Querying Variant Data

Table of Contents:

Overview

The main goal for indexing variant data into OpenCGA Storage is to be able to make queries and extract this data in a efficient way. There are different alternatives ways to access to the data (via CLI, RESTful , Java API, Python API, ...) and multiple implementations of the VariantStorageManager (OpenCGA Storage MongoDB, OpenCGA Storage Hadoop, ...).

All this layers and implementations will use the same specification defined in this document.

There are defined an small set of READ-ONLY methods to achieve all the required functionality.

- **Query:** Return all variants that matches with a given query
- **Count:** Count the result of a given query
- **Aggregation Stats:** Group variants by some field, you can count by region with this query.

Query Parameters

A filter is a pair of `<key>, <value>`, where the keys are predefined, and the values are defined by the user, using an specific format. In the next sections, all this keys are going to be enumerated, explaining their effect and the required format of the value.

There are some general rules that are applied for every case:The API described here fetches the sample data from a variant

1. Returned variants will match positively with **all the filters**, except with the **positional filters**. Variants will need to match with, at least, one positional filter (if any).

   The positional parameters are:

   - **region**
   - **id**
   - **gene**
   - **panel**
   - **xref**
2. When a filter accepts a list of values, the filter can be configured to pass ALL filters or ANY filter depending on the separator:

   - **ANY**: (Comma "**`,`**" separator) The filter will be valid if the variant passes **ANY** of the filters.
     Define an **`OR`** operation between the elements in the query.
     Example: "type : SNV,INDEL" : Variants of type SNV **OR** INDEL
   - **ALL**: (Semicolon "**`;`**" separator) The filter will be valid if the variant passes **ALL** filters.
     Define an **`AND`** operation between the separated elements
     Example: "study : study1;study2" : Variants present in study1 AND in study2

So, the query `ct: missense_variant,stop_lost` will return all variants that are either missense_variant, stop_lost or both, but `ct: missense_variant;stop_lost` will return variants that are at the same time, missense_variant and stop_lost (in two different overlapping transcripts).

Genomic Parameters

This general filters will match with fields from the VCF input files.

All this parameters are positive filters. The output will contain variants that match with this filters

Parameter	Description	Example
id	List of IDs, these can be rs IDs (dbSNP) or variants in the format chrom:start:ref:alt,	rs116600158 COSM6350960 19:7177679:C:T
region	List of regions, these can be just a single chromosome name or regions in the format `<chromosome>:<start>-<end>`	chr22 3:100000-200000

type	List of types, accepted values are SNV, MNV, INDEL, SV, CNV, INSERTION, DELETION,	SNV, INDEL
gene	List of genes, most gene IDs are accepted (HGNC, Ensembl gene, ...). This is an alias to 'xref' parameter. When used together with ct, biotype or transcriptFlag, all filters will need to match within the same transcript.	BRCA2 BMPR ENSG00000174173 ENST00000495642
panel	Filter by genes from the given disease panel	
xref	List of any external reference, these can be genes, proteins or variants. Accepted IDs include HGNC, Ensembl genes, dbSNP, ClinVar, HPO, Cosmic, ...	BRCA2 ENST00000495642 COSM6350960 VAR_057355 rs116600158

Sample and Family Parameters

Parameter	Description	Example
project	Project [user@]project where project can be either the ID or the alias	
study	Filter variants from the given studies, these can be either the numeric ID or the alias with the format user@project:study	
sample	Filter variants where the samples contain the variant (HET or HOM_ALT). Accepts AND (;) and OR (,) operators. This will automatically set 'includeSample' parameter when not provided	HG0097, HG00978
genotype	Samples with a specific genotype: {samp_1}:{gt_1}(,{gt_n})*(;{samp_n}:{gt_1}(,{gt_n})*)* Unphased genotypes (e.g. 0/1, 1/1) will also include phased genotypes (e.g. 0|1, 1|0, 1|1), but not vice versa. Genotype aliases accepted: HOM_REF, HOM_ALT, HET, HET_REF, HET_ALT and MISS This will automatically set 'includeSample' parameter when not provided	HG0097:0/0; HG0098:0/1, 1/1 HG0097: HOM_REF; HG0098: HET_REF, HOM_ALT
sampleAnnotation	Selects some samples using metadata information from Catalog.	age>20; phenotype=hpo:123, hpo:456; name=smith
format	Filter by any FORMAT field from samples. [{sample}:]{key}{op}{value}[,;]*. If no sample is specified, will use all samples from "sample" or "genotype" filter. Many FORMAT fields can be combined.	DP>200 HG0097: DP>200, HG0098: DP<10 . HG0097: DP>200; GT=1/1,0/1, HG0098: DP<10

