Conceptual Modeling for Genomics: Building an Integrated Repository of Open Data

Anna Bernasconi(✉), Stefano Ceri, Alessandro Campi, and Marco Masseroli

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Milano, Italy
{anna.bernasconi,stefano.ceri,alessandro.campi,marco.masseroli}@polimi.it

Abstract. Many repositories of open data for genomics, collected by world-wide consortia, are important enablers of biological research; moreover, all experimental datasets leading to publications in genomics must be deposited to public repositories and made available to the research community. These datasets are typically used by biologists for validating or enriching their experiments; their content is documented by metadata. However, emphasis on data sharing is not matched by accuracy in data documentation; metadata are not standardized across the sources and often unstructured and incomplete.

In this paper, we propose a conceptual model of genomic metadata, whose purpose is to query the underlying data sources for locating relevant experimental datasets. First, we analyze the most typical metadata attributes of genomic sources and define their semantic properties. Then, we use a top-down method for building a global-as-view integrated schema, by abstracting the most important conceptual properties of genomic sources. Finally, we describe the validation of the conceptual model by mapping it to three well-known data sources: TCGA, ENCODE, and Gene Expression Omnibus.

Keywords: Conceptual model · Data integration · Genomics · Next Generation Sequencing · Open data

1 Introduction

Thanks to Next Generation Sequencing, a recent technological revolution for reading the DNA, a huge number of genomic datasets have become available. Sequencing machines perform the *primary data analysis* and produce raw datasets (a single human genome requires about 200GB). Computationally expensive pipelines, collectively regarded as *secondary data analysis* [30], are then applied to raw data for extracting signals from the genome (such as: mutations, expression levels, peaks of binding enrichment, chromatin states, etc.), thereby producing **processed genomic data**, which are much smaller in size.

Processed datasets are collected by worldwide consortia, such as TCGA (The Cancer Genome Atlas) [36], ENCODE (the Encyclopedia of DNA Elements) [28], Roadmap Epigenomics [19], and 1000 Genomes [27]; moreover, it is customary for authors of biological articles to publish their processed datasets on repositories such as GEO (the Gene Expression Omnibus) [4]. These datasets constitute a wealth of information, as they are open and can be used for secondary research. Processed datasets are used in *tertiary data analysis* for giving a global sense to heterogeneous genomic and epigenomic signals, thereby answering complex biological queries. Several systems are dedicated to tertiary data analysis, including FireCloud[1], SciDB-Paradigm4 [26], and BLUEPRINT [1]. In the context of the GeCo Project[2], we developed GMQL [23,17], a high-level query language for genomics; we also proposed GDM [24], a unifying model for processed data formats.

While a lot of efforts are made for the production of genomic datasets, much less emphasis is given to the structured description of their content. Such descriptions, collectively regarded as *metadata*, are fundamental for understanding how each biological sample was processed, to which biological or clinical condition it is associated, which technological process has been used for its production, and so on. There is no standard for metadata, thus each source/consortium enforces some rules autonomously; a conceptual design for metadata is either missing or, when present, overly complex and useless[3]. In summary, in spite of a growing interest on tertiary data analysis and of the availability of many valuable data sources, genomic metadata are lacking a conceptual model for understanding which sources and datasets are most suitable for answering a genomic question.

One of the far-reaching goals of the GeCo project is the development of an integrated repository of open processed data, supporting both structured and search queries; the GMQL prototype[4] already integrates data from three repositories (TCGA, ENCODE, and Roadmap Epigenomics) and structured methods for periodically loading and keeping updated their contents. To overcome the lack of standards, metadata are stored in GMQL as generic attribute-value pairs; with such format, metadata are used for the initial selection of relevant datasets. However, we are aware of the fact that attribute-value pairs are just providing a viable solution, but do not carry enough semantics.

In this paper, we present the **Genomic Conceptual Model (GCM)**, a conceptual model for describing metadata of genomic data sources. GCM is centered on the notion of the *experiment item*, typically a file containing genomic regions and their properties, which is analyzed from three points of view:

- The *technology* used in the experiment, including information about item containers and their formats.

[1] https://software.broadinstitute.org/firecloud/

[2] Data-Driven Genomic Computing, http://www.bioinformatics.deib.polimi.it/geco/, ERC Advanced Grant, 2016-2021.

[3] At https://www.encodeproject.org/profiles/graph.svg see the conceptual model of ENCODE, an ER schema with tens of entities and hundreds of relationships, which is neither readable nor supported by metadata for most concepts.

[4] http://www.bioinformatics.deib.polimi.it/GMQL/interfaces/

- The *biological* process observed in the experiment, in particular the sample being sequenced (derived from a tissue or a cell culture) and its preparation, including its donor.
- The *management* of the experiment, describing the organizations/projects which are behind the production of each experiment.

The conceptual schema is constructed top-down, based on a systematic analysis of metadata attributes and of their properties in many genomic sources, and then verified bottom-up, on TCGA, ENCODE, and GEO; we show that ER schemas describing these sources can be constructed as subsets of GCM. Arbitrary queries on GCM can be propagated to sources, using the *global-as-view* approach [20]. We also show that GCM provides the skeleton to a simple query interface, similar to the one provided by DeepBlue [2]. Driven by GCM, we will add many more data sources to our integrated repository of open data for genomics.

