**GenData Model for Next Generation Sequencing**

**S. Ceri, S. Dahal, M. Masseroli, F. Palluzzi, F. Venco**

**DEIB, Politecnico di Milano**

**Contribution to GenData and IEO-IIT-PoliMi Projects – 25-4-2013**

**Summary**

Every genomic research center has huge amounts of data resulting from experiments conducted in the context of different research units, but their use is often confined within the group and their observation is limited to specific portions of the genome which are the specific target of each biological experiment. The purpose of the **GenData Model** **(GDM)** is to provide a uniform description of experimental data so as to enable a query and analysis process that can combine results from a variety of experiments, across the boundaries of specific units, groups, or even research centers – we regard such a process as **High-order Query and Analysis (HQA)**. The ultimate goal of HQA is to enable the discovery of “new” biological phenomena (e.g., new promising regions of the genome, new correlations between biological features, new clustering of experimental observations based upon patients’ phenotypes) that could not be observed at the small experimental scale.

The GDM should at the same time synthesize experimental data and correlate them to the conditions which have produced them, so as to enable HQA. The objective in designing GDM is **raising the level of abstraction** with regard to current bioinformatics data formats, which are very effective in encoding experimental results, but are not targeted to data interoperability. This document describes a logical format of GDM which precisely explains the data semantics of each involved entity and attributes, disregarding performance-related issues, such as data compression and indexing – but such ingredients will be essential for guaranteeing scalability of the solution.

The design of GDM is driven by contrasting objectives. On one hand, it has to be compact, so as to enable efficient data storage and sharing; but on the other hand, it must suit to describing a great variety of experimental processes upon NGS data (in our joint work with IEO-IIT we recognized such processes as DNA-seq, RNA-seq, ChIP-seq, and DNA methylation). We deal with such contrasting objectives by identifying a **core GDM**, identical throughout all the experimental processes, and then allowing process-specific **GDM extensibility**. For instance, core GDM includes a description of DNA regions which applies to any experimental process, while the description of specific aspects of a clinical protocol is not included in the core GDM and can only be added to the core by using extensibility.

We do not want either to replace the current formats for genomic data storage or to replace the data analysis tools that apply to single experiments; our effort is focused on providing a meta-model (or ontology) for *data interoperability at a larger scale* – across experiments; rather than interfering with the current experimental pipelines that are available at each research center, our work is localized downstream of such pipelines and uses their results as inputs. Indeed, the best way of achieving our goals is to describe the results of HQA in the classic formats (such as BED or BIGWIG files) so that they can be subject of further analysis using conventional tools, or communicated using classic methods (e.g., DAS protocol), or observed by using classic viewing methods (e.g., UCSC or IGB genome browser).

The main difficulty in dealing with NGS data is scalability; therefore GDM should not only represent experimental data at its greater level of detail, but also be able to summarize them. For this purpose, GDM introduces synthetic representations of the genome that enable standardized summarization of experimental data at different levels of detail. Towards this aim and for efficient query execution GDM uses binning systems.

Another important aspect of GDM is that it focuses only on the representation of experimental data and not of the known biological features that can be associated with them – in terms which are familiar to biologists, we represent only “custom tracks” and not “annotation tracks”. However, we are aware that certain HQAs are not possible without superimposing biological features to experimental data; our approach to deal with this aspect is to allow for ad-hoc enhancement activities which add biological features to GDM data in preparation for HQA activities.

The GDM has been conceived having in mind not only data representation issues, but also the involved processes for data production and use. Such processes address:

1. **Populating the GDM.** If we assume that experimental data are encoded into files produced by classic bio-informatics processes (such as data alignment or peak finding) then the GDM must include “backward” information about the context that has produced such files and “forward” information about expressing the content of the files in GDM terms. This task is performed by *data loaders*.
2. **Extending the GDM.** Extensibility is concerned with adding new type of experiments or adding new descriptive elements of the experiment context, e.g. about the cell-line, tissue, organism, phenotype involved in the experiment. This task is performed, at the schema level, by *data extenders*.
3. **Enhancing the GDM.** While GDM is focused on experimental data, the biological meaning of data cannot be explained in the lack of integration of experimental data with known biological features (e.g., the positions of known genes). Thus, we will need to describe processes for GDM enhancementthat integrate GDM data with known biological features, extracted from existing databases and ontologies, that are required for specific HQA. This task is performed by *data enhancers*.
4. **Querying the GDM.** Once information from different experiments is described in GDM, it is possible to support queries upon the GDM that extract experimental data or their summaries. Two orthogonal kinds of queries are possible: *context queries*, about the relationship between experiment data and their context, and *genometric queries*, about the mapping of experiment data to genomic regions, possibly across different bidding systems and using alternative alignments. Genometric queries may require suitable extensions of conventional query languages. This task is performed by *data extractors.*
5. **Analyzing the GDM.** Queries identify specific DNA regions of specific experiments open up the possibility of a variety of new-generation data analysis methods, as well as the possibility of performing classic analysis across broader, enhanced and contextualized data.
6. **Viewing the GDM.** Once data are extracted by queries, enhanced with genomic features, and analyzed by new analysis methods, they can be reduced to standard representations (e.g., BED or BIGWIG files) for visualization using genome browsers.

Building a GDM is the first step of very ambitious projects that could have a great impact on NGS data processing. Besides supporting the integration among the experiments which are performed at a specific research center (such as IEO-IIT), it is an “enabler” of important research directions, which are sketched below:

1. **High-scale data analysis.** Extended GDM data may give rise to new classes of data analysis methods, which add context data to genomic and use GDM ability to support multiple levels of data abstractions (and the corresponding geometric systems).
2. **Integration of query and analysis with genome browsing.** The genome browser, used as viewer of enhanced GDM data, can be used also for query expression, e.g. by expressing patterns that can be translated into geometric queries and extract target regions.
3. **Supporting interoperability across research centers.** Programs for data population can evolve into indexing methods for extracting GDM-encoded information from files in distributed repositories of NGS data. Such distributed indexing hints to a variety of important applications, ranging from peer-to-peer interaction between research centers, to world-wide indexing and extraction of data for building genomic search systems.

**1. NOTATION**

Although we refer to the schema elements as entities interconnected by binary relationships, we describe each entity at a logical level, using a relational format, with an indication of primary and foreign keys. This choice is motivated by our requirement of being at a level of detail that allows the precise description of operations of load, enrichment, extraction as well as the development of the query language and the description of query results. Note that we are not assuming at this stage a relational implementation.

We use sub-entities as a short-hand notation for denoting several relations (one for each sub-entity) which have a core part in common (the entity component, which incorporates the entity key). In particular, sub-entities are useful for modeling specific descriptions; the sub-entity representation is a mechanism for scheme extensibility.

All the attributes that are primary keys are preceded by the symbol **PK** and are at the top of the entity. Primary keys of entities internally generated within the GDM system are characterized by the \_ID suffix. We identify keys of genomic regions with a \_COORD suffix, they are built by a quintuple of values which include the encoding of <chromosome, start, stop, strand, phase>. Keys may include positive numbers (of type integer) which are progressively generated.

We denote foreign keys with the symbol **FK**; external keys are included within a dependent (child) entity and establish an existential dependency of the instances of that entity to the corresponding instances of a parent entity. In GDM, foreign keys can be not included in the primary key of the dependent entity and link to the key of the parent entity. Connectors are drawn from the specific foreign key attribute, to make each link explicit.

Attributes may be encoded as URL and suffixed by \_URL. Some other attributes connect entity instances to specific instances of external collections (ontologies or table schemes); in such case, they have a \_REF suffix and are constituted by a triple <identifier, identifier name, provider>, where *provider* denotes the specific collection (e.g. UCSC\_TABLES, GO), *identifier name* is the name of the type of element identifier used in the collection, *identifier* a specific instance of collection element.

