEAK">
    <field type = "string">chr</field> // Name of reference sequence chromosome or scaffold
    <field type = "long">left</field> // Starting position of the feature in the chromosome or scaffold
    <field type = "long">right</field> // Ending position of the feature in the chromosome or scaffold
    <field type = "string">name</field> // Feature / region name ( '.' if not assigned)
    <field type = "int">score</field> // Feature score (how dark the region is shown in a genome
      browser (0-1000))
    <field type = "char">strand</field> // Chromosome strand
    <field type = "double">signalvalue</field> // Overall (usually, average) enrichment for the region
    <field type = "double">pvalue</field> // Statistical significance (-log10) for the region (-1 if not
      assigned)
    <field type = "double">qvalue</field> // Statistical significance using false discovery rate (-log10)
      for the region (-1 if not assigned)
    <field type = "int">peak</field> // Point-source called for the region: 0-based offset from region left
      end (-1 if not assigned)
  </gdmSchema>
</gdmSchemaCollection>

(id, (chr, left, right, strand), (name, score, signalvalue, pvalue, qvalue, peak))
(1, ("chr1", 9356548, 9356648, '*'), (".", 0, 182, 5.0945, -1, 50))
(1, ("chr1", 9358722, 9358822, '*'), (".", 0, 91, 4.6052, -1, 40))
```

Fig. 4. Called peaks of biomolecular signal enrichment in NARROWPEAK format and their GDM description and schema.

CpG Islands (UCSC) annotations: They are regions of DNA where a cytosine nucleotide is followed by a guanine nucleotide in the linear sequence of bases along the 5' -> 3' direction, provided in a kind of **BED** (Browser Extensible Data) format.

Chrom	chromStart	chromEnd	name	length	cpgNum	gcNum	perCpg	perGc	obsExp
chr1	28735	29810	CpG: 116	1075	116	787	21.6	73.2	0.83
chr1	135124	135563	CpG: 30	439	30	439	13.7	67.2	0.64

```
<?xml version = "1.0" encoding = "UTF-8" standalone = "yes"?>
<gdmSchemaCollection xmlns = "http://www.bioinformatics.deib.polimi.it/GDM/"
  name = "global_schemas">
  <gdmSchema type = "CpG">
    <field type = "string">chr</field> // Name of reference sequence chromosome or scaffold
    <field type = "long">left</field> // Starting position of the feature in the chromosome or scaffold
    <field type = "long">right</field> // Ending position of the feature in the chromosome or scaffold
    <field type = "string">name</field> // CpG island name
    <field type = "long">length</field> // Island length (right - left)
    <field type = "long">cpgNum</field> // Number of CpGs in island
    <field type = "long">gcNum</field> // Number of C and G in island
    <field type = "double">perCpG</field> // Percentage of island that is CpG
    <field type = "double">perGC</field> // Percentage of island that is C or G
    <field type = "double">obsExp</field> // Ratio of observed (cpgNum) to expected (numC * numG /
      length) CpGs in island
  </gdmSchema>
</gdmSchemaCollection>

(id, (chr, left, right, strand), (name, length, cpgNum, gcNum, perCpG, perGC, obsExp))
(1, ("chr1", 28735, 29810, '*'), ("CpG: 116", 1075, 116, 787, 21.6, 73.2, 0.83))
(1, ("chr1", 135124, 135563, '*'), ("CpG: 30", 439, 30, 439, 13.7, 67.2, 0.64))
```

Fig. 5. CpG Island annotations in a kind of BED (Browser Extensible Data) format and their GDM description and schema.

3. Interoperability and integrative querying of heterogeneous genomic feature data

Taking advantage of the GDM characteristics and of the defined XML data description, we modeled several different human and mouse genomic feature datasets in GDM, after downloading them from multiple public sources, such as the ENCODE [1] and Roadmap Epigenomics [2] portals. They comprised processed data in BROADPEAK, NARROWPEAK and BED formats from many types of experiments, including DNA methylation (methylArray, and methylRRBS), open chromatin (DNase-DGF, DNase-seq, and FAIRE-seq), transcription factor binding site and histone modification (ChIP-seq), RNA binding protein (RIPGeneST), RNA profiling (exonArray, RNA-chip, RNA-PET, and RNA-seq), and other (e.g., 5C, ChIA-PET, and repliSeq) essays. We also downloaded all genomic data publicly provided by TCGA [3], and modeled them in GDM; they include copy number variation (CNV), DNA somatic mutation (DNA-seq), DNA methylation, and gene expression (RNA-seq, and miRNA-seq) processed data. Furthermore, for each genomic data sample, we retrieved also the corresponding experiment and biological or (for TCGA data) clinical metadata, and modeled them according to GDM; the metadata free attribute-value format of GDM allowed associating each data sample with a variable number of metadata attributes, ranging from very few to hundreds or even thousands.

