Main Features

In this section you will find a summary of the main features of OpenCGA.

Table of Contents:

Metadata Catalog and Security

OpenCGA Catalog is one of the most important components. Catalog implements the data models, allow custom annotations, implement permissions, ... An audit system has also been implemented.

Catalog Data Models and Annotations

- **Rich data models** implemented for studies, files, samples, individuals, families, ...
- Advanced **free data model** implemented for storing **custom annotations** such as stats or clinical data from patients. Users can define **confidential annotations** only visible for authorised users.

Catalog Database

- Catalog database has been implemented using MongoDB to provide a **high-performance** and **s calable query engine**.
- Catalog can use Solr as a secondary index to calculate **complex annotations** and **stats**.

Authentication and Permissions

- OpenCGA comes with a built-in **authentication** system. Other systems are also supported such as LDAP or Microsoft Azure AD (*under development*). Authentication **tokens** use **JWT** standard which facilitate the creation of federated systems.
- Advanced and efficient resource **permission** system implemented in Catalog. You can define different permissions such as VIEW, WRITE or DELETE at study level or at any specific document. This allow to **share** data with other users. More information at Sharing and Permissions.

Alignment Storage

OpenCGA can manage alignment data. BAM files can be indexed and coverage calculated.

Fetching alignments

- **Query indexed BAM files**, allowed filters include by region, mapping quality, number of mismatches, properly paired, ...
- GA4GH data model used for alignments
- **Google gRPC** is used as an alternative to REST (JSON) to improve performance.

Coverage

- **Coverage** can be calculated and stored in a *BigWig* file.
- **Coverage queries** at any window size or zoom.

Variant Storage

OpenCGA provides a framework for implementing *big data* variant storage engines which support: real-time queries, interactive complex data aggregations, full-text search, variant analysis, ... The framework takes care of several common operations such as variant normalisation, sample genotype aggregation, variant stats calculation, variant annotation, secondary indexing or in-memory cache. Two different engines are implemented using NoSQL databases: MongoDB and HBase. A secondary index using Solr is nicely integrated with the two implementations. By implementing variant storage engines with NoSQL databases we ensure a fast response time and high concurrent queries.

Data Management

- Advanced **variant normalisation** implemented supporting multi-allelic split or left-alignment of INDELs among others.
- High quality **sample genotype aggregation** supporting multi-allelic variants, overlapping SNV-INDEL or structural variants. HBase storage engine can aggregate tens of thousands of samples efficiently. Current design and implementation should scale to hundreds of thousands of samples.
- **Dynamic** variant storage, you can add or remove samples dynamically from the variant storage efficiently

- Rich and efficient **variant data model** implemented. Variant data model support different studies, file information, samples information and a rich variant annotation. **Sample genotypes** are **efficiently** stored to scale to hundreds of thousands of genotypes, this allows to optimise analysis by minimising the disk usage and memory consumption.
- **Structural variants** are fully supported incliuding SNV, INDEL, insertion, deletions, CNV, ...
- Multi-cohort **variant stats** supported. Users can define different cohorts (group of samples) and precompute and index their variant stats, this allows a real-time queries or aggregations. A default cohort called ***all*** is managed automatically.
- CellBase high-performance **variant annotation** tool is integrated providing rich variant annotations which are stored and indexed, this allows a real-time queries or aggregations. Variant annotation data is returned with the variants since it is part of the data model. Multiple variant annotation can be stored and fetched.
- Custom **variant scores** from external analysis tools such as GWAS association can be loaded, indexed and queried by.
- **Export** variant data in different formats such as VCF or Parquet. You can filter which variants and samples are exported.

Query Engine

- OpenCGA implements a **very sophisticated query engine** supporting the combination of more than 25 filters: region, genes, type, file attributes, sample genotypes, consequence types, population frequencies, biotype, conservation scores, variant and gene clinical traits, mode of inheritance, disease panels, ... **Full-text search** is also implemented.
- Other **query options** supported such as *include, exclude, limit, skip, count, ...*
- Some **basic analysis implemented** such as compound heterozygous, *de novo* variants, sex imputation, unique variant saturation, ...
- Variant query engine supports filtering by **sample clinical data** thanks to the integration with *Cat alog.*
- MongoDB or HBase are fully integrated with **Solr secondary indexes** to provide a real-time query engine for all queries and use cases.