2 Design of GCM

2.1 Analysis of Metadata Attributes

Most data sources provide interfaces for metadata extraction; these are based on simple query templates or application programming interfaces (APIs), and enable the selection of experimental data. Some sources also provide tabular descriptions of the metadata that can be more systematically queried, or enable the extraction of matching metadata in semistructured format (XML or JSON files).

Taxonomy of Metadata Attributes. As a first step in developing GCM, we defined a taxonomy of the main properties of metadata attributes; we then systematically applied the taxonomy to each considered source, so as to better characterize its content. According to our taxonomy, attributes are:

- **contextual (C)** when they are present (or absent) only within specific contexts, typically because another attribute takes a specific value. In such cases, there is an *existence dependency* between the two attributes.
- **dependent (D)** when the domain of their possible values is restricted, typically because another attribute takes a specific value. In such cases, there is a *value dependency* between the two attributes.
- **restricted (R)** when their value must be chosen from a controlled vocabulary.
- **single-valued (S)** when they assume at most one value for each specific experiment.
- **mandatory (M)** when they must have a value, either for all experiments or within a specific context.

The resulting taxonomy is shown in Table 1; it includes orthogonal features, and we targeted both completeness and minimality. By default (and in most

Table 1. Taxonomy of features for metadata attributes

Level	Symbol	Feature	Default
Source	C	Contextual	Non-contextual
	D	Dependent	Independent
	R	Restricted	Free
	S	Single-valued	Multi-valued
	M	Mandatory	Optional
Integrated	H	Human Curated	Extracted
Repository	O	Ontological	Ordinary

cases), attributes do not have any of the above properties. Very few attributes are mandatory and unfortunately sources do not always agree on them; in many cases they are named and typed somehow differently.

We use these five categories to describe the attributes that are included in the conceptual model, as explained in the next section; we label the attributes with a feature vector, e.g. $\mathit{Type}^{[RSM]}$ denotes *Type* as an attribute which is mandatory, restricted and single-valued, while $\mathit{Pipeline}^{[D(\mathit{Technique})S]}$ denotes *Pipeline* as a single-valued attribute with a value dependency from the attribute *Technique*.

Source Analysis. We examined several sources; among them TCGA and ENCODE provide the most comprehensive collection of metadata attributes.

- **TCGA** reports many experiment pipeline-specific metadata attributes; out of them we selected 22 attributes, common to all pipelines, which are the most interesting from a biological point of view (Table 2).
- **ENCODE** includes both a succinct and an expanded list of metadata attributes; while the expanded list has over 2000 attributes, the succinct list has 49 attributes for experiments, 44 attributes for biosamples, and 28 attributes for file descriptions.

Other Properties. We next define properties that we could not observe in the sources, but will be used for characterizing the metadata attributes of our integrated repository (they are also included in Table 1). Accordingly, attributes are:

- **human curated (H)** when their value is provided by a curator of the repository (and not extracted from the underlying data source).
- **ontological (O)** when an interface supports similarity-based matches based upon semantic properties, e.g. through the connection to external ontologies.

Rules. Rules may be used for expressing existence and value dependencies.

- The **existence dependency** *Technique* = "Chip-seq" → M(*Target*) indicates that *Target* is a mandatory attribute if *Technique* takes the value "Chip-seq", while *Technique* ≠ "Chip-seq" → NULL(*Target*) indicates that *Technique* is not specified otherwise.
- The following **value dependency** connects the *DataType* and *Format* attributes: *DataType* = "raw data" → *Format* = "fastq".

Table 2. TCGA metadata attributes analysis

C	D	R	S	M	Dependency	Attribute
			×	×		clinical.demographic.id
			×			clinical.demographic.year_of_birth
		×	×	×		clinical.demographic.gender
		×	×	×		clinical.demographic.ethnicity
		×	×	×		clinical.demographic.race
			×	×		biospecimen.sample.id
		×	×	×		biospecimen.sample.sample_type
	×	×	×		sample_type	biospecimen.sample.tissue_type
			×	×		generated_data_files.data_file.⟨type⟩.id
		×	×	×		generated_data_files.data_file.⟨type⟩.data_type
	×	×	×	×	data_type	generated_data_files.data_file.⟨type⟩.data_format
			×	×		generated_data_files.data_file.⟨type⟩.file_size
	×	×	×	×	data_type	generated_data_files.data_file.⟨type⟩.experimental_strategy
×		×	×		data_type	generated_data_files.data_file.⟨type⟩.platform
	×	×	×	×	data_type	analysis.⟨workflow⟩.workflow_type
			×	×		analysis.⟨workflow⟩.workflow_link
			×	×		case.case.id
			×			case.case.primary_site
	×		×		primary_site	case.case.disease_type
			×	×		administrative.program.name
			×	×		administrative.project.name
			×			administrative.tissue_source_site.name

In the next section we show examples of both existence and value dependencies, that complement the conceptual model specification; when the dependencies are specified for attributes belonging to different entities, they hold for all the instance pairs connected with an arbitrary join path connecting the two entities (this is not ambiguous because the conceptual model is acyclic).