All downloaded and modeled datasets are described in Table 1; we aggregated all of them in a single integrative repository, together with known annotation data (also modeled in GDM) regarding transcription start sites (TSS) from SwitchGear Genomics (<http://switchgeargenomics.com/>), human and mouse protein-coding and

non-protein-coding genes from EMBL-EBI Ensembl (<http://www.ensembl.org/>) and NCBI Reference Sequence (RefSeq) (<http://www.ncbi.nlm.nih.gov/refseq/>) databases, RefSeq exons, Vista enhancers (<http://enhancer.lbl.gov/>), and CpG islands, all as provided by the UCSC database (<https://genome.ucsc.edu/cgi-bin/hgTables>). Since all such heterogeneous datasets are modeled with and mapped to the same simple and abstracted GDM, they are made interoperable and can be integratively processed easily.

Table 1

Modeled datasets of processed data from multiple public sources.

Consortium	Dataset	# of samples	File size (MB)
ENCODE	HG19_ENCODE_BED	1,933	32,201
	HG19_ENCODE_BROAD	1,970	23,552
	HG19_ENCODE_NARROW	1,999	7,168
	MM9_ENCODE_BROAD	441	2,355
	MM9_ENCODE_NARROW	277	1,162
ROADMAP	HG19_ROADMAP_EPIGENOMICS_BED	78	595
EPIGENOMICS	HG19_ROADMAP_EPIGENOMICS_BROAD	979	23,244
TCGA	HG19_TCGA_Cnv	2,623	117
	HG19_TCGA_Dnamethylation	1,384	29,696
	HG19_TCGA_DnaSeq	6,361	276
	HG19_TCGA_MirnaSeq_Isoform	9,227	3,379
	HG19_TCGA_MirnaSeq_Mirna	9,227	569
	HG19_TCGA_RnaSeq_Exon	2,544	31,744
	HG19_TCGA_RnaSeq_Gene	2,544	3,584
	HG19_TCGA_RnaSeq_Spljxn	2,544	30,720
	HG19_TCGA_RnaSeqV2_Exon	9,217	114,688
	HG19_TCGA_RnaSeqV2_Gene	9,217	20,480
	HG19_TCGA_RnaSeqV2_Isoform	9,217	49,152
	HG19_TCGA_RnaSeqV2_Spljxn	9,217	105,472
Grand total	19 datasets	80,999	480,154

3.1 The GenoMetric Query Language

We recently proposed the GenoMetric Query Language (GMQL) [19], which has the ability of computing distance-related queries over sets of linear intervals, ordered along a common coordinate system, and taking into account both individual interval attributes and set global characteristics. The GDM fully supports GMQL, since linear intervals can be genomic regions modeled by GDM through their genomic coordinates and features, and GDM sample metadata describe global characteristics of genomic region sets. In this context, a GMQL query (or program) is a sequence of GMQL operations with the following structure:

$$\langle \text{variable} \rangle = \text{operation}(\langle \text{parameters} \rangle) \langle \text{variables} \rangle$$

where each variable is a GDM dataset of samples of genomic regions and metadata. GMQL operations are either *unary* (with one input variable), or *binary* (with two input variables), and construct one result variable. Thus, all operations produce a result dataset usually consisting of several samples, whose identifiers are either inherited by the operands or generated by the operation.

GMQL operations include classic relational algebraic transformations (i.e., six unary operations: SELECT, EXTEND, PROJECT, MERGE, GROUP and ORDER, and two binary operations: UNION and DIFFERENCE), and domain-specific transformations which significantly extend the expressive power of classic relational algebra (i.e., COVER, dealing with replicate data samples of a same experiment; MAP, referring known or experimentally determined genomic features to user selected reference regions; and (distal) JOIN, selecting region pairs based upon distance properties). Each operation separately applies to sample metadata and regions. The region-based part of an operation computes the result regions; the metadata part of the operation computes the associated metadata, so as to trace the

provenance of each resulting sample. Identifiers preserve the many-to-many mapping of regions and metadata, as discussed in Section 2.3. Tracing provenance both of initial samples and of their processing through operations is a unique aspect of GMQL, which relevantly allows knowing why resulting regions were produced.

