Visionary Paper

Data Management for Next Generation Genomic Computing

Stefano Ceri, Abdulrahman Kaitoua, Marco Masseroli, Pietro Pinoli, Francesco Venco
DEIB, Politecnico di Milano
Piazza L. da Vinci 32, 20133 Milano
first.last@polimi.it

ABSTRACT

Next-generation sequencing (NGS) has dramatically reduced the cost and time of reading the DNA. Huge investments are targeted to sequencing the DNA of large populations, and repositories of well-curated sequence data are being collected. Answers to fundamental biomedical problems are hidden in these data, e.g. how cancer arises, how driving mutations occur, how much cancer is dependent on environment. So far, the bio-informatics research community has been mostly challenged by *primary analysis* (production of sequences in the form of short DNA segments, or "reads") and *secondary analysis* (alignment of reads to a reference genome and search for specific features on the reads); yet, the most important emerging problem is the so-called *tertiary analysis*, concerned with multi-sample processing of heterogeneous information. Tertiary analysis is responsible of *sense making*, e.g., discovering how heterogeneous regions interact with each other.

This new scenario creates an opportunity for rethinking genomic computing through the lens of fundamental data management. We propose an essential data model, using few general abstractions that guarantee interoperability between existing data formats, and a new-generation query language inspired by classic relational algebra and extended with orthogonal, domain-specific abstractions for genomics. They open doors to the seamless integration of descriptive statistics and high-level data analysis (e.g., DNA region clustering and extraction of regulatory networks). In this vision, computational efficiency is achieved by using parallel computing on both clusters and public clouds; the technology is applicable to federated repositories, and can be exploited for providing integrated access to curated data, made available by large consortia, through user-friendly search services. Our most far-fetching vision is to move towards an *Internet of Genomes* exploiting data indexing and crawling.

Categories and Subject Descriptors

H.2.1 [**Logical design**]: Data models; H.2.3 [**Languages**]: Query languages; H.2.8 [**Database applications**]: Scientific databases.

Keywords

Genomic data management

1. INTRODUCTION

Modern genomics promises to answer fundamental questions for biological and clinical research, e.g., how protein-DNA interactions and DNA three-dimensional conformation affect gene activity, how cancer develops, how driving mutations occur, how much complex diseases such as cancer are dependent on personal genomic traits or environmental factors. Unprecedented efforts in genomics are made possible by **Next Generation Sequencing (NGS)**, a family of technologies that is progressively reducing the cost and time of reading the DNA. Huge amounts of sequencing data are continuously collected by a growing number of research laboratories, often organized through world-wide consortia (such as ENCODE [1], TCGA [2], the 1000 Genomes Project [3], and Epigenomic Roadmap [4]); personalized medicine based on genomic information is becoming a reality.

Several organizations are considering genomics at a global level. Global Alliance for genomics and Health[1] is a large consortium of over 200 research institutions with the goal of supporting voluntary and secure sharing of genomic and clinical data; their work on data interoperability is producing a data conversion technology[2] recently provided as an API to store, process, explore, and share DNA sequence reads, alignments, and variant calls, using Google's cloud infrastructure[3]. Parallel frameworks are used to support genomic computing, including Vertica[4] (used by Broad Institute and NY Genome Center) and SciDB[5] (used by NCBI for storing the data of the 1000 Genomes project [3]. A survey of current challenges in computational analysis of genomic big data can be found in [5]. According to many biologists, answers to crucial genomic questions are hidden within genomic data already available in these repositories, but such research questions go simply unanswered (or even unasked) due to the lack of suitable tools for genomic data management and processing.

So far, the bio-informatics research community has been mostly challenged by **primary analysis** (production of sequences in the form of short DNA segments, or "reads") and **secondary analysis** (alignment of reads to a reference genome and search for specific features on the reads, such as variants/mutations and peaks of expression); but the most important emerging problem is the so-called **tertiary analysis**, concerned with multi-sample processing, annotation and filtering of variants, and genome browser-driven exploratory analysis [6]. While secondary analysis targets *raw data* in output from NGS processors by using specialized methods, tertiary analysis targets processed data in output from secondary analysis and is responsible of *sense making*, e.g., discovering how heterogeneous regions interact with each other (see Figure 1).