2.2 Genomic Conceptual Model

We next designed the Genomic Conceptual Model top-down, inclusive of the most relevant metadata attributes as scouted from the various sources, building the entity-relationship schema represented in Fig. 1. The schema includes the principal concepts; other source-specific concepts can be made available in semi-structured form aside from this schema (e.g. all clinical diagnosis conditions available for the donor in TCGA). The model is centered on the Item entity, which represents an elementary experimental unit. Three sub-schemata (or views) depart from the central entity, recalling a classic star-schema organization that is typical of data warehouses; they respectively describe biological, technological, and management aspects.

Central Entity. We next describe the attributes of the Item entity and associate each of them with their feature vector. The $\mathit{SourceId}^{[SM]}$ and $\mathit{DataType}^{[RSM]}$ respectively denote the item identifier within the source and the item's data type, and must always be included; *DataType* denotes the specific content of the Item, e.g. "peak". $\mathit{Format}^{[D(\mathit{DataType})RSM]}$ denotes the Item data file format (e.g. ["fastq", "bam", "wiggle", "bed", "tsv", "vcf", "maf", "xml"]) and depends on *DataType* (e.g. "bed" format is compatible with "peak" and not compatible with "read"). Other attributes are: $\mathit{Size}^{[SM]}$, $\mathit{SourceUrl}^{[M]}$, $\mathit{LocalUri}^{[C(\mathit{Format})SM]}$, and $\mathit{Pipeline}^{[D(\mathit{Technique})S]}$.

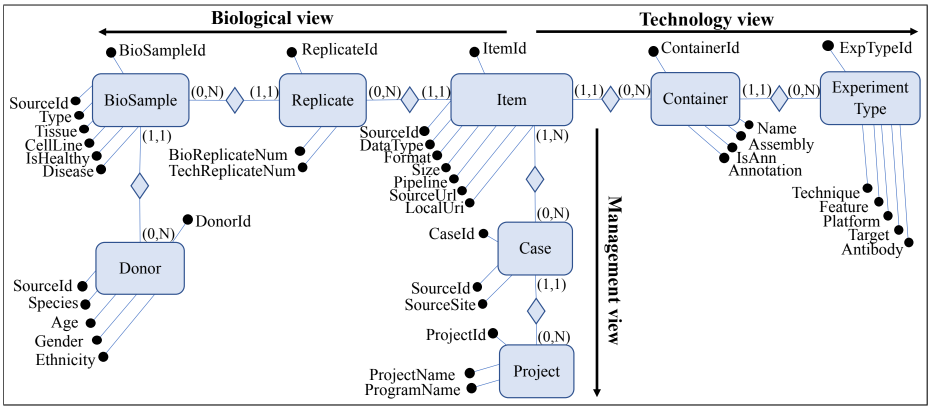

Fig. 1. Genomic Conceptual Model

The use of the last three attributes requires some discussion. Recall that we intend to build an integrated repository that contains only processed data, while in many cases the sources include also the raw data. In our metadata repository we include items relative to both raw and processed data with a reference to the related file in the original source within the *SourceUrl* attribute, that can be multi-valued in case the same data file is derived from different sources. In addition, items relative to processed files also exhibit an attribute *LocalUri* (see rule 1 in Listing 1, at the end of this section) indicating their physical location in our data repository. *Pipeline* is a descriptor of the specific parameters adopted in the pipeline used for producing the processed data. The descriptor is interpreted in the general context of the *Technique* used for producing several items of the same type and format; hence, the feature vector notation for *Pipeline*. Providing parameters and references to the raw data is relevant in the case of processed data, as sometimes biologists resort to original raw data for reprocessing; however, in the data sources such attributes may be missing or hidden within textual attributes.

Biological View. This view consists of a chain of entities: Item-Replicate-BioSample-Donor describing the biological process leading to the production of the Item. All relationships are many-to-one, hence an Item is associated with a given Replicate, each associated with a given BioSample, each associated with a Donor.

Donor represents the individual of a specific organism from which the biological material is derived. It has attributes *SourceId*$^{[S]}$ (donor identifier relative to a source) and *Species*$^{[RSM]}$; *Age*$^{[S]}$, *Gender*$^{[RS]}$, and *Ethnicity*$^{[RS]}$ are other optional attributes of interest.

BioSample describes the material sample taken from a biological entity and used for the experiment. Its *SourceId*$^{[M]}$ is an identifier of the bio-sample within a source, mandatory but also multi-valued (when the same sample is linked to different sources). *Type*$^{[RSM]}$ is restricted to the values ["cell line","tissue"]. Based on the value of this attribute, either *Tissue*$^{[CSMO]}$ or *CellLine*$^{[CSMO]}$ becomes mandatory, but not both of them; this dependency is expressed by rules 2-3. *IsHealthy*$^{[RS]}$ is Boolean and *Disease*$^{[C(IsHealthy)D(Tissue)O]}$ contextually depends on *IsHeathy*, as expressed by rule 4, and can be multi-valued; moreover, its values depend on *Tissue* because given diseases can only be related to given tissues. We marked *Tissue*, *CellLine* and *Disease* as ontological[5], as we intend to extend the values of these attributes with their synonyms and generalizations/specializations, so as to ease their search; for example, the *Tissue* "blood vessel" will match the terms "vessel", "arteries" and "veins". Preliminary work for giving an extended ontological interpretation to ENCODE metadata is reported in [11].