Required attributes are **bold**; all other attributes are optional and therefore may be missing. Complex attributes are followed by the symbol **[**]; they correspond toseveral <attribute-value> pairs which are not further described in the schema, but which can be used in the query language.

The minimum and maximum cardinality of the binary relationships used in the schema are described by a graphic notation. Assuming a binary link between entities A and B, the use of the following symbols in the arc connecting A to B, close to entity B, indicates that:

Every instance of A is connected to one and only one instance of B.

Every instance of A is connected to zero or one instance of B

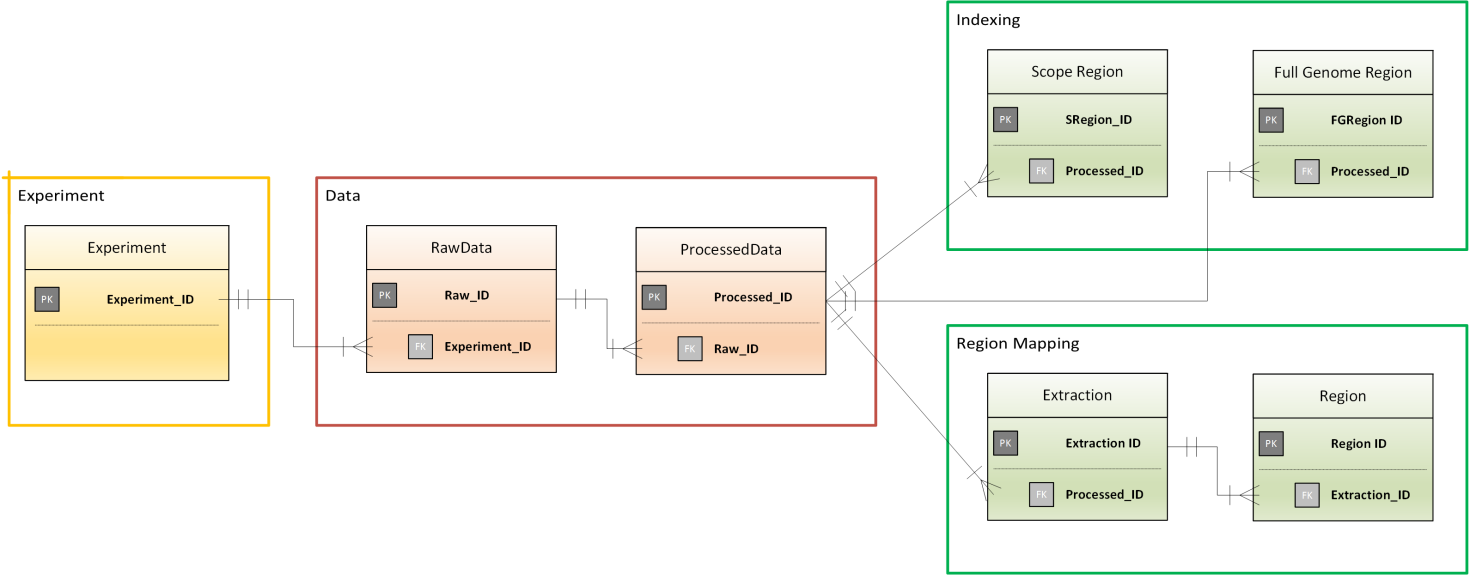
Every instance of A is connected to one or more instances of B.

Every instance of A is connected to zero or more instances of B.

**2.** **SCHEMA OVERVIEW**

The GDM model is divided in four parts, which allow for a left-to-right reading.

* The first part describes the **experiment sub-schema**, connecting NGS experimental data to the biological context and conditions under which the experiment has been produced.
* The second part describes the **data sub-schema**, which describes the data files which are produced by NGS experiments, and the dependencies among them reflecting the chain of transformations between such files. The workflow of data production connects an experiment to several raw files, each of which can then be transformed into several processed files, which include aligned files (whose data is aligned to a reference genome) and annotated files (which are produced by data analysis algorithms).
* The third part describes the **region mapping sub-schema**, representing information of specific genomic regions extracted from aligned and annotated data files. At a bird eye’s view, it includes a description of the extraction method used and a description of the extracted genomic regions.
* The fourth part describes the **indexing sub-schema**, linking genomic regions which either represent fixed genome segmentations or known ontological features to the data subschema. The indexing sub-schema can also be applied to the regions describing ontological features or to certified experimental data extracted from trusted data sources to the data sub-schema, thereby allowing the interplay of experimental data with a priori knowledge.

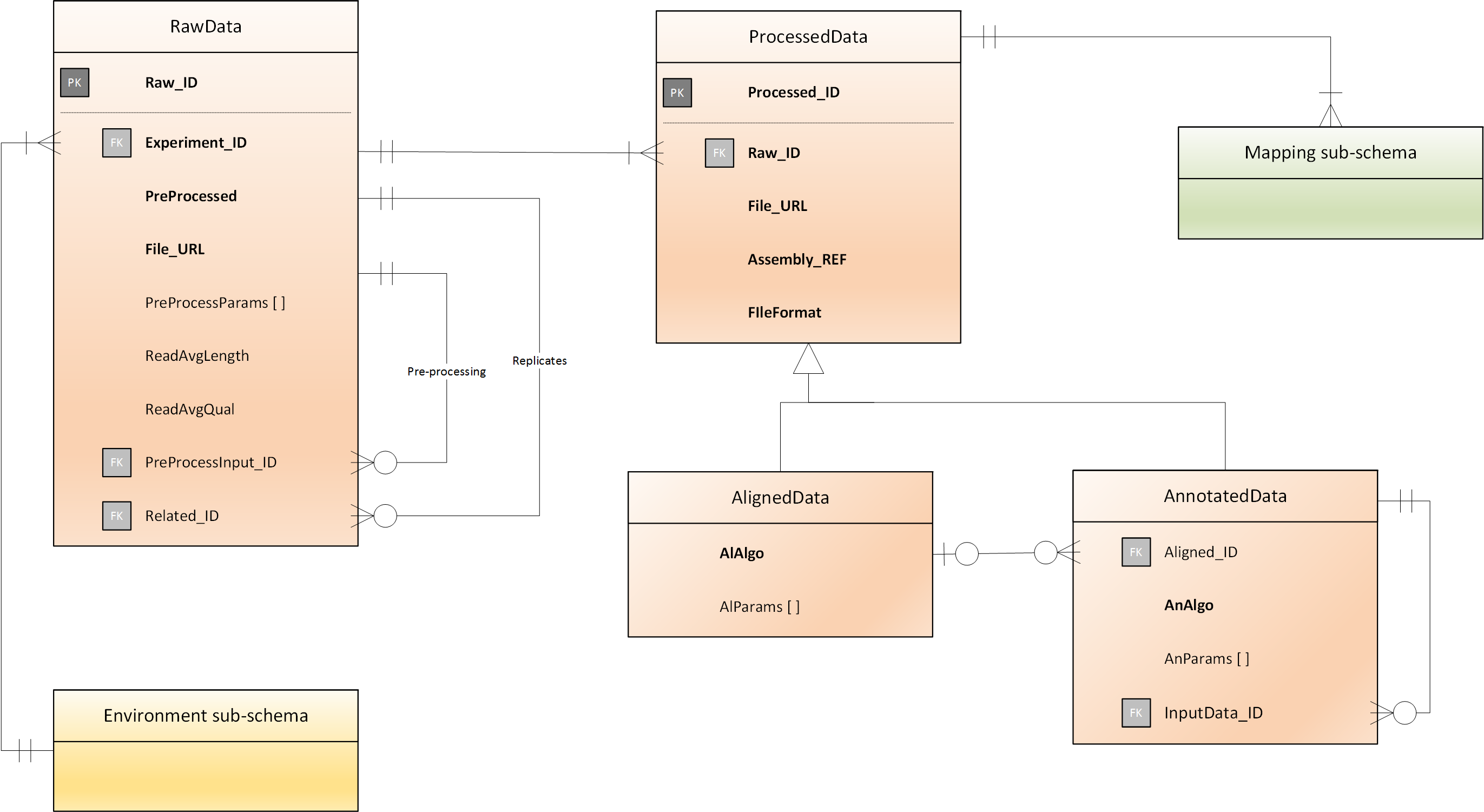


In our description, we start with the data and genomic region sub-schemas, which are core of the GDM model, and continue with the experiment sub-schema and the indexing sub-schema. References to external entities, referred by the \_REF suffix, are considered to be external to the GDM and not further described, although a specific instantiation of GDM may add entities and attributes of such external entities.