Compared with languages which are currently in use in the bioinformatics community, GMQL is *declarative* (it specifies the structure of the results, leaving result computation to each operation implementation) and *high-level* (one GMQL query typically substitutes for a long program which embeds calls to region manipulation libraries); its progressive computation of variables resembles other data management algebraic languages, such as *Pig Latin* [20]. GMQL has been implemented to be executed both on a single computer and in a cloud computing environment [19]; thus, it can well support knowledge discovery across thousands or even millions of samples, for what concerns both regions that satisfy biological conditions and their relationship to experimental, biological or clinical metadata.

For all these features, GMQL may inspire a change of paradigm in genomic data management, along a direction that was indicated long ago by Edgar F. Codd's seminal paper [21] for large data collections in general. In [19], we demonstrated the expressive power and flexibility of GMQL through examples of biological interest, which include finding binding sites in transcription regulatory regions, associating transcriptomics and epigenomics, and finding somatic mutations in exons. The combined use of GDM and GMQL shows its assets particularly when it is applied on heterogeneous datasets of multiple data types, each containing numerous samples with many genomic feature regions, as discussed in the following Section 3.2.

3.2 Comprehensive querying of heterogeneous genomic data through GMQL

Modeling heterogeneous datasets in GDM makes them interoperable and ready for common processing and comprehensive querying through GMQL. We demonstrate this valuable property provided by GDM, as well as the power and flexibility of GMQL, by illustrating some exemplar GMQL queries over heterogeneous data from multiple sources modeled in GDM, in a rich set of biological use cases. Reported performances refer to the execution of the exemplar queries on a server equipped with Intel[®] Xeon[®] Processor with CPU E5-2650 at 2.00 GHz, six cores, RAM of 128 GB and hard disk of 4x2 TB.

3.2.1 Example 1: Combining multiple replicate samples in different data formats

“For all antibody targets of the K562 chronic myelogenous leukemia cell line in ENCODE, merge broad and narrow peaks in ChIP-seq replicate samples and calculate the average enrichment (signal) for each obtained peak.”

```
HM_TF_rep_broad = SELECT(dataType == 'ChipSeq' AND view == 'Peaks' AND
    setType == 'exp' AND cell == 'K562') HG19_ENCODE_BROAD;
HM_TF_rep_narrow = SELECT(dataType == 'ChipSeq' AND view == 'Peaks' AND
    setType == 'exp' AND cell == 'K562') HG19_ENCODE_NARROW;
HM_TF_rep = UNION HM_TF_rep_broad HM_TF_rep_narrow;
HM_TF = COVER(1, ANY; GROUP BY cell, antibody_target; AVG(signal)) HM_TF_rep;
MATERIALIZE HM_TF;
```

Considering NGS experimental variability, replicates are usually performed and have to be taken into account in result evaluation, which can be done in multiple ways with

different stringency. Thanks to the use of GDM and GMQL, this example illustrates how it can be easily done even when replicate samples are in different formats. All ChIP-seq peak samples in BROADPEAK or NARROWPEAK format from the ENCODE data collection that regard the K562 chronic myelogenous leukemia cell line are selected and included in a single unifying dataset. Then, multiple replicate samples, in case existing for an antibody target of the K562 cell line, are combined in a single sample including the disjointed DNA regions where at least one peak in the replicates exists. The average enrichment of the peaks in the replicates that contribute to each obtained region is calculated and assigned to such region.

When this example query was executed over the HG19_ENCODE_BROAD and HG19_ENCODE_NARROW datasets described in Table 1, 130 BROADPEAK and 130 NARROWPEAK samples regarding 75 and 78 antibody targets, respectively, were selected, including a total of 4,566,008 and 4,426,212 peaks, respectively. After combining the replicates, 136 samples were obtained, containing a total of 5,121,711 regions regarding 136 antibody targets of the K562 cell line. Processing required 5.5 minutes.

3.2.2 Example 2: Combining ChIP-seq and DNase-seq data in different formats and sources

“Extract broad peaks of ChIP-seq transcription factor binding sites and histone modifications from ENCODE samples that intersect DNase-seq open chromatin regions from Roadmap Epigenomics in normal H1 embryonic stem cells.”