Replicate is used when multiple material samples are generated from the same BioSample, giving rise to items that are replica for the same experiment. This entity is relevant in some epigenomic data sources (such as ENCODE), that differentiate between technical and biological replication; such distinction is not present in most of the other sources.

Technology View. This view consists of a chain of entities: Item-Container-ExperimentType describing the used technology leading to the production of the Item. Through this chain, an Item is associated by means of (1:N) relationships to a given Container of a given ExperimentType.

Container is used to describe common properties of homogeneous items - sharing the same data structure and produced by the same experiment type. Its attributes include *Name*$^{[SM]}$ and *Assembly*$^{[C(DataType)D(Species)RSM]}$; *Assembly* is only present for items of particular types (see rule 5) and is restricted to a smaller vocabulary according to the *Species* (e.g. see rule 16 and 17). The Boolean attribute *IsAnn*$^{[RSM]}$ is used for distinguishing experimental items from known annotations (i.e., regarding known genomic regions): when true, *Annotation*$^{[C(IsAnn)RSM]}$ exists (see rules 6-7); annotations have a restricted vocabulary, including: ["Gene", "Exon", "TSS", "Promoter", "Enhancer", "Cpg-Island"].

ExperimentType refers to the specific methods used for producing each item. It includes the mandatory attribute *Technique*$^{[RSM]}$ (e.g., ["Chip-seq", "Dnase-seq", "RRBS", ...]). *Feature*$^{[D(Technique)RSMH]}$ is a mandatory manually curated attribute that we add to denote the specific feature described by the experiment (e.g., "Copy Number Variation", "Histone Modification", "Transcription Factor"). The value of *Platform*$^{[C(DataType)RSM]}$ illustrates the NGS platform used for sequencing and depends on the *DataType* of the item (see rule 8). When the *Technique* is "Chip-seq", the two attributes *Target*$^{[C(Technique)RSM]}$ and *Antibody*$^{[C(Technique)D(Target)RSM]}$ are present (see rules 9-12). The *Target* value is usually aligned to the vocabulary of UniProtKB[6]. The *Antibody* value depends on the *Target* since it is specific against that antigen.

[5] We will use the BRENDA Tissue and Enzyme Source Ontology [32] for tissues, the Cell Line Ontology [31] for cell lines, and the Human Disease Ontology [33] for human diseases.

[6] http://www.uniprot.org/uniprot/

Management View. This view consists of a chain of entities: Item-Case-Project describing the organizational process for the production of each item and the way in which items are grouped together to form a case.

Case represents a set of items that are gathered together, because they participate to a same research objective.

Project represents the project or program, occurred at a given institution (e.g., individual laboratory or consortium) that was responsible of the production of the item. *ProjectName*$^{[S]}$ and *ProgramName*$^{[S]}$ may be present, but none of them is mandatory.

Dependencies Rules 1-12 of Listing 1 exhaustively describe the existence dependencies of the global schema. Rules 13-17 show some examples of value dependencies. Note that an attribute can be contextual but not mandatory (such as *Disease*, rule 4), contextual and mandatory (such as *Target*, rules 9-10), and also mandatory but not contextual (such as *Technique*). Note also that, when an attribute is marked as mandatory and the related information is missing from the source, then either human curation or rule-based management are needed.

Item.*Format*=“bed” → M(Item.*LocalUri*) 1

BioSample.*Type*=“tissue” → M(BioSample.*Tissue*) 2

BioSample.*Type*=“cell_line” → M(BioSample.*CellLine*) 3

BioSample.*IsHealthy* → NULL(BioSample.*Disease*) 4

Item.*DataType* in [“aligned read”,“peak”,“signal”] → M(Container.*Assembly*) 5

Container.*IsAnn* → M(Container.*Annotation*) 6

NOT(Container.*IsAnn*) → NULL(Container.*Annotation*) 7

Item.*DataType*=“raw data” → M(ExperimentType.*Platform*) 8

ExperimentType.*Technique*=“Chip-seq” → M(ExperimentType.*Target*) 9

ExperimentType.*Technique*≠“Chip-seq” → NULL(ExperimentType.*Target*) 10

ExperimentType.*Technique*=“Chip-seq” → M(ExperimentType.*Antibody*) 11

ExperimentType.*Technique*≠“Chip-seq” → NULL(ExperimentType.*Antibody*) 12

Item.*DataType*=“raw data” → Item.*Format*=“fastq” 13

BioSample.*Tissue*=“liver” → BioSample.*Disease* ∈ [“viral hepatitis”,“liver lymphoma”,...] 14

BioSample.*Tissue*=“liver” → BioSample.*Disease* ∉ [“acute leukemia”,“pilorus cancer”,...] 15