[1] http://genomicsandhealth.org/

[2] http://ga4gh.org/#/api

[3] https://cloud.google.com/genomics/

[4] https://www.vertica.com/

[5] http://www.paradigm4.com/

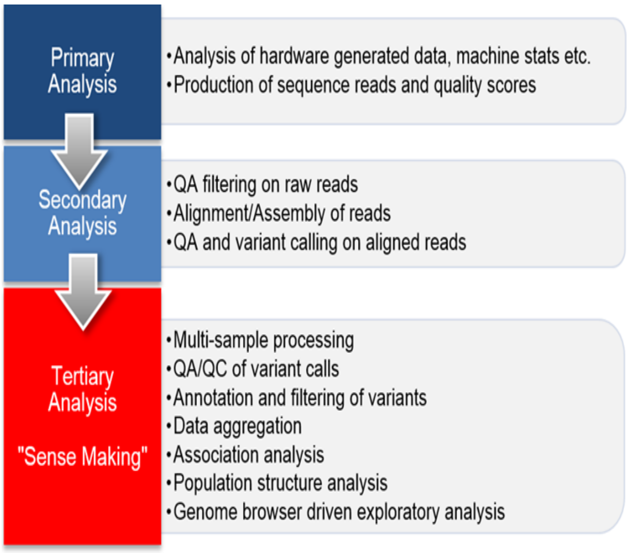

Primary Analysis
•Analysis of hardware generated data, machine stats etc.
•Production of sequence reads and quality scores
Secondary Analysis
•QA filtering on raw reads
•Alignment/Assembly of reads
•QA and variant calling on aligned reads
Tertiary Analysis
"Sense Making"
•Multi-sample processing
•QA/QC of variant calls
•Annotation and filtering of variants
•Data aggregation
•Association analysis
•Population structure analysis
•Genome browser driven exploratory analysis

Figure 1. Phases of genomic data analysis, source: http://blog.goldenhelix.com/grudy/a-hitchhiker%E2%80%99s-guide-to-next-generation-sequencing-part-2/

Tertiary processing consists of integrating DNA features; these can be specific DNA variations (e.g., a variant or mutation in a DNA position), or signals and peaks of expression (e.g., regions with higher DNA read density). Processing can also give structural properties of the DNA, e.g., break points (where the DNA is damaged) or junctions (where DNA creates loops, and then locations which are distant on the 1D string become close in the 3D space).

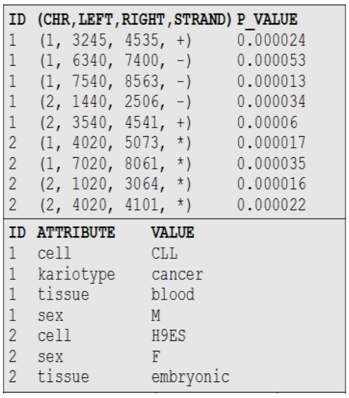
While gigantic investments are targeted to sequencing the DNA of larger and larger populations, comparably much smaller investments are directed towards a computational science for mastering tertiary analysis. Bio-informatics resources are dispersed in provisioning a huge number of tools for ad-hoc processing of genomic data, targeted to specific tasks and adapted to technology-driven formats, with little emphasis on powerful abstractions, format-independent representations, and out-of-the-box thinking and scaling. Programming data manipulation operations directly in Python or R is customary.

Another source of difficulty comes from "metadata", which describe DNA region-invariant properties of the biological sample processed by NGS, i.e., the sample cell line, tissue, preparation (antibody used), experimental conditions, and in case of human samples the race, gender, and other phenotype-related traits. This information should be stored in principled data schemes of a "LIMS" (laboratory information management system) and be compliant with standards, but biologists are very liberal in omitting most of it, even in well-cured repositories.

2. OUR CONTRIBUTION

Bio-informatics suffers its interdisciplinary nature and is considered by biologists and clinicians as a commodity that should immediately respond to their pressing needs, while it stays too far from foundational science to attract the interest of many core computer scientists. We understood that it is "mission impossible" for basic computer science to have an impact on primary and secondary analysis: algorithms are biologically driven and very specialized and efficient. Hence, we decided not to interfere with current biologists' practices, but rather to empower them with radically new data processing capabilities.