```
CHIPSEQ = SELECT(dataType == 'ChipSeq' AND view == 'Peaks' AND setType == 'exp'  
AND cell == 'H1-hESC') HG19_ENCODE_BROAD;
```

```

DNASESEQ = SELECT(assay == 'DNase.hotspot.broad' AND
    Standardized_Epigenome_name == 'H1 Cells')
    HG19_ROADMAP_EPIGENOMICS_BED;
DNASESEQ1 = COVER(1, ANY) DNASESEQ;
CHIPSEQ_IN_DNASESEQ = JOIN(distance < 0, project_right_distinct) DNASESEQ1
    CHIPSEQ;
MATERIALIZE CHIPSEQ_IN_DNASESEQ;

```

Combining data available, but in different formats and sources, this example shows how to improve the quality of ChiP-seq called peaks by filtering out those peaks that are not in open chromatin regions, where only they can be present biologically. For the same tissue, available ChIP-seq broad peaks from the ENCODE data collection, and DNase-seq open chromatin regions from the Roadmap Epigenomics Project, are selected. Multiple DNase-seq replicate samples in case existing are first combined in a single sample including all identified open chromatin regions, which are then joined with ChIP-seq peaks; only the peaks that at least partially overlap any of these regions are finally extracted. The join is performed for each of the selected ChIP-seq samples individually, so that each resulting sample is a selected ENCODE ChIP-seq sample, but including only the peaks that intersect open chromatin regions.

By executing this GMQL example query, whose HG19_ENCODE_BROAD and HG19_ROADMAP_EPIGENOMICS_BED input datasets are described in Table 1, 90 ChIP-seq samples regarding 54 antibody targets and 1 DNase-seq sample were initially selected, including a total of 3,071,136 peaks and 412,042 regions, respectively. ChIP-seq called peaks finally obtained were 2,097,289 in total, regarding 54 different antibody targets. Processing was performed in 4.5 minutes.

3.2.3 Example 3: Combining heterogeneous omics data of patients

“In TCGA data of breast cancer patients, find the DNA somatic mutations within the first 2000 bp¹⁰ outside of the genes that are expressed and methylated in at least one of these patients, and extract the top five patients with the highest number of such mutations and their somatic mutations.”

```
EXPRESSED_GENE = SELECT(dataType == 'rnaseqv2' AND tumor_tag == 'brca')
    HG19_TCGA_RnaSeqV2_Gene;
METHYLATION = SELECT(dataType == 'dnamethylation' AND tumor_tag == 'brca')
    HG19_TCGA_Dnamethylation;
MUTATION = SELECT(data_type == 'dnaseq' AND tumor_tag == 'brca')
    HG19_TCGA_DnaSeq;
GENE_METHYL = JOIN(left->bcr_sample_barcode == right->bcr_sample_barcode,
    distance < 0, project_left_distinct) EXPRESSED_GENE METHYLATION;
GENE_METHYL1 = COVER(1, ANY) GENE_METHYL;
MUTATION_GENE = JOIN(DISTANCE < 2000 AND DISTANCE > 0, left) MUTATION
    GENE_METHYL1;
MUTATION_GENE_count = AGGREGATE(mutation_count AS COUNT)
    MUTATION_GENE;
MUTATION_GENE_top = ORDER(DESC mutation_count; TOP 5)
    MUTATION_GENE_count;
MATERIALIZE MUTATION_GENE_top;
```

¹⁰ Distances along the DNA are measured in *base pairs (bp)*, i.e., number of nucleotides (or bases) present between two points of the DNA.

Comprehensively considering genomic, epigenomic and transcriptomic data of cancer patients can provide a better view of the patients' complex biomolecular system, which may lead to interesting findings. Leveraging on GDM and GMQL, this example presents how to do it focusing on expressed genes, DNA methylations (which generally repress gene expression) and the DNA somatic mutations close to methylated expressed genes. From the TCGA data collection, first all expressed gene, DNA methylation and DNA somatic mutation data of patients affected by breast cancer are selected. Then, by joining these heterogeneous data, the expressed genes with at least a DNA methylation are identified, and the DNA somatic mutations close to these genes are extracted.