Donor.*Species*=“Homo sapiens” → Container.*Assembly* ∈ [“GRCh38”, “hg19”, “hs37d5”] 16

Donor.*Species*=“Mus musculus” → Container.*Assembly* ∈ [“mm9”, “mm10”, “GRCm38”] 17

Listing 1. Examples of existence and value dependencies

2.3 Source-Specific Views of GCM

We verify that the global-as-view approach really captures the three data sources considered, by showing them as *views of GCM* in Fig. 2; we use the following notation:

- We place the attributes of each source in the same position as in GCM, but we use for them the name that we found in the documentation of each source; missing attributes correspond to white circles.
- We cluster the conceptual entities corresponding to a single concept in the original source by encircling them within grey shapes. The entity names corresponding to the original source are reported with a bold bigger font on the clustered shape (e.g. Series in GEO) or directly on the new entity (e.g., Case in TCGA) when this corresponds to the name given in our GCM.

- We indicate specific relationship cardinalities where GCM differs from the source, using a bold font (e.g., see (1,1) from Item to Case in ENCODE).
- We enclose fixed human curated values in inverted commas and use the functions notation *tr*, *comb*, and *curated* to describe a transformation of a source field, a combination of multiple source fields, and curated fields, respectively.

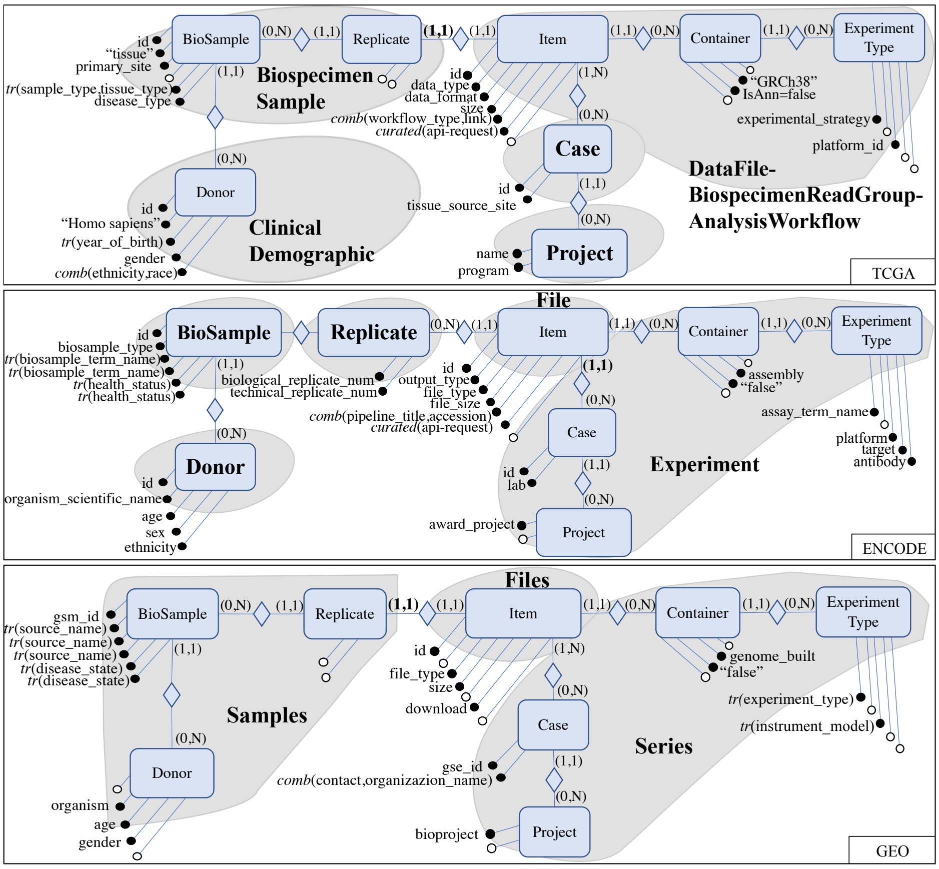

Fig. 2. Source-specific views of GCM for TCGA, ENCODE, and GEO

Note that the Gene Expression Omnibus (GEO) source is at the same time a very rich public repository of genomic data (as most research publications include links to experimental data uploaded to GEO), but is also a very poor source of metadata, which are not well structured and often lack information; hence our mapping effort is harder and less precise for GEO than for the more organized TCGA and ENCODE sources[7]. The mapping to GEO captures as well the mapping to Roadmap Epigenomics, another relevant source of public data.

2.4 User-Friendly Interface

An important side effect of providing a global and integrated view of data sources is the ability to build user-friendly query interfaces for selecting items from multiple data sources. We show a mock-up of an interface that supports conjunctive queries over our entities, very similar to the user interface currently provided by DeepBlue [2] (Fig. 3); attributes are rendered by pop-up lists and values are then entered by users, with autocomplete support.