**2.1 DATA SUB-SCHEMA**

The **RawData** entity describes the output produced by a sequencing experiment. Each run of a sequencing machine usually produces more than one file, thus the connection between the experiment (in the experiment sub-schema) and the RawData entity is one-to-many. The mandatory attribute *File\_URL* indicates the URL of the data file; when the file is not associated with a URL, a filename\_path in a known repository space is used. The optional attributes *ReadAvgLenght* and *ReadAvgQuality* describe the average length and the average quality of the reads respectively.

Each file produced can be pre-processed to improve the overall quality of the read sequences. For this reason, many *RawData* instances can be obtained as subsequent applications of pre-processing techniques. Preprocessed is a Boolean mandatory attribute which is set to FALSE by default; when it is set to TRUE, the *PreProcessParams* attribute contains the parameters used as input to the pre-processing algorithms and the foreign key *PreProcessInput\_ID* indicates, for a pre-processed file, the file from which such pre-processing was performed. Note that describing pre-processing steps is not mandatory: a legal modeling can simply contain all the RawData instances connected to a given sequencing experiment without describing their mutual dependencies. When files are replicas, they are interconnected.

****

The **ProcessedData** entity describes the files which are produced from RawData after alignment or further data analysis. Each raw data file usually produces more than one processed file, thus the connection between them is one-to-many. The mandatory attribute *File\_URL* indicates the URL of the data file; when the file is not associated with a URL, a filename\_path in a known repository space is used. The mandatory attribute *Assembly\_REF* indicates the particular assembly (reference genome) that was used for the alignment. Finally, the mandatory attribute *FileFormat* indicates the specific data format used (e.g., SAM, BAM, BED, WIG, BIGWIG).

Processed data can belong to one of two different sub-entities: AlignedData and AnnotatedData. **AlignedData** contains read sequences from a raw file aligned to a reference genome. **AnnotatedData** are obtained from aligned data through some subsequent analysis algorithms (e.g. a pick calling algorithm such as MACS). Thus, aligned and annotated data sub-entities are connected by a one-to-many relationship: from each aligned data, different analysis algorithms can be used to produce different annotated files. The binary relationship is modeled by adding to AnnotatedData the *Aligned\_ID* attribute, whose values are taken from the domain of the *Processed\_ID* attribute. As aligned data are produced by an alignment algorithm *AlAlgo*, which can be further described by its parameters (AlParams[]), annotated data are produced by an annotation algorithm *AnAlgo*, which can be further described by its parameters (AnParams[]).

In addition, each file resulting from an analysis can be further processed by a different analysis. Thus, instances of the AnnotatedData sub-entity may be connected by a one-to-many relationship: from each data analysis, different analysis algorithms can be used to produce other annotated files. Such AnnotatedData instances are connected to the annotated data from which they are derived using the optional *InputData\_ID* attribute, whose values are taken from the domain of the *Processed\_ID* attribute*.*

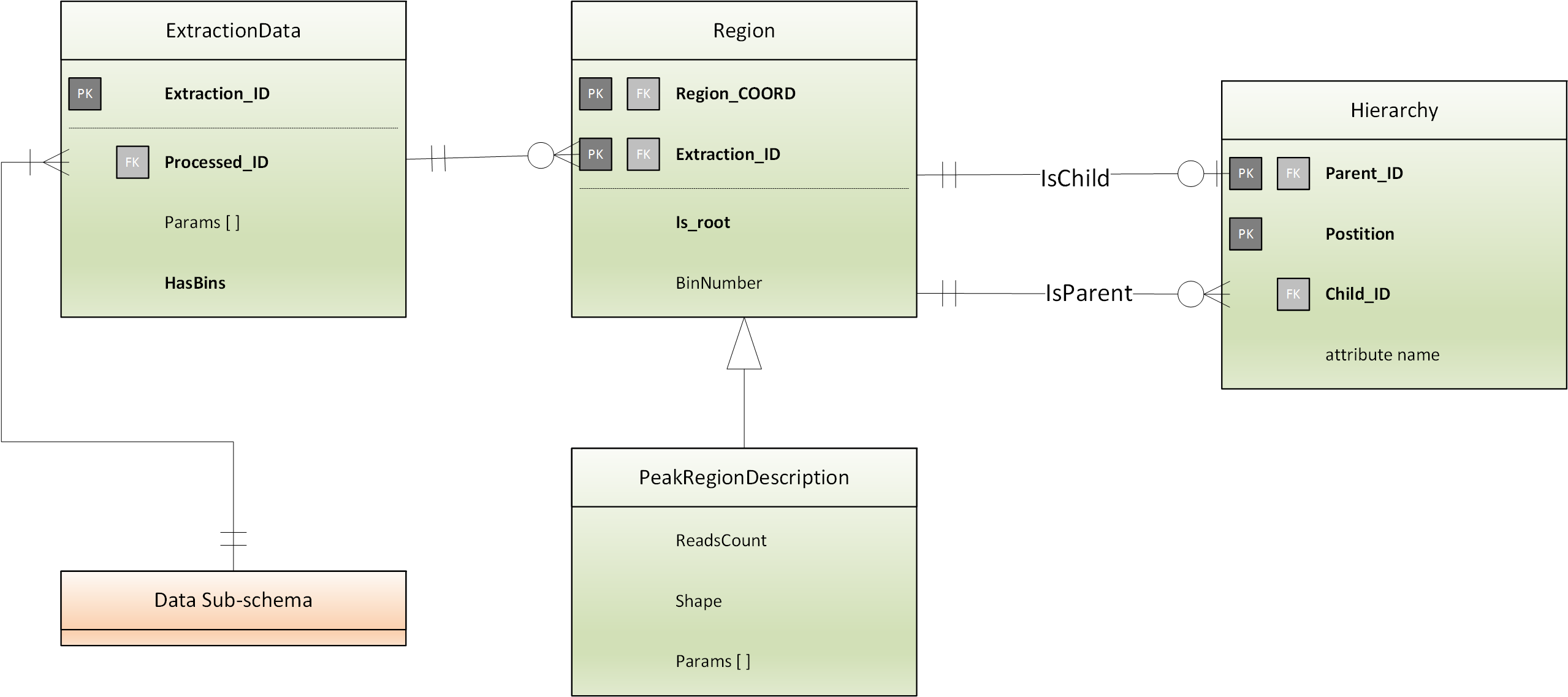
Note that describing derivation steps is not mandatory: a legal modeling can simply contain all the ProcessedData instances connected to a given RawData entity without describing their mutual dependencies.

**2.3 REGION MAPPING SUB-SCHEMA**

The region mapping subs-chema describes how the various experimental features which are revealed by an experiment are mapped to specific regions of the genome. The elements of this schema are built by data extractor programs. Each extraction program applies to a processed data file; we denote as experiment’s *target regions* those genomic regions where the experiment results are statistically significant according to the extraction parameters.

The **ExtractionData** entity denotes an activity performed by an extractor program on a processed data file. Each extraction is linked to one specific instance of ProcessedData and can be further described by extraction parameters *Params[]*, in <attribute-value> format, which are used to drive the extraction process.