We propose a paradigm shift based on introducing a very simple data model which mediates all existing data formats, and a high-level, declarative query language which supports data extraction as well as the most standard data-driven computations required by tertiary data analysis. The **Genomic Data Model (GDM)** is based on just two entities: genomic region and metadata. Regions (upper part of Figure 2) have a normalized schema (i.e., a table of typed attributes) where the first five attributes are fixed and the next attributes are variable and reflect the "calling process" that produced them. The fixed attributes include the sample identifier and the region coordinates (the chromosome whom the region belongs to, its left and right ends, and the strand - i.e., the "+" or "–" of the two DNA strands on which the region is read, and "*" if the region is not stranded). The model can be adapted to the rare cases of regions across chromosomes. Metadata (lower part of Figure 2) are even simpler. They are arbitrary, semi-structured attribute-value pairs, extended into triples to include the sample identifier. We consider this model a paradigm shift, because a single model describes, though simple concepts, all types of processed data (peaks, signals, mutations, DNA sequences, loops, break points).



ID	(CHR,LEFT,RIGHT,STRAND)	P_VALUE
1	(1, 3245, 4535, +)	0.000024
1	(1, 6340, 7400, -)	0.000053
1	(1, 7540, 8563, -)	0.000013
1	(2, 1440, 2506, -)	0.000034
1	(2, 3540, 4541, +)	0.00006
2	(1, 4020, 5073, *)	0.000017
2	(1, 7020, 8061, *)	0.000035
2	(2, 1020, 3064, *)	0.000016
2	(2, 4020, 4101, *)	0.000022

ID	ATTRIBUTE	VALUE
1	cell	CLL
1	karyotype	cancer
1	tissue	blood
1	sex	M
2	cell	H9ES
2	sex	F
2	tissue	embryonic

Figure 2. GDM schema and instances for NGS ChIP-Seq data.

The data model is completed by a constraint: data samples can be included into a named dataset when their genomic regions have the same schema. Thus, the above figure shows the PEAKS dataset for "ChIP-Seq" data with two samples (1 and 2) whose regions fall within two chromosomes (1 and 2) and whose variable part of the schema consists of the attribute P_VALUE (each peak's statistical significance). Note that the sample ID provides a many-to-many connection between regions and metadata of the same sample; e.g., sample 1 has 5 regions and 4 metadata attributes, sample 2 has 4 regions and 3 metadata attributes; regions of the first sample are stranded (positively or negatively oriented along the DNA), while regions of the second sample are not stranded. Metadata tell us that sample 1 has karyotype "cancer" and sample 2 was taken from a "female". This example is simple, but we can associate a schema with arbitrarily complex processed data, where typed and named attributes serve the purpose of any numerical or statistical operation across compatible values. An important operation is the **schema merging**, which allows merging datasets with different schemas (the operation builds a new schema such that fixed attributes are in common and variable attributes are concatenated; in this way, we provide interoperability across heterogeneous processed data.

We also defined a query language, called **GenoMetric Query Language (GMQL)** - the name derives from its ability of computing distance-related queries along the genome, seen as a sequence of positions. GMQL is a **closed algebra over datasets**: results are expressed as new datasets derived from their operands. Thus, GMQL operations compute both regions and metadata, connected by IDs; they perform schema merging when needed. GMQL operations include classic algebraic transformations (SELECT, PROJECT, UNION, DIFFERENCE, JOIN, SORT, AGGREGATE) and domain-specific transformations (e.g., COVER deals with replicas of a same experiment; MAP refers genomic signals of experiments to user selected reference regions; GENOMETRIC JOIN selects region pairs based upon distance properties). The language brings to genomic computing the classic algebraic abstractions, rooted in Ted Codd's seminal work, and adds suitable domain-specific abstractions. Tracing provenance both of initial samples and of their processing through operations is a unique aspect of our approach; knowing why resulting regions were produced is quite relevant. In [7], we show GMQL at work in many heterogeneous biological contexts.