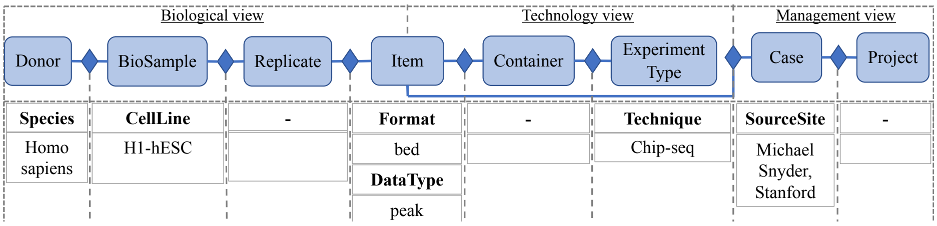

Fig. 3. Retrieval interface mock-up

3 Building the Integrated Repository

In this section, we describe high level rules for loading the content of the integrated repository from the original data sources, with a global-as-view approach. These transformations drive our approach.

3.1 Available Repositories at the Sources

Most genomic repositories offer Web interfaces for accessing their metadata. In addition, some of them offer Web APIs for querying the metadata, used for accessing storage structures for metadata (typically relational tables). Table 3 describes the schemas of the tables available at TCGA[8], ENCODE, and GEO[9]. TCGA and ENCODE tables result from the translation of a hierarchical json format representation (the only one provided by the sources) into a relational representation that has required several normalization steps and simplifications for illustration purposes. GEO tables result from a selection of a small subset of attributes used for mapping GEO to GCM.

[7] Textual analysis to extract semantic information from the GEO repository is reported in [12]; we plan to reuse their library.

[8] The metadata is provided in the NCI Genomic Data Commons portal, `https://docs.gdc.cancer.gov/Data_Dictionary/viewer/`.

[9] GEO information can be retrieved through the R package *GEOmetadb* [37].

Table 3. Relational schema of TCGA(T), ENCODE(E), and GEO(G) repositories

T.Case(**id**,project_id,disease_type,primary_site,tissue_source_site)
T.Project(**id**,name,program)
T.ClinicalDemographic(**id**,case_id,year_of_birth,gender,ethnicity,race)
T.BiospecimenSample(**id**,case_id,sample_type,tissue_type)
T.BiospecimenReadGroup(**id**,sample_id,platform_id)
T.DataFile(**id**,readgroup_id,workflow_id,data_type,data_format,size,experimental_strategy)
T.AnalysisWorkflow(**id**,workflow_type,workflow_link)
E.Donor(**id**,organism_scientific_name,age,sex,ethnicity)
E.Biosample(**id**,donor_id,biosample_type,biosample_term_name,health_status)
E.Replicate(**id**,biosample_id,experiment_id,biological_replicate_num,technical_replicate_num)
E.Experiment(**id**,assembly,assay_term_name,target,antibody,lab,award_project,platform)
E.File(**id**,experiment_id,output_type,file_type,file_size,pipeline_title,pipeline_accession)
G.File(**id**,gsm_id,file_type,size,download)
G.Gse(**id**,organization_name,contact,bioproject,experiment_type)
G.Gsm(**id**,gse_id,organism,age,gender,source_name,disease_state,genome_built,instrument_model)

3.2 Mapping Rules

Mapping rules are used to describe how data are loaded from the sources into the integrated repository; for illustration purposes, in Table 4 we provide some of the mappings, related to the Donor, BioSample, Case, and ExperimentType entities. Each mapping rule is a logic formula with variables in its left end side (LHS) which are computed from the variables in its right end side (RHS). The order of the LHS variables is the same reported in our global schema in Fig. 1 and the order of the RHS variables is the same reported in Table 3 for each source. As an example, the entity ExperimentType of the global schema is filled with data from ENCODE's entity Experiment, together with data from TCGA's BiospecimenReadGroup and DataFile (joined on the *readgroup_id* attribute), and data from GEO's Gse and Gsm (joined on the *gse_id* attribute).

As we already discussed in Section 2.3, the values of some of the attributes are acquired exactly as they are in the original source, others need the application of simple manually provided functions for textual transformation (denoted as *tr*), others are computed as textual combination of multiple source fields (denoted as *comb*, and finally others need manual curation (values are enclosed in inverted commas). As an example, Donor.*Ethnicity* corresponds to a combination of the attributes *race* and *ethnicity* of the ClinicalDemographic table, taken from TCGA source. *Tissue* and *CellLine* attributes of the BioSample are both produced by *biosample_term_name* of ENCODE which uses this attribute for both of them - the content of this attribute depends on the value of *biosample_type* (either "cell_line" or "tissue"). Relevant integration efforts are addressed towards defining a shared set of homogenized values for each attribute. The values of the global attributes, given to the LHS variables, are to be intended as already homogenized to the reference ontologies (as indicated in Section 2.2), or to the chosen finite restricted dictionaries. Notice that all the mappings preliminarily perform a value homogenization step, implicit in the integration process.

4 Related Works

A long stream of research tackled the problem of providing integrated access to multiple, heterogeneous sources. A survey of very preliminary works is [14].

