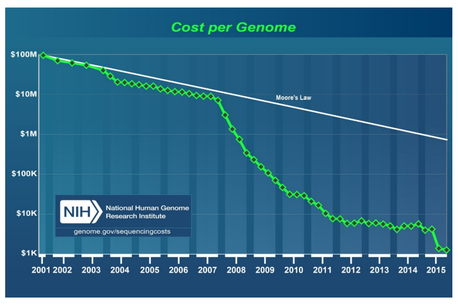
ERC WEEK
ERC's 10th anniversary
Data-Driven Genomic Computing (GeCo)

Stefano Ceri

DEIB (Dept. Of Electronics, Information and Bio-Engineering)



CONTEXT: Next Generation Sequencing is progressively reducing the cost and time of reading the DNA of each individual. Huge amounts of sequence data are continuously collected by a growing number of research laboratories, often organized through world-wide consortia (such as ENCODE, TCGA, the 1000 Genomes Project); personalized and precision medicine based on individual genomic information is becoming a reality.

GECO OBJECTIVE. Genomic Computing (**GeCo**) aims at developing data-driven basic science for the management of sequence data, based on a simple driving principle: ***data should express high-level properties of DNA regions and samples, high-level data management languages should express biological questions with simple, powerful, orthogonal abstractions***.

- **FireCloud (Broad Inst.)**
- **Paradigm4 (Spinoff)**
- **GMQL/Geco (PoliMi)**
- **DeepBlue (Blueprint)**

PRIMARY ANALYSIS

SECONDARY ANALYSIS

TERTIARY ANALYSIS

SOFTWARE FOR GENOMICS. **Primary data analysis** for NGS technology produce ***raw data***, i.e., short reads of DNA or RNA. **Secondary data analysis** produces *aligned sequences* (to the reference genome) and then extracts their *genomic features* (e.g., data about genome mutations or gene expression), associated with DNA regions. GeCo is focused on **tertiary data analysis**, dealing with the *integration of heterogeneous features* for discovering interesting regions or properties of the genome associated with experimental conditions (e.g. normal vs tumor cells).

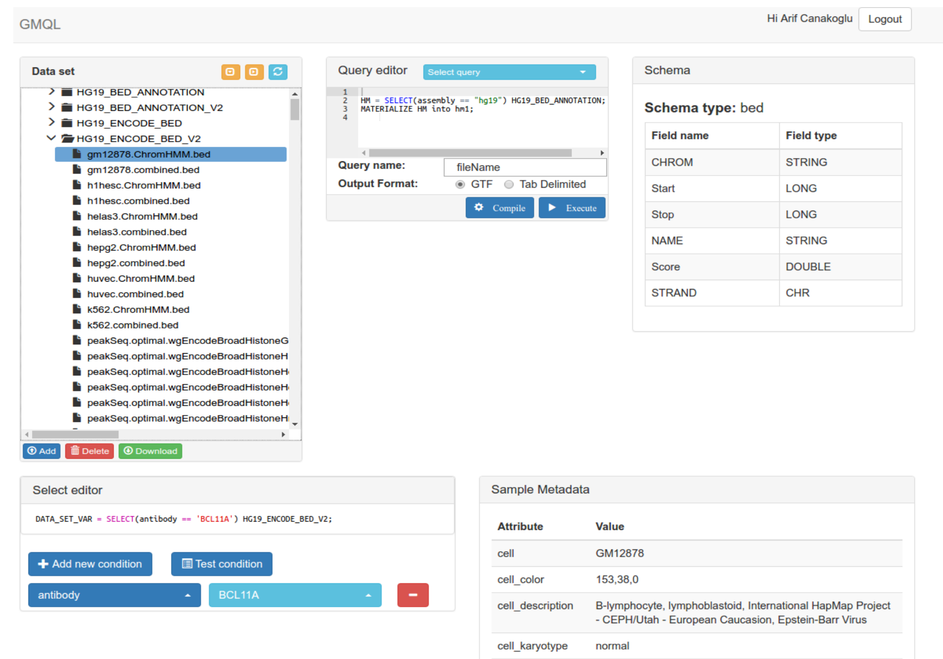
GECO PRELIMINARY RESULTS. Development of Genomic Data Model (GDM) and GenoMetric Query Language (GMQL) for tertiary data analysis.

- **GDM** describes arbitrary region-based genomic features with their metadata
- **GMQL** adds to relational operations some domain-specific operators for region calculus.

GMQL is implemented on cloud computing using *Spark, Flink* and *SciDB*, hosted at CINECA: http://www.bioinformatics.deib.polimi.it/GMQL/interfaces/.

The prototype integrates a repository of processed data from Encode, TCGA, Epigenomic Roadmap. → see a screenshot of GMQL prototype.

GeCo research is published on several *international journals in bioinformatics and computer science*: BioInformatics, Methods, BMC-Bioinformatics, Information Sciences, IEEE-TC, IEEE-TKDE, IEEE-TS, IEEE-TCBB.



SHORT-TERM GOALS

METADATA TRACING

Develop methods and tools supporting users in **explaining query results**. Determining data lineage (or provenance).

PATTERN-BASED REGION EXTRACTION

Define complex patterns of genomic features enabling the formulation of **similarity queries** (e.g., use of distal patterns or notions of similar/dense/sparse genomic regions).

DESCRIPTIVE STATISTICS

Provide automatic addition of descriptive statistics to query results; integrate **classic data science tests** (e.g. significance or regression) within the query capabilities.

MID-TERM GOALS

INTERACTION NETWORKS

Provide automatic translation of query results as interaction networks, and then use powerful **data analysis methods**, e.g., based upon deep learning.

INTEGRATED REPOSITORY

Produce an **integrated repository** with semantically well-defined and compatible metadata, by integrating GDM with ENCODE, TCGA, 1000 Genomes, Roadmap Epigenomics and many other sources.

WEB SERVICES

Use GMQL for building **public web services,** supporting statistics to indicate the significance of query results

LONG-TERM GOALS

INTERNET OF GENOMES

Use GMQL as the basis for simple interaction protocols for:

- **Requesting information** about remote datasets
- **Sending a query** and get data about its compilation and result size estimates
- **Launching execution** and then controlling the staging of resources and the communication load

METADATA AND FEATURE-BASED SEARCH

Develop **semantic indexing and searching**, supporting keyword-based search with semantic query expansion (matching terms to ontologies, e.g., OBO, UMLS). Provide results in ranking order.

Trace query histories and build **recommending systems**.