The execution of this GMQL example query, over the HG19_TCGA_RnaSeqV2_Gene, HG19_TCGA_Dnamethylation and HG19_TCGA_DnaSeq datasets described in Table 1, initially selected 1,218 samples of gene expression data, 11 of DNA methylation data, and 993 of DNA somatic mutations of TCGA breast cancer patients, containing a total of 24,986,052 expressed genes, 4,024,460 methylation sites, and 90,490 DNA mutations, respectively. The combination (through a GMQL join operation) of each patient's gene expression and DNA methylation data, modelled with GDM, identified 10 breast cancer patients presenting methylated expressed genes, with an average of 11,481 of such genes for each identified patient; these patients presented an average *age at diagnosis* of 57.80, an average *percent stromal cells* of 13.90 %, an average *percent tumor nuclei* of 69.0 %, and an average *tumor necrosis percent* of 0.11 %, versus the average of 58.0, 22.42 %, 72.34 %, and 6.62 %, respectively, of all the initially considered patients with expressed gene data, based on the patients' clinical data reported in the available

sample metadata managed by GDM. Then, the query takes into account all and only the expressed genes methylated in at least one of the considered patients and, for each TCGA breast cancer patient with DNA somatic mutation data, extracts the mutations occurring within the first 2,000 bp outside of these genes (801 patients were found with such mutations). Finally, these mutations in each patient are count (their average number per patient was 5.5) and the top 5 patients with the highest number of such mutations are selected.

Thanks to the unique and innovative seamless management provided by GDM, the MUTATION_GENE_top result dataset includes both genomic somatic mutations and clinical metadata of the finally selected patients. The former ones indicate interesting mutations that could be associated with breast cancer (which can be further inspected using viewers, e.g., genome browsers [22]); the latter ones allow tracking the provenance of resulting samples and ease the biomedical interpretation of the results. This association between processed genomic data and their biological/clinical metadata is not supported by any other system currently available, and represents one of the new relevant aspects of GDM and GMQL. Table 2 reports an excerpt of the six most relevant metadata attributes and of their values associated with the five finally selected patients: the patient's mutation count and order within the patients with the highest number of mutations, the age of the patient at her breast cancer diagnosis, and the percentage of stromal cells and tumor nuclei in the evaluated patient's histological sample.

Table 2

Metadata excerpt of the five patients finally selected in Example 3.

patient id	mutation count	order	age at diagnosis	percent stromal cells	percent tumor nuclei	tumor necrosis percent
a046	210	1	68	10	80	0
a23h	199	2	90	25	85	0
a0a6	150	3	64	24	75	20
a18g	89	4	81	5	90	0
a0t5	75	5	39	25	75	0

4. Discussion and conclusions

New biotechnologies are increasingly providing high amounts of reliable data describing a growing number of different biomolecular aspects characterizing the cellular status and activity of an individual. Comprehensive processing of these valuable data can provide biological system views which are paving the way to personalized and precision medicine. Yet, the amount and high heterogeneity of these data, and of the formats in which they are produced, hamper their effective use. Furthermore, their complexity is manifesting itself also in the heterogeneity and large number of underlying samples, conditions, etc. that these data represent. GDM provides interoperability across tens of processed data formats, while GMQL supports their high-level query processing. Hundreds of datasets and thousands of samples of heterogeneous processed data, as those provided by large consortia such as ENCODE, Roadmap Epigenomics or TCGA, can be made interoperable and comprehensively evaluated thanks to GDM and GMQL.

The far majority of genomic data are available in tab-delimited ASCII text formats or in their serialized binary version. The defined GDM, XML description of the data

and associated software loaders support seamless data interoperability and integration. In particular, the GDM provides a unifying modelling and mapping of the many and heterogeneous genomic data and formats; the XML description of the data allows changing data attribute names to uniform them in different GDM datasets when they represent the same data, while the associated software loaders allow to add necessary data attributes and to convert data attribute content when required (e.g., region coordinates from 1-based to 0-based system if needed, or feature attributes to make them comparable to equivalent attributes with the same name in other GDM datasets).

Data mapping and conversion/normalization at data usage time allows not interfering with the data stored in their original format, thus preserving their availability and usability for the plethora of tools currently used by biologists and bioinformaticians; yet, it can slow the usage/reading time of such data which are usually big. When an integrative repository is built for the management and querying of these data through GMQL, as we did, data conversion/normalization can be performed while integrating the data in the repository, through classic extraction, transformation and loading (ETL) operations typically performed in data warehouse construction [23].