The **Region** entity is used to represent the genomic coordinates of the result of data extraction at the maximum level of detail. A *genomic coordinate* consists of a quintuple giving the Chromosome, the Start and Stop position (usually separated by the “:” symbol to denote intervals), the Strand (1 positive, -1 negative, 0 for both – the usual case), the Phase(i.e., the sequence reading frame: 1, 2, or 3; possibly missing as it is meaningful only for coding sequences). We use inter-base coordinates, thereby allowing the specification of regions of zero bases (e.g., 123:123), which are useful to denote insertion points. Optional coordinate elements, which enable simplified encodings, are possible, allowing for the lack of strand and phase, or for regions spanning across chromosomes. Attributes describing genomic coordinates are suffixed with \_COORD; their values is a unique identifier within GDM obtained using suitable encoding of genomic coordinates; functions *Chrom, Start, Stop, Phase*, and *Strand*, once applied to a coordinate attribute, return the corresponding values. An optional BinNumber attribute is used to describe the smallest region of the binning system that contains the region; we use the UCSC binning system. A **RegionDescription** sub-entity can describe region data which depend upon the specific kind of observed region. In particular, when regions describe experimentally identified peaks, it has the attributes *ReadCount* and *Shape;* other parameters *Params[]* (such as FoldEnrichment, P-Value, FDR, Q-Value) can be added for specific regions as attribute-value pairs of the *Params[]* attribute.



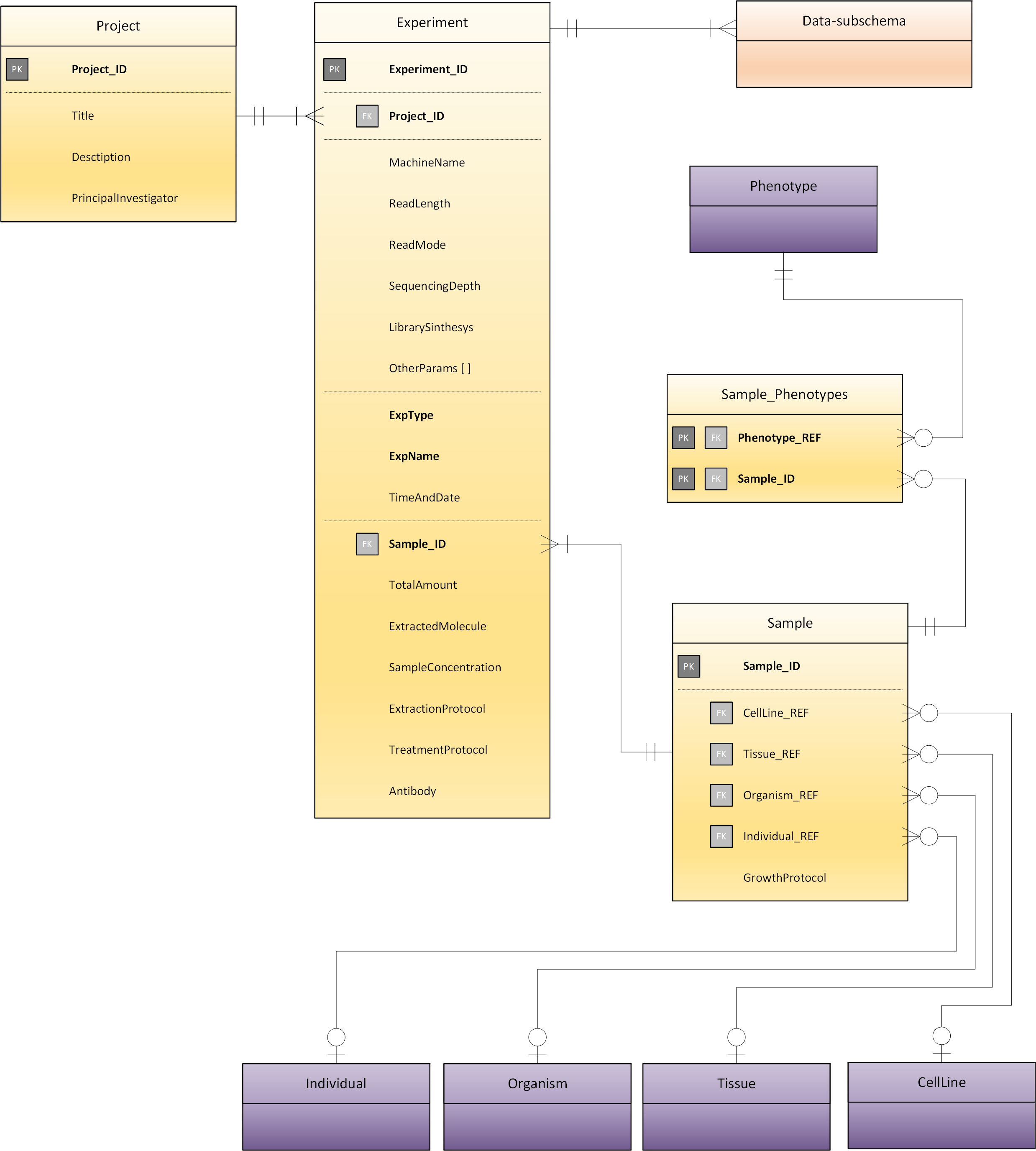
Some regions can be defined as a root (by setting the *Is\_root* attribute to true); root regions are associated with hierarchies in the GDM, represented by the **Hierarchy** entity. Every instance of the Hierarchy entity represents an edge of a tree; the key is the pair <*Parent\_ID, Position*> which allows to extract all the children of a given node; thus, the Hierarchy instances support tree traversal in either direction. Note that in the specific situation each key is a pair <Region\_COODR, Extraction\_ID>. The use of entity hierarchies is optional, as indicated by the cardinalities of the relationships *IsParent* and *IsChild*.

**2.3 EXPERIMENT SUB-SCHEMA**



The experiment sub-schema is inspired by the MIAME, a standard defined for microarrays (<http://www.mged.org/Workgroups/MIAME/miame.html>) which is still a reference standard for experiment description. A similar standard for NGS experiments is still in draft format (<http://www.mged.org/minseqe/MINSEQE.pdf>). The experiment sub-schema is also inspired by the data schema of Gene Expression Omnibus (GEO), as it includes all the information required to upload experiments on GEO repository (see <http://www.ncbi.nlm.nih.gov/geo/info/overview.html> <http://www.ncbi.nlm.nih.gov/geo/info/GEOHandoutFinal.pdf>); GEO main concepts (Platform, Samples, Series) can be reconstructed from the sub-schema entities.

Experiment is the central entity of the sub-schema; each experiment consists in the use of a NGS sequencing technique on a biological sample, producing a raw data file set, performed in the context of a project. Many of the attributes of this schema are optional, as we cannot expect that a registration protocol at each research center will enforce the production of all the corresponding information.



The **Experiment** entity has an *ExpType* (e.g., DNAseq, RNAseq, ChIPseq), an *ExpName* (e.g., TF region, histone marker, gene expression, …), a *MachineName* (e.g., Illumina 2000), and relates to a project through the external key *Project\_ID*. Other optional attributes describing the NGS process include *ReadLength*, *ReadMode, SequencingDepth, LibrarySynthesis* and *Time (of the sequencing);* *OtherParams[]* takes free <attribute-value> format. The foreign key *Sample\_ID* connect the experiment to the specific biological sample used. The same biological sample, although treated differently, is usually used in different experiments and projects. Sample extraction is described by the attributes *ExtractedMolecule* (e.g., DNA), *TotalAmount*, *SampleConcentration, ExtractionProtocol, TreatmentProtocol* and *Antibody*.

The **Sample** entity describes a biological sample. The optional attribute *GrowthProtocol* label the protocol used to grow the cells of the sample, if applicable. All the remaining attributes provide references to external data sources. Each sample is associated with an *Organism* (e.g. human or mouse), a *Tissue* (i.e., the specific biological part in a complex organism), a *CellLine* (the specific cell culture), an *Individual* (e.g., a given mouse), and several *Phenotypes*.