Table 4. Examples of mapping rules for building the integrated repository from the sources

Donor(SID,SP,AGE,G,E)	$\supseteq$	E.Donor(SID,SP,AGE,G,E)
Donor(SID,“Homo S.”,$tr(Y)$,G,$comb(ET,R)$)	$\supseteq$	T.ClinicalDemographic(SID,_,Y,G,EY,R)
Donor(_,SP,AGE,G,_)	$\supseteq$	G.Gsm(_,_,SP,AGE,G,_,_,_,_)
BioSample(SID,T,$tr(BT)$,$tr(BT)$,$tr(HS)$,$tr(HS)$)	$\supseteq$	E.Biosample(SID,_,T,BT,HS)
BioSample(SID,“tissue”,PS,_,$tr(ST,TT)$,DIS)	$\supseteq$	T.BiospecimenSample(SID,CID,ST,TT), T.Case(CID,_,DIS,PS,_)
BioSample(SID,$tr(SN)$,$tr(SN)$,$tr(SN)$,$tr(D)$,$tr(D)$)	$\supseteq$	G.Gsm(SID,_,_,_,_,SN,D,_,_)
Case(SID,SS)	$\supseteq$	E.Experiment(SID,_,_,_,_,_,SS,_)
Case(SID,SS)	$\supseteq$	T.Case(SID,_,_,_,SS)
Case(SID,$tr(O,C)$)	$\supseteq$	G.Gse(SID,O,C,_,_,_)
ExperimentType(TE,$comb(TE,T)$,P,T,A)	$\supseteq$	E.Experiment(_,_,TE,T,A,_,_,P)
ExperimentType(TE,$tr(TE)$,P,_,_)	$\supseteq$	T.BiospecimenReadGroup($RGID$,_,P), T.DataFile(_,$RGID$,_,_,_,_,TE)
ExperimentType($tr(TE)$,$tr(TE)$,$tr(P)$,_,_)	$\supseteq$	G.Gse(EID,_,_,_,TE), G.Gsm(_,EID,_,_,_,_,_,_,P)

Buneman et al. [6] described the problem of querying and transforming scientific data residing in structured files of different formats. Along that work, BioKleisli [8] and K2 [9] describe early systems supporting queries across multiple sources. BioKleisli was a federated database offering an object-oriented model; its main limitation was the lack of a global schema, imposing users to know the structure of underlying sources. To improve this aspect, K2 included GUS (Genomics Unified Schema), an extensive relational database schema supporting a wide range of functional genomics data types. The BioProject [3] database was recently established to facilitate the organization and classification of project metadata submitted to NCBI, EBI and DDBJ databases.

A common approach in integrated data management is data warehousing, consisting of a-priori integration and reconciliation of data extracted from multiple sources, such as in EnsMart/BioMart [13,34]. Along this direction, [22] describes a warehouse for integrating genomic and proteomic information using generalization hierarchies and a modular, multilevel global schema to overcome differences among data sources. ER modeling (and UML class diagrams) were used in [5]; models describe protein structures and genomic sequences, with rather complex concepts aiming at completely representing the underlying biology. [35] is a biomedical data warehouse supporting a data model (called BioStar) capturing the semantics of biomedical data and providing some extensibility to cope with the evolution of biological research methodologies.

Many other works [29,16,25,10,15,18,21] present conceptual models for explaining biological entities and their interactions in terms of conceptual data structures. With our approach, similar to DeepBlue [2], we instead use conceptual modeling for driving the continuous process of metadata integration and for offering high-level query interfaces on metadata for locating relevant datasets, under the assumption that users will then manage these datasets for solving biological or clinical questions. Similarly to DeepBlue, we hide the data source differences so as to provide easy-to-use interfaces, but differently from them we disclose the semantic properties of the underlying sources and the metadata integration process; moreover, we cover a broader spectrum of sources and provide a richer set of concepts, including the management view.

5 Conclusions

The interest on an integrated repository for genomics stems from the huge amount of resources that are becoming available. In this paper we provide GCM, a genomic conceptual model capable of capturing the metadata of heterogeneous sources with a global-as-view approach. The model is supported by a method for conceptually designing global metadata through source attribute analysis and is validated by using three data sources: TCGA, ENCODE, and GEO.

Our GMQL system already provides access to datasets from TCGA, ENCODE, and Roadmap Epigenomics, that were identified as the most relevant in the course of collaborative projects with many biologists; we already developed some tools for automatically importing such datasets and for converting them to an integrated format, e.g., TCGA2BED [7]. Thanks to GCM, we can also provide a coherent semantics to the metadata of integrated sources; throughout the GeCo project we plan to add more sources, according to needs of biologists, and to continuously integrate their metadata within GCM.

Acknowledgement. This research is funded by the ERC Advanced Grant project GeCo (Data-Driven Genomic Computing), 2016-2021.