We give an intuition of GMQL through a simple example, consisting of three operations. We start from two datasets called ANNOTATIONS and ENCODE, the former includes samples with the reference regions from the UCSC database[6], the latter includes thousands of samples from ENCODE (in BED format); both are available at our server, with both regions and metadata. Two selections are used to produce two intermediate datasets: PROMS extracts from ANNOTATIONS a single sample with all the promoter regions of known genes; PEAKS extracts the samples of type 'ChipSeq' from ENCODE. Then, a map operation applies to the intermediate datasets PROMS and PEAKS and produces the RESULT dataset. The MAP operation, as well as all GMQL operations, implicitly iterates over all the samples of its operand datasets; it counts, for each input peak sample, all the peaks of expression over each region of PROMS, representing gene promoters. Thus, RESULT contains one output sample for each PEAK input sample, each with all the regions of PROMS; for each of such regions, it has the counter of peaks of the sample which fall within such region. This simple example shows the power of the language: with tree algebraic operations, we select reference regions and experiments and then compute aggregate properties of each experiment over each reference region, with implicit iteration over all the experiment samples.

```
PROMS = SELECT(annType == 'promoter') ANNOTATIONS;
PEAKS = SELECT(dataType == 'ChipSeq') ENCODE;
RESULT = MAP(peak_count AS COUNT) PROMS PEAKS;
```

This query above was executed over 2,423 ENCODE samples including a total of 83,899,526 peaks, which were mapped to 131,780 promoters, producing as result 29 GB of data.

3. DATA-DRIVEN GENOMIC PROBLEMS

An open problem that we are nowadays studying concerns the search for a correlation of cancer-inducing mutations and DNA string breaks with abnormal gene activity during cell replication [8], as one of the possible basic mechanisms of cancer. The assumption under consideration is that the abnormal production of DNA string breaks correlates with the presence of mutations (simply explained: mutations occur where the genome is most fragile, fragility is revealed by DNA break points); this in turn may be caused by gene dis-regulation during the process of cell replication (certain genes omit to perform a regulatory function that should prevent mutations during replication, or should fix them afterwards). In this problem, we are therefore confronted with correlating the cell replication with gene regulations; we do it in experimental conditions (exposure of cells to oncogenes), and we study how the induction of the oncogene changes both replication time and expression of other genes. The study requires genome-wide comparison of heterogeneous datasets (breakpoints, mutations, gene replication times and gene expressions under different experimental conditions), challenging both GDM and GMQL, and then calling for specific data analysis; specifically, GMQL can extract differentially dis-regulated genes, intersect them with regions where string breaks occur, and then count the mutations in various conditions.

Another open problem is concerned with the tri-dimensional layout of DNA, which is induced by the chromatin structure revealed by peaks of the CTCF transcription factor, and **understanding how CTCF loops influence gene regulation** [9]; a loop is simply a binding of the DNA, so that two DNA regions which are far away from a 1D perspective become very close from a 3D perspective. In Figure 3, within yellow (thin) rectangles we see three signals which identify three non-coding regions of the genome, called enhancers, and within a black (thick) rectangle we see signals which identify the promoter of the gene Fbln2. They are enclosed within regions which represent short CTCF loops, and the assumption to be tested is whether there is a direct relationship between active enhancers and active genes (where activity is revealed by experiments) when enhancers and promoters are enclosed within CTCF loops (as this spatial condition may favor the enhancer-to-gene relationship); *determining the relationships of genes with enhancers is a fundamental aspect of epigenetics.* Such question corresponds to searching a pattern within the whole genome; GMQL can be used to extract candidate gene-enhancer pairs by suitable intersections of the signals in Figure 3 - i.e., CTCF regions, the regions of the three methylation experiments (H3K27AC, H3K4me1, H3K4me3), and gene promoter regions (from RefSeq).