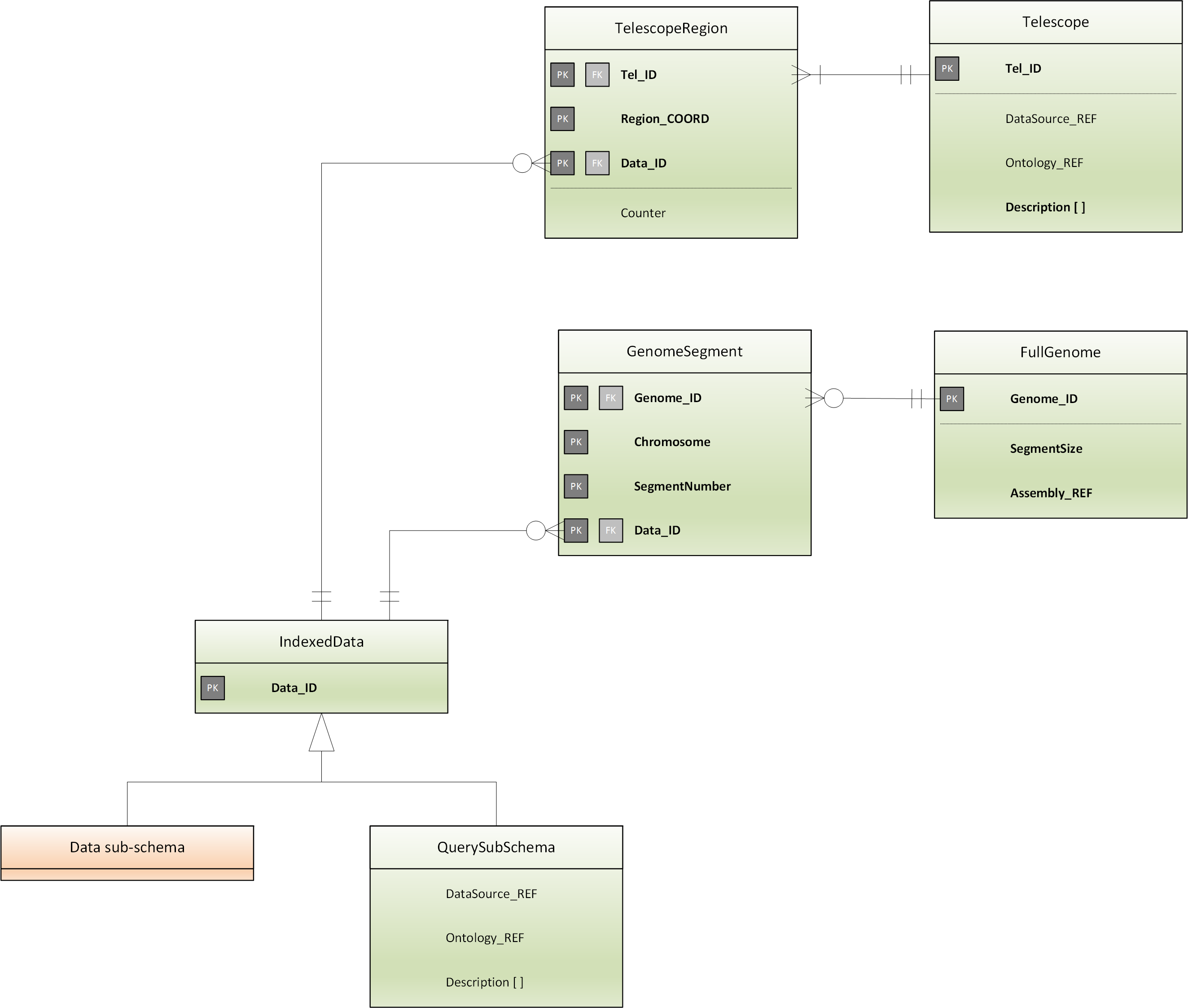
The **Project** entity connects experiments belonging to the same research. A project may have a *Title*, a *Description* and a *PrincipalInvestigator*; other parameters take free <attribute-value> format.

Information in this schema partially maps the Laboratory Information Management System at IEO-IIT and is subject to further discussion with biologists; its design is subject to trade-offs which depend upon the possibility of inferring its content using data loaders, that will in any case require center-specific implementations.

**2.4 INDEXING SUB-SCHEMA**

(Tele)-scopes describe sets of regions of interest for supporting the query process. Each telescope may describe DNA positions with given biological meanings, e.g., the regions associated with given genes or mutations; they can also be created as result of data analysis (e.g., regions enriched in reads of a given experiment) and then be used for retrieving genomic information from another experiment. Two kinds of indexing are possible:

* *Summary indexing* produce the regions of the telescope with a non-empty intersection with the regions identified in an experiment; each such region has an associated weight, which is a measure of how the experiment is expressed in the region (for instance, it can be a counter of peaks or of mutations).
* *Binary indexing* associate to each region of the telescope a Boolean bit indicating whether at least one of the experiment results falls in that region.

****

Full genomeindexing consider a whole reference assembly; the assembly is divided into segments of the same size, and each segment has a bit set to 1 when at least one of the experiment results falls in that segment.

The **Telescope** entity either refers to regions extracted from external certified data sources (through a *DataSource\_REF*) or to ontological knowledge which in turn is interpreted as a sequence of regions of interest (though an *Ontological\_REF*). Each telescope is relative to an *Assembly*, has a *Description* which may include information such as the telescope’s *Name*, *Creator* and *Time of Creation*.

The **TelescopeRegion** entity describes a given region (whose coordinates are defined in the *Region\_COORD* attribute) of a given telescope which is used for indexing aligned data, referred through their *Data\_ID*; such reference is either to experiment ProcessedData or to other data, which are external to GDM, but are used for expressing queries to the GDM. The *Counter* attribute reports the number of experiment items falling in a telescope region.

The **QuerySubSchema** entity introduces such external data by indicating whether it is a certified data source (using the *DataSource\_REF* attribute) or an ontological knowledge (though the *Ontological\_REF* attribute).

The **FullGenome** entity refers to a whole reference assembly, defined as external reference through the attribute *Assembly\_REF*; the *SegmentSize* attribute indicates the size of the equal size segments which are used to subdivide the genome.

The **GenomeSegment** entity has a compound key formed by the FullGenome key (*Genome\_ID*), the *Chromosome*, the *SegmentNumber* and the key of the extracted data file, and no attributes; a GenomeSegment instance exists if the extraction process produces at least one target region.

Of course, the relational representation of indexes is hereby introduced for explaining their nature, whereas their actual storage will be optimized. The use of bidding systems both in the GenomeSegment and in the TelescopeRegion may speed up genometric queries.