References

1. D. Adams et al. BLUEPRINT to decode the epigenetic signature written in blood. *Nature biotechnology*, 30(3):224–226, 2012.
2. F. Albrecht et al. DeepBlue epigenomic data server: programmatic data retrieval and analysis of epigenome. *Nucleic acids research*, 44(W1):W581–W586, 2016.
3. T. Barrett et al. BioProject and BioSample databases at NCBI: facilitating capture and organization of metadata. *Nucleic Acids Research*, 40(D1):57–63, 2012.
4. T. Barrett et al. NCBI GEO: archive for functional genomics data sets — update. *Nucleic acids research*, 41(Database issue):D991–D995, 2013.
5. E. Bornberg-Bauer and N. W. Paton. Conceptual data modelling for bioinformatics. *Briefings in Bioinformatics*, 3(2):166–180, 2002.
6. P. Buneman et al. A data transformation system for biological data sources. In *Int. Conf. on Very Large Data Bases*, pages 158–169, 1995.
7. F. Cumbo et al. TCGA2BED: extracting, extending, integrating, and querying The Cancer Genome Atlas. *BMC Bioinformatics*, 18(6):1–9, 2017.
8. S. B. Davidson et al. Biokleisli: A digital library for biomedical researchers. *International Journal on Digital Libraries*, 1(1):36–53, 1997.
9. S. B. Davidson et al. K2/Kleisli and GUS: Experiments in integrated access to genomic data sources. *IBM Systems Journal*, 40(2):512–531, 2001.
10. H. El-Ghalayini et al. Deriving conceptual data models from domain ontologies for bioinformatics. In *Information and Communication Technologies, 2006. ICTTA'06. 2nd*, volume 2, pages 3562–3567, 2006.
11. J. D. Fernández et al. Ontology-based search of genomic metadata. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 13(2):233–247, 2016.
12. E. Galeota and M. Pelizzola. Ontology-based annotations and semantic relations in large-scale (epi)genomics data. *Briefings in bioinformatics*, 2016.
13. S. Haider et al. BioMart Central Portal - unified access to biological data. *Nucleic Acids Research*, 37(Web Server issue):23–27, 2009.

14. T. Hernandez and S. Kambhampati. Integration of biological sources: Current systems and challenges ahead. *SIGMOD Rec.*, 33(3):51–60, Sept. 2004.
15. M. Idrees et al. A review: Conceptual data models for biological domain. *JAPS, Journal of Animal and Plant Sciences*, 25(2):337–345, 2015.
16. F. Ji, R. Elmasri, et al. Incorporating concepts for bioinformatics data modeling into EER models. In *ACS/IEEE Int. Conf. on Computer Systems and Applications*, pages 189–192, Washington, DC, USA, 2005. IEEE Computer Society.
17. A. Kaitoua, P. Pinoli, M. Bertoni, and S. Ceri. Framework for supporting genomic operations. *IEEE Trans. Computers*, 66(3):443–457, 2017.
18. M. C. Keet. Biological data and conceptual modelling method. *Journal of Conceptual Modeling*, 29(1):1–14, 2003.
19. A. Kundaje et al. Integrative analysis of 111 reference human epigenomes. *Nature*, 518(7539):317–330, 2015.
20. M. Lenzerini. Data integration: A theoretical perspective. In *Symp. on Principles of Database Systems*, PODS, pages 233–246, New York, NY, USA, 2002. ACM.
21. B. Louie et al. Data integration and genomic medicine. *Journal of Biomedical Informatics*, 40(1):5 – 16, 2007.
22. M. Masseroli, A. Canakoglu, and S. Ceri. Integration and querying of genomic and proteomic semantic annotations for biomedical knowledge extraction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 13(2):209–219, Mar. 2016.
23. M. Masseroli et al. GenoMetric Query Language: a novel approach to large-scale genomic data management. *Bioinformatics*, 31(12):1881–1888, 2015.
24. M. Masseroli et al. Modeling and interoperability of heterogeneous genomic big data for integrative processing and querying. *Methods*, 111:3–11, 2016.
25. F. Rechenmann. Data modeling: the key to biological data integration. *EMBnet.journal*, 18(B):59–60, 2012.
26. Anonymous paper. Accelerating bioinformatics research with new software for big data to knowledge (BD2K), Paradigm4, April 2015 (available `www.paradigm4.com`).
27. Consortium 1000Genomes. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073, 2010.
28. Consortium ENCODE. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 2012.
29. J. F. R. Román et al. Applying conceptual modeling to better understand the human genome. In *Int. Conf. on Conceptual Modeling ER*, pages 404–412, 2016.
30. A. Roy et al. Massively parallel processing of whole genome sequence data: An in-depth performance study. In *ACM Sigmod, Boston*, 2017.
31. S. Sarntivijai et al. CLO: the cell line ontology. *Journal of Biomedical Semantics*, 5(1):37, 2014.
32. I. Schomburg et al. BRENDA in 2013: new options and contents in BRENDA. *Nucleic Acids Research*, 41(Database issue):D764–D772, 2013.
33. L. M. Schriml et al. Disease Ontology: a backbone for disease semantic integration. *Nucleic Acids Research*, 40(Database issue):940–946, 2012.
34. D. Smedley et al. The BioMart community portal: an innovative alternative to large, centralized data repositories. *Nucleic Acids Research*, 43(W1):589–598, 2015.
35. L. Wang et al. BioStar models of clinical and genomic data for biomedical data warehouse design. *Int. J. Bioinformatics Res. Appl.*, 1(1):63–80, Apr. 2005.
36. J. N. Weinstein et al. The Cancer Genome Atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.
37. Y. Zhu et al. Geometadb: powerful alternative search engine for the gene expression omnibus. *Bioinformatics*, 24(23):2798–2800, 2008.