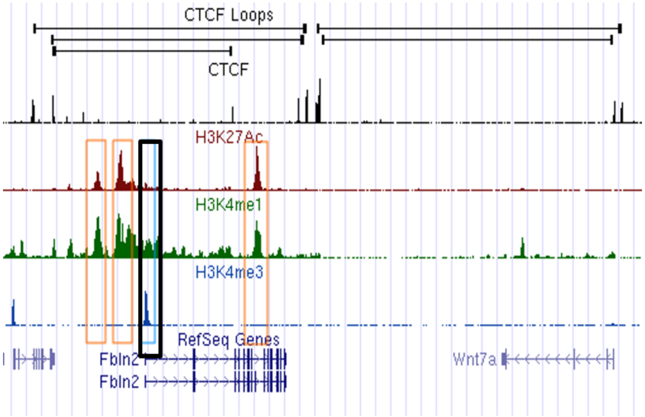


Figure 3. Interaction between CTCF loops and gene regulation by enhancers.

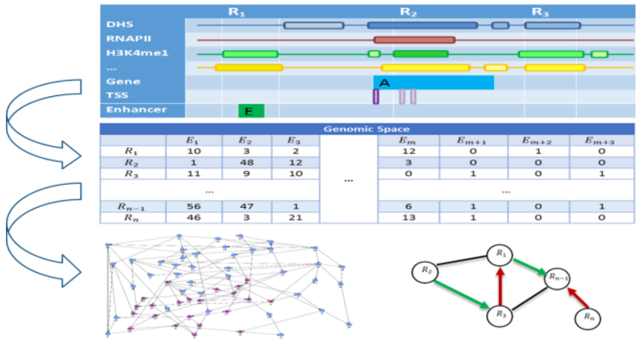
[6] http://genome.ucsc.edu/cgi-bin/ hgTables ?hgsid=445319346_kHaTO493uLRZhjuqCvaKTaFt7HL3

4. VISION

GDM and GMQL open new scenarios in approaching tertiary analysis of genomic data. We next discuss them.

4.1 Data Analysis

Data analysis methods which are most useful for genomic computing can be bridged to the high-level language, with a bottom-up, problem driven approach. In particular, query results can be expressed in the form of interaction networks between genomic regions. In biology, many genes are involved in complex regulatory processes; for us, genes are just DNA regions of a specific sample (they are "known annotations of the genome") and thus we can MAP (using the domain-specific operation shown in the example of Section 2) arbitrary experiments to genes. MAP is the first transformation in Figure 4, which computes aggregates over those regions of regions of experiments that intersect with genes (represented by regions R1, R2, R3). In general, every map operation produces what we call a **genome space**, i.e., a tabular space of regions vs. experiments (in the middle of Figure 4), which is the starting point for data analysis (including advanced data mining and computational intelligence). Such table can be also interpreted as an adjacency matrix representing a network, where regions are nodes and arcs have a weight obtained by further aggregating properties across experiments; thus, the second transformation in Figure 4 yields to a **gene network**, producing as well the strength of gene-to-gene interactions. The interpretation of genome spaces in the form of networks is particularly important in genomics, as regulatory gene activities typically depend on multiple interacting genes.



	E_1	E_2	E_3		E_m	E_{m+1}	E_{m+2}	E_{m+3}
R_1	10	3	2	...	12	0	1	0
R_2	1	48	12		3	0	0	0
R_3	11	9	10		0	1	0	1
	...				...			
R_{n-1}	56	47	1		6	1	0	1
R_n	46	3	21		13	1	0	0

Figure 4. Interpretation of GMQL "map" query as a genome space, and further transformation of the genome space into a gene network.

Several data mining and computational intelligence approaches, including advanced *latent semantic analysis* and *topic modelling*, can be applied to evaluate relationships among genomic data, and between them and biological or clinical features of experimental samples expressed in their metadata, i.e., for *genotype-phenotype* correlation analysis.

4.2 Distributed Processing and Cloud Computing

With the growth of NGS experiments (whose cost is expected to drop to about 100 Euro in less than a decade, https://www.genome.gov/sequencingcosts/), we will see a deluge of NGS data. Although processed data are "smaller" than raw data (0.3 TB per full genome sample), we are still talking of samples with tens of thousands or even millions of regions. Genomic repositories store thousands of full genome samples (i.e., 4,660 samples in [1], 5,400 samples in [2], and 2,500 samples in [3]). Our simple query in Section 3 produced 83 million regions, and simple queries over genes may produce genome spaces of 10K genes and 100M relationships between them, whose analysis requires using large-scale network management packages. Moreover, NGS is increasingly used for massive testing on restricted, pathology-specific mutation panels, so as to accelerate the use for diagnostics and for clinics. We are clearly facing one of the **most important "big data" problems for mankind**.

