DBGen

BioDec's DataBase of Genomics data

August 2026

BIODEC

Evolving ICT Infrastructures

Executive Summary

The problem

The management of omics data has some very peculiar characteristics, which make the problem not trivially solvable with traditional IT systems (*file systems*, relational databases, etcetera). The key features is that multiple complex aspects must be managed together, but they have conflicting requirements.

Source heterogeneity Data may originate from external sequencing services: this implies a variability in the data are accessed (it does not necessarily occur via the Internet; physical media such as portable *hard disks* might be used). Other kind of data comes from public datasets as, for example, gnomAD, CADD, ClinVAr, etcetera. All of these kind of data are needed to execute the analysis pipelines.

Data volume Data of *omics* type are often very large (*i.e.* several GB, even tens or hundreds): it is extremely easy to have dedicated file systems that quickly exceed tens of TB of storage.

Computational requirements To analyze omics data often requires substantial computational resources and, in particular, systems capable of handling a highly efficient *I/O* workload, otherwise the analysis execution will slow down: unfortunately, the requirement for high *performance* does not pair well with that of having ample *storage* — indeed, they are conflicting requirements.

Access granularity and *privacy* Last but not least, in case of *human* omics data, there are privacy requirements to comply with, which complicates the access rules, as these must ensure the principle of *privacy-first* and the audit of the data management system.

When other standard aspects of managing a production IT system are added to the above issues (such as the need for *backups* and solutions for *business continuity*), it becomes immediately clear that the problem cannot be solved with a simple architecture (*i.e.* using a standard file system with its access mechanisms).

DBGen

This document describes BioDec's **DataBase of Genomics data** (*DBGen*) whose principal use cases are:

1. consolidation of **datasets** of public-domain omics data (gnomAD, CADD, ClinVar, etcetera) and proprietary research data;
2. automation of data extraction for the execution of genomics pipelines;
3. data analysis from multiple sources that are usually stored across different files — an activity that is further complicated due to the fact that data is scattered among many places.

Data types

Pipelines analysis begins with the sequence data, typically in compressed FASTA/FASTQ format: additional information from heterogeneous sources is then associated with these data by the analysis algorithms. The sources managed with DBGen are primarily of three kinds.

Metadata They may relate to projects, patients, and samples; they are typically provided as XLSX or CSV tables and constitute internal data sources of the client.

Variant calls These are performed on sample sequences and take the form of one or more VCF files processed by VEP (or another variant extractor program).

Variant annotations In addition to more common annotations, such as the *prediction score*, population frequencies, etcetera, there is a plethora of heterogeneous sources that can be provided as CSV, TSV, gff, and other formats.

Database types

DBGen will provide the following databases, accessible via a single API service.

- a database for the metadata;
- a database for the variant calls;
- a variable number of databases, where each one is dedicated to map a specific dataset of annotation sources.

Given the volume of annotations considered and the need to use different versions of the same dataset, **information will not be stored within a database**, but it will nonetheless be accessible as if it were.

The DBGen solution is based on a native extension of the PostgreSQL database server, namely the *Foreign Data Wrapper* (FDW). These integration[1] will enable access to heterogeneous annotation sources[2], allowing them to be native queried from PostgreSQL and to be linked to all the other data of the database.

Access via SQL without consolidation within a database

Note that the use of FDWs enables to **not consolidate annotation data** within PostgreSQL, but makes it accessible as if it were stored on a database table (*i.e.* through SQL command invocation).

[1] Whose implementation depends on the type of omics dataset.
[2] Which are generally distributed via tabular text files.

Avoiding physical consolidation of data on a server that implements transactional mechanisms prevents the management of massive databases — on the order of tens of terabytes — which entail equally massive management costs and often severe usage limitations precisely due to their size.

Data life cycle

An important distinction exists among the types of data managed by genomics databases, and an effective way to classify them is to assess whether their life cycle features a **slow** or **fast** mutation rate.

Slow Data subject to a life cycle with a slow change rate are those that, once created, do not change over time, or rarely change, or change *in toto*, i.e., in their entirety: a typical example of this type of data is dataset annotations.