**REFERENCES**

*Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D.: “****The human genome browser at UCSC****”. Genome Res. 2002 Jun;12(6):996-1006*

*Meyer LR, Zweig AS, Hinrichs AS, Karolchik D, Kuhn RM, Wong M, Sloan CA, Rosenbloom KR, Roe G, Rhead B, Raney BJ, Pohl A, Malladi VS, Li CH, Lee BT, Learned K, Kirkup V, Hsu F, Heitner S, Harte RA, Haeussler M, Guruvadoo L, Goldman M, Giardine BM, Fujita PA, Dreszer TR, Diekhans M, Cline MS, Clawson H, Barber GP, Haussler D, and Kent WJ.: “****The UCSC Genome Browser database: extensions and updates 2013****”. Nucleic Acids Res. 2012 Nov 15*

*Rosenbloom KR, Sloan CA, Malladi VS, Dreszer TR, Learned K, Kirkup VM, Wong MC, Maddren M, Fang R, Heitner SG, Lee BT, Barber GP, Harte RA, Diekhans M, Long JC, Wilder SP, Zweig AS, Karolchik D, Kuhn RM, Haussler D, Kent WJ.: “****ENCODE Data in the UCSC Genome Browser: year 5 update”.*** *Nucleic Acids Res. 2012; doi: 10.1093/nar/gks1172.*

*Danford T, Rolfe A., Gifford D.:* ***“GSE: a conprehensive database system for the representation, retrieval, and analysis of microarray data”.*** *Pac Symp Biocomput. 2008:539-50 2008*

*Chen JY, Carlis JV.: “****Genomic Data Modeling****”. Information Systems - Special issue: Data management in bioinformatics archive. Volume 28 Issue 4, June 2003*

*Pastor O, Casamayor JC , Celma M, Mota L, Pastor MA, Levin AM: “****Conceptual Modeling of Human Genome: Integration Challenges****”.* Conceptual Modeling and its Theoretical Foundations 2012: 231-250

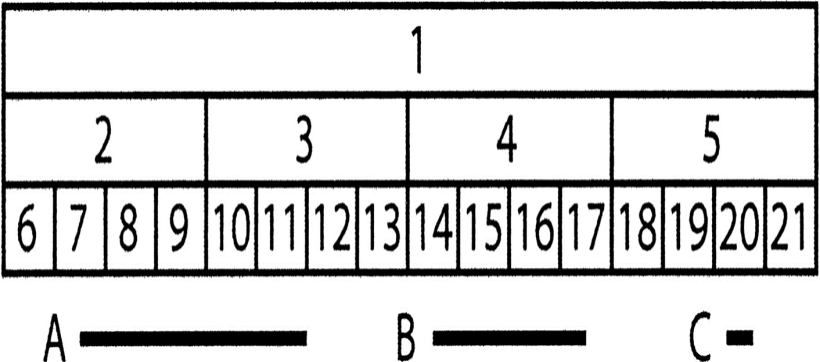
*Lifschitz S, Viana CJM, TristaoC , Catanho M, Degrave W, de Mirand ABa, Bezerra M, Otto TD: “****Design and Implementation of ProteinWorldDB****” BSB 2012, 144-155, Springer, 2012.*

*Bafna V, Deutsch A, Heiberg A, Kozanitis C, Ohno-Machado L., Varghese G.:* ***Abstractions for Genomics****, Communications of the ACM, Vol. 56 No. 1, Jan. 2013, Pages 83-93.*

**APPENDIX 1: BINNING SYSTEMS**

**Binning schemes** are particularly useful for geometric queries; the following note is taken from the article on the Human Genome Browser at UCSC.

Binning scheme for optimizing database accesses for genomic annotations that cover a particular region of the genome. This diagram shows bins of three different sizes. Features are put in the smallest bin in which they fit. A feature covering the range indicated by line A would go in bin 1. Similarly, line B goes in bin 4 and line C in bin 20. When the browser needs to access features in a region, it must look in bins of all different sizes. To access all the features that overlapped or were enclosed by line A, the browser looks in bins 1, 2, 3, 7, 8, 9, 10, and 11. For B the browser looks in bins 1, 4, 14, 15, 16, 17. For C, the browser looks in bins 1, 5, and 20.



**APPENDIX 2: REFERENCE DATA SCHEMES**

**A2.1 DAS SERVER[[1]](#footnote-1)[[2]](#footnote-2)**

The **Distributed Annotation System** (**DAS**) is a network of server and client software installations distributed across the Web. The DAS protocol is a standard mechanism through which clients can communicate with servers in order to obtain various types of biological data. The protocol defines:

* The communication method
* The query model
* The data format

By enforcing these constraints, DAS allows a client to integrate data from many diverse sources implementing the protocol at a saleable development cost. The DAS network of servers comprises a *registry*, several *reference servers* and several *annotation servers*.

**Reference Objects**: are items of data with stable identifiers that are targets for annotation. For example, “P15056” refers to a protein sequence upon which annotations can be based. Similarly, “chromosome 21” refers to a DNA sequence.

**Coordinate System**: is a stable, logical set of reference objects. It provides a mechanism to uniquely identify reference objects that share identifiers, such as chromosomes. For example, chromosome 21 might identify several reference objects from different species, but only one within the NCBI 36 human assembly. An example of coordinate system is presented below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Authority** | **Version** | **Species** |
| Chromosome | NCBI | 36 | Homo sapiens |
| Scaffold | ZFISH | 7 | Danio rerio |
| Protein sequence | UniProt | - | - |

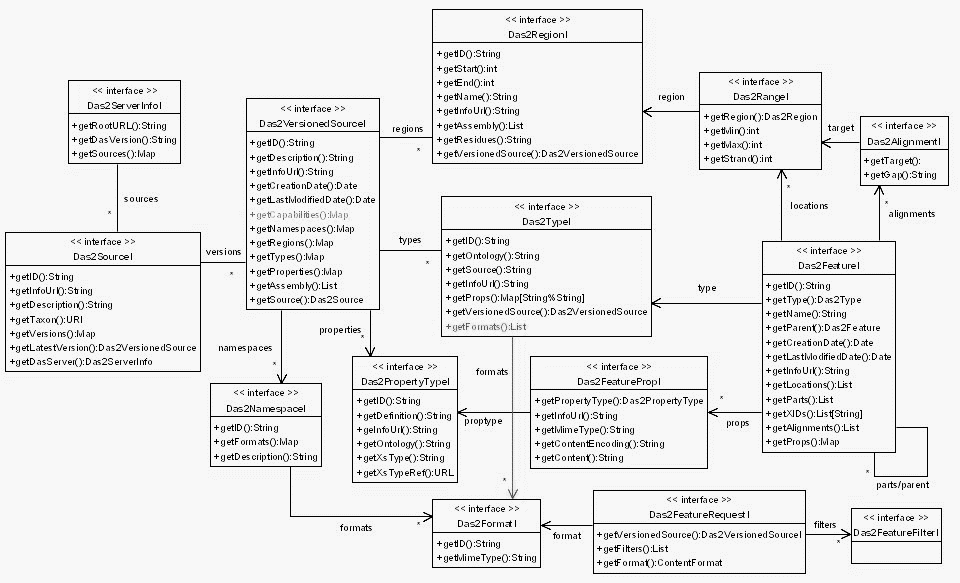
**Reference and Annotation Servers:** a reference server is a DAS server that provides core data for the reference objects in a particular coordinate system. Annotation servers are specialized for returning a list of annotations for the reference objects within a coordinate system. The distinction between reference and annotation servers is conceptual rather than physical, as a single server instance can play both roles, by offering both sequences and annotations of those sequences.

**DAS Registry**: it catalogues and describes the capabilities and coordinate systems of DAS servers, allows discovery of available DAS services, automatically validates registered DAS sources to ensure that they correctly implement the protocol, and provides a mechanism for activating or highlighting individual DAS services in clients.

**Client**: a DAS client typically integrates data from a number of DAS servers, making use of the different data types. For e.g., a client may implement the following procedures for a particular sequence location:

* Contact DAS registry to find reference and annotation servers for a particular genomic assembly
* Obtain sequences from a reference server
* Obtain sequence features from annotation servers
* Display the annotations in the context of the sequence

Below is the DAS/2 UML modeling schema.