Aggregation and Stats

- Solr integration allows the execution of **complex aggregations** (faceted search) interactively. Nested and range aggregations are supported. For instance, you can aggregate variants by chromosome and type over 46 million variants in just 2 seconds: http://bioinfo.hpc.cam.ac.uk/hgva/webservices/rest/v1/analysis/variant/stats?timeout=60000&study=reference_grch37%3AUK10K&fields=chromosome%3E%3Etype
- **Variant query filters** – for filtering variants – and **aggregation** analysis can be combined to calculate the aggregation of any variant query result.
- **Aggregation stats** such as average, median, percentile, min, max, ... are also supported

Big Data Analysis

- Variants can be exported to **parquet** file which is an efficient columnar file format. This parquet file can be used by Hive or Spark big data technologies.
- Some **complex analysis** such as IBS are implemented using a custom Spark library to extend the number of uses cases supported. Note that this analysis can take some time and Spark is not a highly concurrent technology, therefore this analysis are queued by OpenCGA.
- Variant data model store genotypes efficiently ensuring we can execute analysis with tens of thousands of samples.

Performance and scalability

- HBase storage engine have been implemented to provide **real-time queries** and **interactive aggregations** *(faceted)* even with tens of thousands of whole genomes.
- **Google gRPC** is used as an alternative to REST (JSON) to improve performance.
- Some benchmarks with more than 11,000 whole genomes accounting for 25TB show that we can load more than 2,000 files a day and execute most queries in less than 1-2 seconds in a small Hadoop cluster of 20 nodes.
- You can go to HGVA to test OpenCGA query engine performance. HGVA uses OpenCGA and IVA and load about 700 million unique variants from different human studies.

Clinical Analysis

OpenCGA aims to provide a full solution for Clinical Genomics analysis, this covers patient clinical data, interpretation algorithms and a pathogenic variant database.

Clinical Data

- Catalog can **store and index any clinical data model** for samples, individuals or families. Models are defined by users.
- User can configure the **permission** and **visibility** of clinical data using *Catalog* permissions.

Clinical Interpretation Analysis

- Open a **patient case** study by creating a *clinical analysis*, this contains all the patient and family data from Catalog at that moment, the phenotype to be analysed or the files among other information. A rich **interpretation data model** has also been modelled – combining GEL and other data models – to capture all the relevant information from the interpretation.
- Complete **disease panel** management implemented: create, update and delete disease panels. You can also import them automatically from **PanelApp (GEL)**. Updated panels are **versioned** t o keep track of existing interpreted analysis.
- Several **rare disease interpretation analysis** implemented such as TEAM or Tiering which is based on GEL RD Tiering tool (Cancer interpretation analysis coming soon). You can use one or more disease panels in the interpretation analysis.
- You can save more than one interpretation analysis result in the *clinical analysis* to create one or more **clinical reports**.
- Together with a tier classification a **semi-automatic ACMG classification** has been also implemented.

Pathogenic Variant Database

- Interpreted variants – and their variant annotation – can be indexed in a high-performance **patho genic variant database**. Clinical data from *catalog,* the *clinical analysis* and *interpretation* are also indexed together with interpreted variants.
- Real-time **queries** and **complex aggregations** have been implemented.

RESTful Web Services

OpenCGA implements more than 150 RESTful web services to allow users to manipulate and query Catalog metadata and data such as *alignment*, *variants* and *pathogenic variants*. REST web services are documented using Swagger, you can see OpenCGA Swagger documentation at http://bioinfo.hpc.cam.ac.uk/hgva/webservices/. To facilitate the usage all of these web services we have implemented different client libraries and a command line (see below in *Usability).*

REST web services can be grouped in different categories: *Catalog, Alignment, Variant, Clinical* and *Admi n.*

Catalog

- Catalog **data manipulation**, you can create, update, delete change permission of data.
- **Advanced search** web services to query any resource (file, samples, ...)

Alignment

- You can **index** BAM files to query reads and calculate **coverage** in BigWig format
- **Query** endpoint to fetch alignments in GA4GH format from several files. Filters implemented include: region, mapping quality, number of mismatches, number of hits, properly paired, ...

Variant

- Query variant endpoint allows to query variants by any variant filter. Full control of which fields are returned
- Aggregation stats implemented.
- Others: fetch old variant annotation, variant study metadata, ...

Clinical

- Several web services to create clinical analysis, execute interpretations or query pathogenic variant database.

Admin

- Administrative web services, only OpenCGA root user can execute them

Usability

REST Clients

- Four REST clients have been implemented in different programming language: Java, Pythong, R and JavaScript.

Command-line Interface (CLI)

- A fully functional command-line has been implemented

Visualisation

OpenCGA web catalog

- Web-based application to query and aggregate metadata from catalog

IVA

- Web-based application for Intercative Variant Analysis
- Highly customisable
- Plugin oriented

Genome Browser

- Genome browser for NGS