We are currently working towards a new GMQL release, that will be available in 2016, and will support two parallel implementations, respectively using Flink[7] and Spark[8], two emerging data frameworks. In our architecture, the two implementations differ only in the encoding of about twenty GMQL language components, while the compiler, logical optimizer, and APIs/UIs are independent from the adoption of either framework. In a recent paper [10] we present an early comparison of Flink and Spark at work on three genomic queries inspired by GMQL. Several tools were developed within the Hadoop framework for primary and secondary analysis, including BioPig [11], SeqPig [12] and SparkSeq [13]. Our preliminary work shows open source frameworks are effective computing systems also for tertiary data analysis; we foresee a growth of systems for genomic based upon parallel computing frameworks.

So far, our focus on tertiary data analysis is shared just by Paradigm4, a startup company founded by the Turing award Mike Stonebraker, whose products include genomic add-ons to SciDB, a vector-based data management system for scientific applications. They provide access to data from TCGA and 1000 Genomes Project, and they advocate the use of specialized databases for scientific computing rather than cloud computing – indeed, we find in [6] several arguments against the use of Spark. We expect that the alternative between open frameworks and specialized systems will shape the evolution of genomic data management in the forthcoming years.

4.3 Integrated Access to Repositories

Very large-scale sequencing projects are emerging; as of today, the most relevant ones include:

- The **Encyclopedia of DNA elements (ENCODE)** [1], the most general and relevant world-wide repository for basic biology research. It provides public access to more than 4,000 experimental datasets, including the just released data from its Phase 3, which comprise hundreds of epigenetic experiments of processed data in human and mouse;
- The **Cancer Genome Atlas (TCGA)** [2], a full-scale effort to explore the entire spectrum of genomic changes involved in human cancer;
- The **1000 Genomes Project** [3], aiming at establishing an extensive catalogue of human genomic variations from 26 different populations around the globe;
- The **Epigenomic Roadmap Project** [4], a repository of "normal" (not involved in diseases) human epigenomic data from NGS processing of stem cells and primary ex vivo tissues.

[7] https://**flink**.apache.org/

[8] https://**spark**.apache.org/

Data collected in these projects are open and public; all the Consortia release both raw and processed data, but biologists in nearly all cases trust the processing, which is of high-quality and well controlled and explained. All consortia provide portals for data access; some systems already provide integrated access to some of them (e.g., [14]; see also http://www.paradigm4.com/). The use of a high-level model and language, such as GDM and GMQL, is the ideal setting for provisioning next generation services over data collected and integrated from these and other repositories, improving over the current state-of-the-art in four directions:

- All the processed datasets available in the above data sources will be provided of compatible metadata;
- It will be possible to choose among a set of custom queries, representing the typical/most needed requests;
- It will be possible to provide user input samples to the services, whose privacy will be protected;
- Deferred result retrieval will be possible, through limited amount of staging at the sites hosting the services.

A simple protocol will facilitate input and output file transmissions and it will also be possible to visualize results on genome browsers or to selectively retrieve regions or metadata. Users will be enabled to write personalized queries, whose privacy will be protected. The main challenges in this vision include two new research objectives: the mediation of ontological knowledge and the statistical description of custom queries.

- Ontological reasoning will be required in order to establish the appropriate **conceptual relationships between the metadata** which are present at the various sources. The best option is to use the global ontology provided by the Unified Medical Language System (UMLS) [15], which collects and integrates well-established biomedical ontologies. Our initial solution, presented in [16], consists in semantically annotating the metadata of each repository's datasets by means of UMLS, and completing the information by performing the semantic closure [17] of such annotations. Then, a suitable UI would allow users to search for relevant experiments through keyword-based or free text queries.
- **Custom queries** will need to be augmented with suitable mechanisms for reasoning about data; such services could imitate the *Great* service developed by Gill Bejerano's group at Stanford [18], which includes powerful statistics to indicate the significance of query results.