As shown with the examples in Section 2.4, GDM provides interoperability across tens of processed data formats; thousands or even millions of processed experimental samples, which are becoming available [24], can be modeled and managed through the GDM. We consider the GDM a paradigm shift, because a single model describes, through simple concepts, all types of (epi)genomic feature data (binding peaks, histone modifications, methylations, expressions, mutations, DNA sequences, loops, break points, etc.) and allows the seamless integration of heterogeneous genomic, epigenomic, transcriptomic and gene activity regulation data.

At <http://www.bioinformatics.deib.polimi.it/GMQL/queries/> the power of GDM and GMQL can be tested through a set of predefined GMQL queries on ENCODE and Roadmap Epigenomics data modeled with GDM; query results can be automatically shown on the Integrated Genome Browser [22], and can be downloaded to be post-processed with data analysis tools (e.g., supporting data mining or machine learning algorithms) and visualized, e.g., through heat maps.

Both GDM and GMQL are part of our genomic computing new holistic approach to genomic big data modeling and querying (http://www.bioinformatics.deib.polimi.it/genomic_computing/), which was recently awarded an ERC Advanced Grant (*Data-Driven Genomic Computing - GeCo*, 2016-2021). Our current work is along three dimensions: i) From a technological point of view, we are completing our third implementation of GMQL version 2.0, each supported by a different execution engine (Apache Flink, Apache Spark, and SciDB¹¹); the Apache Spark implementation of the system was installed at CINECA, an interuniversity consortium within Italy, and can be freely used at <http://www.bioinformatics.deib.polimi.it/GMQL/interfaces/>. ii) From the data integration point of view, we are currently working on improving metadata interoperability across datasets provided by international consortia, starting with ENCODE [25] and TCGA [26]. In particular, for what concerns ENCODE, we developed a method for matching metadata entries (both attribute names and values) to the well-established Unified Medical Language System (UMLS) ontologies [25]; we plan to generalize the method and apply it across various data collections. iii) For what concerns biological research, we are currently integrating data about the

¹¹ <http://www.scidb.org/>

tridimensional structure of the genome [27], mapped to GDM, and investigating, by using public TCGA datasets modeled and made interoperable through GDM, the relationships between the disruption of the boundaries of genomic topologically associating domains (TADs) and various types of cancer, along the direction of [28].

Acknowledgements

This work is part of and supported by the “Data-Driven Genomic Computing (GenData 2020)” PRIN project (2013-2015), funded by the Italian Ministry of the University and Research (MIUR). We thank Heiko Muller and Yuriy Vaskin to suggest the biological use case and GMQL query regarding the intersection of genomic regions from ENCODE ChIP-seq and Roadmap Epigenomics DNase-seq assays.

References

- [1] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489:7414 (2012) 57-74.
- [2] C.E. Romanoski, C.K. Glass, H.G. Stunnenberg, L. Wilson, G. Almouzni, Epigenomics: Roadmap for regulation. *Nature* 518: 7539 (2015) 314-316.
- [3] Cancer Genome Atlas Research Network, J.N. Weinstein, E.A. Collisson, G.B. Mills, K.R. Shaw, B.A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J.M. Stuart. The Cancer Genome Atlas pan-cancer analysis project. *Nat. Genet.* 45:10 (2013) 1113-1120.
- [4] 1000 Genomes Project Consortium, G.R. Abecasis, D. Altshuler, A. Auton, L.D. Brooks, R.M. Durbin, R.A. Gibbs, M.E. Hurles, G.A. McVean. A map of human genome variation from population-scale sequencing. *Nature* 467:7319 (2010) 1061-1073.

- [5] R. Gabe. A hitchhiker's guide to Next Generation Sequencing - Part 2. <http://blog.goldenhelix.com/grudy/a-hitchhikers-guide-to-next-generation-sequencing-part-2/> (2010). Last accessed 30 April 2016.
- [6] Accelerating bioinformatics research with new software for big data knowledge. White paper from: <http://www.paradigm4.com/>. Last accessed 30 April 2016.
- [7] T. Hey, S. Tansley, K. Tolle. Jim Gray on eScience: a transformed scientific method. In T. Hey, S. Tansley, K. Tolle (Eds.). The fourth paradigm. Data-intensive scientific discovery. Microsoft Research. Redmond, WA. 2009, pp. XVII-XXXI.
- [8] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25:16 (2009) 2078-2079.
- [9] H. Li, R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 25:14 (2009) 1754-1760.