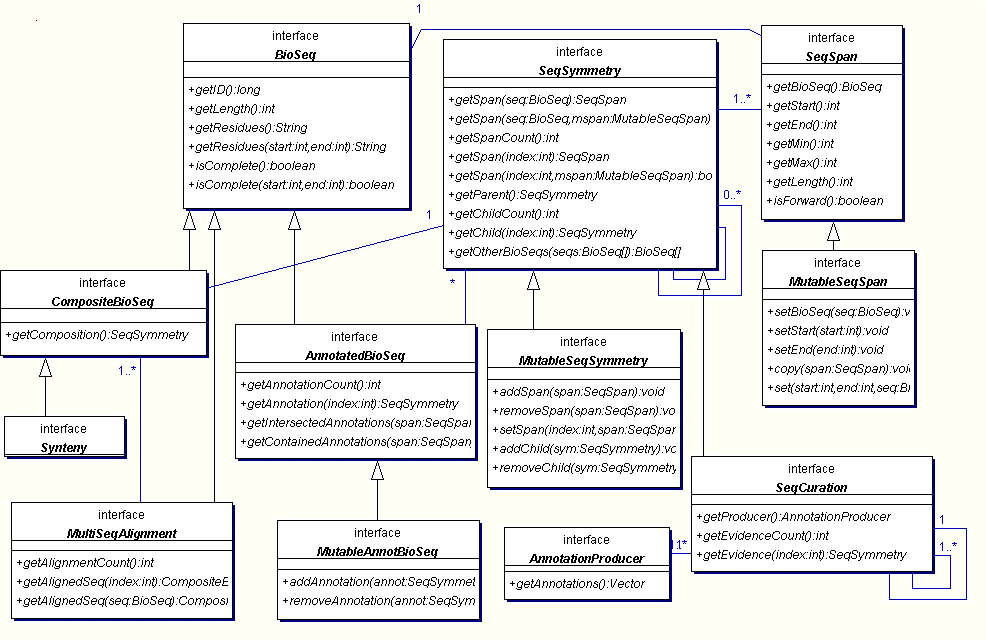


In the above schema, the *Das2Source*, *Das2Region*, *Das2Feature* and *Das2Type* interfaces comprise the core model, which gives the source information of each server, regions contained in each source, features of each region and type of features respectively. The *Das2Feature* interface has part/parent relationship, as shown in the extreme right of the figure above.

The basic query types of DAS are *Source*, *Segments*, *Types* and *Features*. The *Source* lists available genome in a server, *Segments* lists chromosomes per genome, *Types* lists types of annotations (file format, etc.) available and the *Features* lists annotation details in specific regions.

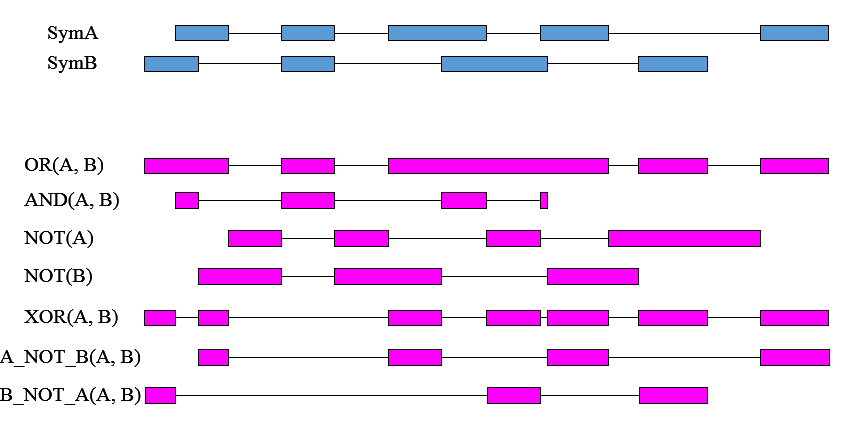
**2.2 IGB[[3]](#footnote-3)**

The **Integrated Genome Browser** (**IGB**) is an interactive, zoom able scrollable software program which can be used to visualize and explore genome-scale data sets, such as tiling array data, NGS results, genome annotations, microarray designs and the sequence itself. The IGB interacts with DAS using the underlying object-oriented data model called the Genometry, which is also the underlying data model of the IGB.



The figure above shows the Genometry model of the IGB, where the central concept is the **SeqSymmetry**, which is composed of **SeqSpan** that represents the breadth of the sequence and the depth is represented by the hierarchy of parent and child relationships, as shown in the interface SeqSymmetry in the figure above. The model was conceived2 with a desire for a unified data model to represent relationships between biological sequences, such as *annotation*, *alignments* and *sequence composition*. The core model consists of **BioSeq**, which represents the length and residues, **SeqSpan**, which represents the start and end of a bio-sequence, and the **SeqSymmetry** that comprises SeqSpan, which is the breadth, and the parent-child hierarchy, or the depth, as depicted in the figure below.

The Genometry model can be manipulated using various logical operations over the symmetry, like intersection (AND), Union (OR), Inverse (NOT), Mutual Exclusive (XOR), Exclusion (A\_NOT\_B, B\_NOT\_A), Projections (Flattening), Summary (Landscapes) and Transformation/Mapping. Below is an illustration of Symmetry Boolean Operations.

****

The **AnnotatedBioSeq** contains a collection of **SeqSymmetries** that annotate the sequence; it is the interface to retrieve annotations covered by a span within the sequence.



The **CompositeBioSeq** contains a SeqSymmetry describing the mapping of BioSeqs used in composition to the CompositeBioSeq itself. The composition representations can be sequence assembly, piecewise data loading, genotypes, chromosomal rearrangements, primer construction, reverse complement and coordinate shifting, etc.

****

**2.3 MIAME**

**Minimum Information About a Microarray Experiment** (**MIAME**) describes the minimum information required to ensure that microarray data can be easily interpreted and that results derived from their analysis can be independently verified. We used the MIAME standard as a guideline to define experimental protocols also for high throughput sequencing data, in accordance with the MIAME implementation in the **Gene Expression Omnibus** (**GEO**) repository.

A collection of gene expression data can be viewed abstractly as a table with rows representing genes, columns representing various samples and each position in the table describing the measurement of a particular gene in a particular sample: this is called a gene expression matrix. In addition to the matrix, the description of a microarray experiment must contain information about the genes whose expression has been measured and the experimental conditions under which the samples were taken. Raw data from microarray experiments are images. These images have to be quantified by an image analysis software, which identifies spots related to each element on the array and measures the ﬂuorescence intensity of each spot in each channel, together with the background intensity and a number of other quantifications, depending on the particular software. A cDNA sequence may be spotted on a microarray several times; in addition, several distinct cDNA sequences may be spotted that map to the same gene. To yield a single value for these, the corresponding measurements have to be combined. Moreover, the same biological condition can be measured in several (replicate) hybridizations, and the information from all replicates has to be summarized to derive a single gene expression data matrix. Finally, to compare microarray data from several samples, the data must be appropriately normalized.

Because microarray data have meaning only in the context of the particular biological sample analyzed and the exact conditions under which the sample was taken, a major element of the standard addresses sample annotation. For instance, different cell types may react to treatments with various chemical compounds in different ways. One must record unambiguous information about the cell types and compounds used in these experiments. MIAME standard fulfill at least two main requirements: first, the recorded information about each experiment should be sufficient to interpret the experiment and should be detailed enough to enable comparisons to similar experiments and permit replication of experiments; second, the information should be structured in a way that enables useful querying as well as automated data analysis and mining. Thus, the minimum information about a published microarray-based gene expression experiment includes a description of the following six sections:

