

Introducing the GOR API

Purpose-built for Translational Research from the ground up, the Genomically Ordered Relational Database (GORdb) gives bioinformaticians the power to ingest and query next generation sequencing (NGS) data at biobank scale. Petabyte-performant™ speed is made possible by the genomically-ordered, yet index-free, nature of GOR.

Researchers can interact with this query engine using the GOR query language (GORql), which provides a rich set of commands to join, filter, and aggregate genomic data. With its pipe syntax, a hybrid of SQL and Unix shell scripting, users will feel right at home utilizing GORql for iterative query-driven discovery. The language is also ideal for preprocessing in pipeline and data science environments in that it allows for extensions, written in any programming language, to be seamlessly piped into its parallel execution system.

Flexible Integration via API

Genuity Science (formerly WuXi NextCODE) recognizes that computational biology teams have invested heavily in developing in-house analytical workflows and data warehouse infrastructure across multiple omics. Consequently, the RESTful API of GORdb makes it easy to build on top of an existing in-house architecture in order to integrate a scalable analytical engine for genomics. With just a few lines of code, bioinformaticians can call the GORapi from their programming language of choice and integrate genomic data into their aggregate research initiatives.

GOR API Services

Release I Core Services	Release III Management Services
1. User Management (SSO-enabled)	7. Job Management
2. GOR Direct Query	8. Metering (performance & billing)
	9. Audit Logging
Release II Core Services	
3. GOR Template Query	
4. GORetl (extract, transform, load)	
5. Project Data Organization	

R SDK

The R programming language is the foundation for statistical analysis in bioinformatics. GORdb provides a software development kit (SDK) for R that helps accelerate and standardize how teams can integrate GOR queries directly into their R scripts, e.g. within R Studio. Using the SDK, server-side data can be seamlessly queried alongside local files and session-based intermediate results. By default, the GORapi asynchronously provides compact query results in TSV format with an option for JSON. Furthermore, the R SDK simplifies downstream analysis by formatting responses as R data frames with built-in support for Tidyverse data frames.

With the combination of (a) GOR's whole-genome query capability, (b) an SDK designed for use with R Studio as an IDE, and (c) the thousands of R packages for statistical modeling and data visualization - bioinformaticians finally have the freedom to query the genome using their native toolset.

Flexible Deployment

GORdb is available in major clouds including AWS and GCP with support for S3, GS, and DNAnexus object storage. The system can also be deployed on-premise with support for NFS and POSIX block storage. In both cloud and local deployments, GORdb integrates with Single Sign-On (SSO) and Identity Access Management (IAM) solutions such as Active Directory.

The GOR Developer Console is deployed alongside every instance of GORdb. Within this web-based user interface, researchers will find everything they need to get started; including their API Key, tutorials, endpoint reference documentation, and downloadable packages.

Partner-Driven Development

Genuity Science is working closely with a select group of customers to develop additional API services in cycles of close, iterative feedback. Building upon the Direct Query Service, the focus of the next phase of development is the Query Template Service. Templates allow users to define a GOR query and the parameters that it should accept at a RESTful endpoint.

```

my-gor-script.R
1 library(gorr)
2 conn <- gor_connect(
3   host = "10.0.0.1", port = "8888",
4   project = my_project, protocol = "https",
5   header = header)
6 results <- gor_query(
7   my_query.gor, conn)
8 results

Console
--- Executing Query ---
gor #WGSvars#
| top 10000
| where FILTER = 'PASS'
| varjoin -l -r /ref/dbnsfp_max.gorz
-----
Submitting query
Server assigned query ID #346994
Fetching results (Page #1, Offset: 0, Limit: 0)
--- Done ---
> results
# tibble: 10,000 * 25
  chrom pos   ref call filter tumor gene_sym
  <chr> <int> <chr> <chr> <chr> <chr> <chr>
1 chr1  959201 G     C     PASS  143:~ NOC2L
2 chr1  959201 G     C     PASS  143:~ NOC2L
3 chr1  960672 C     T     PASS  110:~ KLHL17
4 chr1  960672 C     T     PASS  110:~ KLHL17
5 chr1  962397 G     T     PASS  267:~ KLHL17
    
```