4.4 Federated Query Processing and Protocols

The availability of a core data model as a data interoperability solution and of a high-level data processing language is a strong prerequisite for defining data exchange protocols. We expect that each data repository will be the owner of the data that are locally produced, and that nodes of cooperating organizations will be connected to form a federated database. In such systems, queries move from a requesting node to a remote node, are locally executed, and results are communicated back to the requesting node; this paradigm allows for distributing the processing to data, transferring only query results which are usually small in size.

Supporting a high-level query interface to a server is already making one big step forward, which is similar to the gigantic step made by SQL in the context of client-server architectures (which dates a couple of decades). Indeed, once a system supports an API for submitting GMQL queries, these have the following properties: they are short texts and produce short answers. This comes from the nature of problems: the more they are biologically inspired, the more they produce results which are both short and ranked, and these will eventually be transmitted along any GMQL API; in contrast, most of today's implementations requires first a full data transmission and then to evaluate server-side imperative programs. This scenario opens up to the design of simple interaction protocols, typically for:

- **Requesting information about remote datasets**, facilitated by the availability of metadata (for locating data of interest) and of their region schemas (for formalizing queries).
- **Transmitting a query in high-level format and obtain data about its compilation**, not only limited to correctness, but including also estimates of the data sizes of results.
- **Launching query execution and then controlling the transmission of results**, so as to be in control of staging resources and of communication load.

4.5 Search Methods and Internet of Genomes

After having provided access to integrated sources of sequence data, we come to the question of how such knowledge can be searched. The problem can be approached progressively, starting first with opening search services over the integrated repositories. There are two intertwined problems:

- **Metadata search.** Search methods should locate relevant samples within very large bodies, using classical measures of precision and recall; keyword-based search or free text querying should be supported.
- **Feature-based region search.** Best-matching regions with user-specified features should be provided. For some regions (e.g., known genes) it is possible to define a priori the typical features, store them as attributes, and then use indexing; but in general features should be computed. We envision general search mechanisms where the user selects interesting regions, then provides information about the features of interest, then those features are computed, and finally regions are ordered based on their computed features and presented to the user. So, search and feature evaluation have to intertwine in a clever way.

The most ambitious and challenging vision is **building a search system upon an Internet of genomes**. The prerequisite to this vision is of course not in today's reach, and requires all research centers to agree on a deployment technology playing the role of HTML and HTTP for the Web. However, biologists are forced to publish the data which go together with their experiments: it is already in their practice to provide a link to a download site where experimental data should be available for downloading by reviewers. In such context, it is possible to envision the definition of a simple protocol for data publishing, prescribing how to publish a link to genomic data in their native format with suitable metadata; the protocol should offer the possibility of making such link public, i.e., visible within a host open to the visits of crawlers. With such infrastructure, a third party hosting a search service could periodically launch the crawlers, and these would download the metadata and links from the host; the search service could also download datasets from the hosts by using those links, with an agreed, non-intrusive protocol. The search service would then have all the required information for indexing all the metadata and for storing some of the samples within a large repository, possibly pre-computing some features of their regions. Such search system could accept search queries and produce result snippets, with an indication of the presence of each dataset in the repository. In any case, users of the search system would be able to locate genomic data available at another host (a research or clinical center) and could download them asynchronously.

5. CONCLUSIONS

The progress in DNA and RNA sequencing technology has been so far coupled with huge computational efforts in primary and secondary genomic data management, consisting of producing “raw” data, aligning them to the reference genomes, and calling for specific features such as expression peaks and mutations. However, a new pressing need is emerging: making sense of data produced by these methods, in the so-called tertiary analysis. This need requires a substantial change of the dominating approach to bio-informatics. While primary and secondary analyses produce data formats which are typically intricate and incompatible, tertiary analysis must worry about their interoperability and ease of use. Tertiary analysis calls for raising the level of abstractions of models, languages and tools for genomics, going towards a broader vision where biologists and clinicians can observe the huge and complex body of genomic knowledge at a much higher level, using simple interfaces similar to search queries which have become widely available in the Internet.