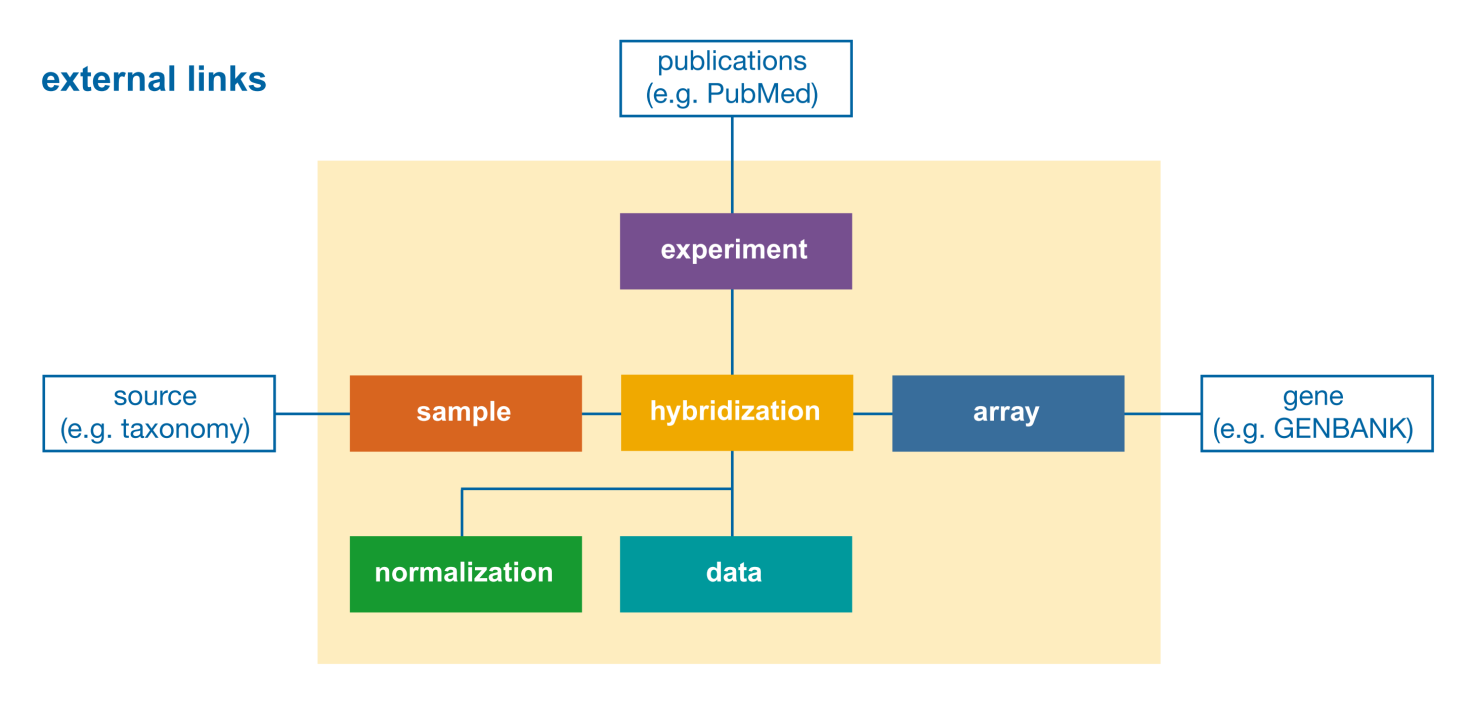
1. **Experimental design**: the set of *hybridization experiments* as a whole, which may consist of one or more hybridizations that are interrelated, and address a common biological question. Each experiment should have an author (i.e., the submitter) as well as contact information, links (i.e., URLs), citations and a single-sentence experiment title.
2. **Array design**: a systematic definition of all arrays used in the experiment, including the genes represented and their physical layout on the array.
3. **Samples**: This section describes the second partner in the hybridization reaction: the labeled nucleic acids that represent the transcripts in the sample. The MIAME *sample* concept represents the biological material (or biomaterial) for which the gene expression profile has being established. This section is divided into three parts which describe the source of the original sample (such as organism taxonomy and cell type) and any biological in vivo or in vitro treatments applied, the technical extraction of the nucleic acids, and their subsequent labeling.
4. **Hybridizations**: It defines the laboratory conditions under which the hybridizations were carried out. Other than a free-text description of the hybridization protocol, MIAME requires that a number of critical hybridization parameters are explicitly specified: choice of hybridization solution (such as salt and detergent concentrations), nature of the blocking agent, washing procedure, quantity of labeled target used, hybridization time, volume, temperature and descriptions of the hybridization instruments**.**
5. **Measurement data**: The actual experimental results. It includes three specifications:

* Original scans of the array (images);
* Microarray quantification matrices based on image analysis;
* Final gene expression matrix after normalization and consolidation from possible replicates.

1. **Normalization controls**: A typical experiment follows a *reference design* in which many samples are compared to a common reference sample so as to facilitate inferences about relative expression changes between samples. For these comparisons, the reported hybridization intensities derived from image processing must be normalized. Normalization adjusts for a number of technical variations between and within single hybridizations.

MIAME standard includes:

* The normalization strategy;
* The normalization and quality control algorithms used;
* The identity and location of the array elements serving as controls, and their type;
* Hybridization extracts preparation.

Each of these sections contains information that can be provided using controlled vocabularies, as well as fields that use free-text format.

**2.4 LIMS**

Many labs already store information about the performed experiments similarly as in MIAME/GEO. This kind of information is usually managed by a Laboratory Information Management System (LIMS). We briefly describe here the LIMS implementation at IEO-IIT. First the experimenter must compile a form similar to the following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Fill All Fields!!** | **Sample #1** | **Sample #2** | **Sample #3** | **Sample #4** | **Sample #5** | **…** |
| **UserName** | *#######* |  |  |  |  |  |
| **UserTel** | *#######* |  |  |  |  |  |
| **Institute** | *IIT* |  |  |  |  |  |
| **PI** | *###* |  |  |  |  |  |
| **CostCenter** | *IIT* |  |  |  |  |  |
| **SampleName** | *Sample ###* |  |  |  |  |  |
| **Application** | *ChIP-seq* |  |  |  |  |  |
| **Instrument** | *Illumina* |  |  |  |  |  |
| **ReadMode** | *SingleRead* |  |  |  |  |  |
| **ReadLenght** | *50* |  |  |  |  |  |
| **SequencingDepth** | *20* |  |  |  |  |  |
| **SampleType** | *IP* |  |  |  |  |  |
| **LibrarySynthesis** | *Needed* |  |  |  |  |  |
| **Organism** | *Mouse* |  |  |  |  |  |
| **Antibody** | *MYC* |  |  |  |  |  |
| **SampleDescription** | *Control* |  |  |  |  |  |
| **SampleConcentration** | *0.16* |  |  |  |  |  |
| **TotalAmount** | *6.5* |  |  |  |  |  |
| **BulkFragmentSize** | *300* |  |  |  |  |  |
| **Comments** | *………….* |  |  |  |  |  |

It is easy to find most of the information above in the form in our GDM schema, although scattered in several entities; however, the relationship of “samples” to the files representing experimental results is not expressed in the LIMS. For this reason, the IEO-IIT LIMS is backed by a MySQL database containing the following relationships:



Each ***sample*** is connected with N:N relations to a ***request***, i.e. the request for actually doing the experiment, an ***application***, i.e. the description of the specific experiment, as ChIP-seq, and finally the ***run*** of the machine.

We expect that experimental files supported by GDM will be “profiled” by taking information from the various entities in the data end experiment sub-schemas, which are functionally determined by the ProcessedData entity; every profile will then be rather similar to the LIMS information. For instance, the following is an example of GDM Profile, where some of the data are attributes from the GDM and some other <attribute, value> pairs represent the extensible parts of the GDM schema.

**Example GMD Profile**

Extraction: Params [<pvalue, 40>]

Processed: File\_URL = <http://myProcessedfile>,

Assembly\_REF = [http://www.ensamble.org/Homo\_sapiens/](http://ensamble.org/Homo_sapiens/)hs19

AnAlgo = MACS, AnParams [<pvalue, 20>]

Experiment: ExpType = ChIP-seq, ExpName = Transcription Factor Binding Sites

MachineName = Illumina 2000

Phenotype: Phenotype [<disease = myelodysplastic neoplasm>]

Individual: Individual [<sex = female>]

1. <http://www.biodas.org/> [↑](#footnote-ref-1)
2. [www.ebi.ac.uk/~aj/1.6\_draft7/documents/spec.html](http://www.ebi.ac.uk/~aj/1.6_draft7/documents/spec.html) [↑](#footnote-ref-2)
3. [**http://www.bioviz.org/igb/**](http://www.bioviz.org/igb/) [↑](#footnote-ref-3)