Slow data: do not need a database at all, but it is better used if accessed through a SQL interface

- These data do not require a true database server, as static content offers no benefit from features such as transactions.
- On the opposite, since dataset are updated as a whole, it is important to have a way of distinguishing among different versions of the datasets.
- These data require tools to manage the life-cycle of the entire dataset: preparation, integration, decommissioning and so far.

Fast These data may change frequently, meaning that they grow over time, as new *records* are added with each new analysis.

Fast data: naturally advantaged by being managed with a database and accessed through a SQL interface

- These data benefit from a database server, as the transactional aspects guarantee the *ACID* properties (Atomicity, Consistency, Integrity, and Duration).
- These are data that (typically) lack a versioning concept: if a data property is updated, it is because the previous value is to be considered no longer valid.
- The frequency of changes requires targeted management tools (for the so-called *CRUD* operations — Create, Read, Update, Delete) as well as interactive features.
- Tools may be based interchangeably on web interfaces, CLI, or both options.

DBGen **annotations** are subject to a **slow** mutation cycle, while DBGen **metadata** and **variant calls** are data characterized by a **fast** mutation cycle.

Advantages

The main advantage of using DBGen is that from the users' perspective all databases are accessed as a single object, via a specialized API service. Consolidating data into a single system will therefore allow:

1. the execution of the pipelines that are already in use;

2. easily access a larger amount of annotation information;
3. filter and group annotations in ways that are currently impossible or that are very expensive to build and maintain;
4. develop new statistical analysis applications that leverage the new opportunities provided.

Glossary

Annotation is the operation of adding information to the variant.

ACID *Atomicity, Consistency, Integrity, Duration* are the properties of a transactional database.

API *Application Programming Interface*, the programming interface of an application.

BCL is the file format that contains the *Binary Base Call* and is generated by sequencing systems.

CRUD *Create, Read, Update, Delete*, are the fundamental operations on data.

DPIA *Data Processing Impact Assessment* is the document required by Article 35 of the GDPR.

FASTQ is a text format, now the *de facto* standard, for managing biosequences and their associated quality metrics (technically called *quality scores*)[3].

FDW *Foreign Data Wrapper* is an integration technology specific to the PostgreSQL database server.

Filtering is the selection of a group of variants based on certain criteria, also relying on annotations.

GDPR *General Data Protection Regulation* Also known as the European General Regulation 2016/679.

HPO *Human Phenotype Ontology* is an ontology of human phenotypes.

Object storage is a data management approach that persists them as objects (also called *blobs*) rather than as a file hierarchy within a file system.

Pipeline is the term used to designate a series of programs that, often chained together, perform a bioinformatics analysis. The output of a pipeline may be of interest in itself, or serve as the *input* for a subsequent pipeline (in this latter case, the term *sub-pipeline* is also used to refer to the components of a broader pipeline that aggregate them). Pipelines are the computational components of a computational analysis platform.

[3]Cock, P. J. A.; Fields, C. J.; Goto, N.; Heuer, M. L.; Rice, P. M. (2009). . Nucleic Acids Research. 38 (6): 1767–1771. doi:10.1093/nar/gkp1137. PMC 2847217. PMID 20015970.

S3 API is the HTTP(S) access protocol for the corresponding object storage service of *Amazon Web Services*[4]. Numerous alternative implementations of the protocol exist to avoid dependency on a single vendor; for example, it is recommended to use the Minio software[5].

VCF *Variant Call Format* denotes the format for representing variations (variants) in a gene sequence. It is also a text-based format, its specification being public[6], and it has long been the reference for the majority of bioinformatics programs.

VEP *Variant Effect Predictor* is a tool for analyzing the effect that variants (*SNP*, *CNV*, *indel* or structural variations) have on a specific gene, sequence, protein, transcript, or transcription factor. One of the most well-known implementations is that of the ENSEMBL project[7].

[4] https://docs.aws.amazon.com/AmazonS3/latest/API/Type_API_Reference.html
[5] https://min.io/
[6] https://samtools.github.io/hts-specs/VCFv4.3.pdf
[7] https://github.com/Ensembl/ensembl-vep

Contacts

Company BioDec s.r.l.

Address Via Calzavecchio 20/2, Casalecchio di Reno (BO)

VAT number IT02327271207

Phone +39 051 0548263

Contact person Michele Finelli (m@biodec.com)