- [10] B. Langmead, C. Trapnell, M. Pop, S.L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10:3 (2009) R25.
- [11] P.J. Cock, C.J. Fields, N. Goto, M.L. Heuer, P.M. Rice. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res.* 38:6 (2010) 1767-1771.
- [12] P. Danecek, A. Auton, G. Abecasis, C.A. Albers, E. Banks, M.A. DePristo, R.E. Handsaker, G. Lunter, G.T. Marth, S.T. Sherry, G. McVean, R. Durbin,

- 1000 Genomes Project Analysis Group. The variant call format and VCFtools. *Bioinformatics* 27:15 (2011) 2156-2158.
- [13] Y. Zhang, T. Liu, C.A. Meyer, J. Eeckhoute, D.S. Johnson, B.E. Bernstein, C. Nusbaum, R.M. Myers, M. Brown, W. Li, X.S. Liu. Model-based analysis of ChIP-Seq (MACS). *Genome Biol.* 9:9 (2008) R137.
- [14] N.U. Rashid, P.G. Giresi, J.G. Ibrahim, W. Sun, J.D. Lieb. ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions. *Genome Biol.* 12:7 (2011) R67.
- [15] P. Cockerill. Structure and function of active chromatin and DNase I hypersensitive sites. *FEBS J.* 278:13 (2011) 2182-2210.
- [16] P. Park. ChIP-seq: advantages and challenges of a maturing technology. *Nat. Rev. Genet.* 10:10 (2009) 669-680.
- [17] A.S. Zweig, D. Karolchik, R.M. Kuhn, D. Haussler, W.J. Kent. UCSC genome browser tutorial. *Genomics* 92:2 (2008) 75-84.
- [18] Genome Informatics Group of the University of California at Santa Cruz (UCSC). UCSC database Table Browser. <https://genome.ucsc.edu/cgi-bin/hgTables> Last accessed 30 April 2016.
- [19] M. Masseroli, P. Pinoli, F. Venco, A. Kaitoua, V. Jalili, F. Palluzzi, H. Muller, S. Ceri. GenoMetric Query Language: A novel approach to large-scale genomic data management. *Bioinformatics* 31:12 (2015) 1881-1888.
- [20] C. Olston, B. Reed, U. Srivastava, R. Kumar, A. Tomkins. Pig Latin: A not-so-foreign language for data processing. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (2008) 1099-1110.

- [21] E.F. Codd. A relational model of data for large shared data banks. *Comm. ACM* 13:6 (1970) 377-387.
- [22] J.W. Nicol, G.A. Helt, S.G. Blanchard Jr., A. Raja, A.E. Loraine. The Integrated Genome Browser: free software for distribution and exploration of genome-scale datasets. *Bioinformatics* 25:20 (2009) 2730-2731.
- [23] R. Kimball, J. Caserta. *The Data Warehouse ETL toolkit: Practical techniques for extracting, cleaning, conforming, and delivering data*, first ed., Wiley Publishing, Inc., Indianapolis, IN, 2004.
- [24] Z.D. Stephens, S.Y. Lee, F. Faghri, R.H. Campbell, C. Zhai, M.J. Efron, R. Iyer, M.C. Schatz, S. Sinha, G.E. Robinson. Big Data: Astronomical or Genomical? *PLoS Biol.* 13:7 (2015) e1002195.
- [25] J.D. Fernandez, M. Lenzerini, M. Masseroli, F. Venco, S. Ceri. Ontology-based search of genomic metadata. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 13:2 (2016) 233-247.
- [26] F. Cumbo, G. Fiscon, M. Masseroli, S. Ceri, E. Weitschek. TCGA2BED: extracting, extending, integrating, and querying The Cancer Genome Atlas. *BMC Bioinformatics* (submitted).
- [27] J.R. Dixon, S. Selvaraj, F. Yue, A. Kim, Y. Li, Y. Shen, M. Hu, J.S. Liu, B. Ren. Topological Domains in mammalian genomes identified by analysis of chromatin interactions. *Nature* 485:7398 (2012) 376-380.
- [28] W.A. Flavahan, Y. Drier, B.B. Liau, S.M. Gillespie, A.S. Venteicher, A.O. Stemmer-Rachamimov, M.L. Suvà, B.E. Bernstein. Insulator dysfunction and oncogene activation in IDH mutant gliomas. *Nature* 529:7584 (2016):110-114.