In this paper, we have shown that a change of paradigm is possible, by means of a new data model and query language; we have then shown the biological applications that have become feasible thanks to this approach, and examined the relevant advantages that this approach may bring in the contexts of data analysis, distributed processing, integrated repository access, federated data management, and search of genomic data over the Internet. The corresponding scenario traces a five-to-ten year research trajectory.

6. ACKNOWLEDGEMENTS

We acknowledge the essential contributions of Pier Giuseppe Pelicci, Ivan Dellino, Lucilla Luzi, Laura Riva, Mattia Pelizzola (IEO), of Heiko Muller (IIT) and of Michele Bertoni, Gianpaolo Cugola, Vahid Yalili, Matteo Matteucci, Fernando Paluzzi (PoliMI) in developing our approach to genomics. This research is supported by the PRIN Project GenData 2020 and by a grant from Amazon Web Services.

7. REFERENCES

[1] ENCODE Project Consortium. 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489(7414), 57-74.

[2] Weinstein, J. N., et al. 2013. The Cancer Genome Atlas pan-cancer analysis project. *Nat. Genet*. 45(10), 1113-1120.

[3] 1000 Genomes Project Consortium. 2010. A map of human genome variation from population-scale sequencing, *Nature* 467, 1061-1073.

[4] Romanoski, C. E., et al. 2015. Epigenomics: Roadmap for regulation. *Nature* 518, 314-316.

[5] Qin, Y., et al. 2015. The current status and challenges in computational analysis of genomic big data. *Big Data Research* 2(1), 12-18.

[6] Accelerating bioinformatics research with new software for big data knowledge. White paper retrieved from: http://www.paradigm4.com/, April 2015.

[7] Masseroli, M., Pinoli, P., Venco, F., Kaitoua, A., Jalili, V., Paluzzi, F., Muller, H., Ceri, S. 2015. GenoMetric Query Language: A novel approach to large-scale genomic data management. *Bioinformatics* 12(4), 837-843.

[8] Dellino, G. I., et al. 2013. Genome-wide mapping of human DNA-replication origins: Levels of transcription at ORC1 sites regulate origin selection and replication timing. *Genome Res*. 23(1), 1-11.

[9] Handoko, L., et al. 2011. CTCF-mediated functional chromatine interactome in pluripotent cells. *Nat. Genet*. 43(7), 630-638.

[10] Bertoni, M., Ceri, S., Kaitoua, A., Pinoli, P. 2015. Evaluating cloud frameworks on genomic applications. *IEEE Conference on Big Data Management*, Santa Clara, CA.

[11] Nordberg, H., et al. 2013. BioPig:a Hadoop-based analytic toolkit for large-scale sequence data. *Bioinformatics* 29(23), 3014-3019.

[12] Schumacher, A., et al. 2014. SeqPig: simple and scalable scripting for large sequencing data sets in Hadoop. *Bioinformatics* 30(1), 119-120.

[13] Weiwiorka, M. S., et al. 2014. SparkSeq: Fast, scalable and cloud-ready tool for the interactive genomic data analysis with nucleotide precision. *Bioinformatics* 30(18), 2652-2653.

[14] Nielsen, C. B., et al. 2012. Spark: A navigational paradigm for genomic data exploration. *Genome Res*. 22(11), 2262-2269.

[15] Bodenreider, O. 2004. The Unified Medical Language System (UMLS): Integrating biomedical terminology. *Nucleic Acids Res*. 32(Database issue), D267-D270.

[16] Fernandez, J. D., Lenzerini, M., Ceri, S., Masseroli, M., Venco, F. 2016. Ontology-based search of genomic metadata. *IEEE/ACM Trans. Comput. Biol. Bioinform*. (in press).

[17] Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M. 2010. The combined approach to query answering in *DL-Lite*. *Int. Conf. Knowledge Representation and Reasoning*, 247-257.

[18] McLean, C. Y., et al. 2010. GREAT improves functional interpretation of cis-regulatory regions. *Nat. Biotechnol*. 28(5), 495-501.