file	Filter variants from the files specified. This will set includeFile parameter when not provided	
info	Filter by INFO attributes from file. [{file}:]{key}{op}{value}[,;]* If no file is specified, will use all files from "file" filter. Many INFO fields can be combined.	AN>200 file_1.vcf: AN>200; file_2.vcf: AN<10 file_1.vcf: AN>200; DB=true; file_2.vcf: AN<10
filter	Specify the FILTER for any of the files. If 'file' filter is provided, will match the file and the filter.	PASS, LowGQX
qual	Specify the QUAL for any of the files. If 'file' filter is provided, will match the file and the qual.	>123.4
family	Filter variants where any of the samples from the given family contains the variant (HET or HOM_ALT)	
familyMembers	Sub set of the members of a given family	
familyDisorder	Specify the disorder to use for the family segregation	
familySegregation	Filter by mode of inheritance from a given family. Accepted values: [monoallelic, monoallelicIncompletePenetrance, biallelic, biallelicIncompletePenetrance, XlinkedBiallelic, XlinkedMonoallelic, Ylinked, MendelianError, DeNovo, CompoundHeterozygous]	

Cohort Stats Parameters

Apart from the data provided on the files, there are some statistics calculated from the genotypes, or parsed from the INFO column, if the input was an aggregated file.

This filters are related with the statistics from a specific study and cohort. Knowing that, the format will be the same for each filter: `<study>:<cohort><comparator><value>`, where the available comparators are: <, <=, >, >=, = and `!=`.

Parameter	Description	Example
cohort	Select variants with calculated stats for the selected cohorts	
cohortStatsRef	Reference Allele Frequency: [{study:}]{cohort}[<|>|<=|>=]{number}.	ALL>0.6
cohortStatsAlt	Alternate Allele Frequency: [{study:}]{cohort}[<|>|<=|>=]{number}.	ALL<=0.4
cohortStatsMaf	Minor Allele Frequency: [{study:}]{cohort}[<|>|<=|>=]{number}.	study:ALL<0.01
cohortStatsMgf	Minor Genotype Frequency: [{study:}]{cohort}[<|>|<=|>=]{number}.	COH1<0.1,COH2<0.3

Variant Annotation Parameters

Parameter	Description	Example
biotype	List of biotypes, When used together with gene, ct or transcriptFlag, all filters will need to match within the same transcript.	protein_coding

ct	List of SO consequence types, When used together with gene, biotype or transcriptFlag, all filters will need to match within the same transcript.	missense_variant, stop_lost SO:0001583,SO:0001578
proteinSubstitution	Protein substitution scores include SIFT and PolyPhen. You can query using the score {protein_score}[<|>|<=|>=]{number} or the description {protein_score}[~=|=]{description}	polyphen>0.1,sift=tolerant
conservation	Filter by conservation score: {conservation_score}[<|>|<=|>=]{number}	phastCons>0.5,phylop<0.1,gerp>0.1
populationFrequencyAlt	Alternate Population Frequency: {study}:{population}[<|>|<=|>=]{number}.	1kG_phase3:ALL<0.01
populationFrequencyRef	Reference Population Frequency: {study}:{population}[<|>|<=|>=]{number}.	1kG_phase3:ALL<0.01
populationFrequencyMaf	Population minor allele frequency: {study}:{population}[<|>|<=|>=]{number}.	1kG_phase3:ALL<0.01
transcriptFlag	List of transcript annotation flags. When used together with gene, biotype or ct, all filters will need to match within the same transcript.	CCDS, basic, cds_end_NF, mRNA_end_NF, cds_start_NF, mRNA_start_NF, seleno
geneTraitId	List of gene trait association id.	umls:C0007222 OMIM:269600
trait	transcriptFlagList of traits, based on ClinVar, HPO, COSMIC, i.e.: IDs, histologies, descriptions,...	
clinicalSignificance	Clinical significance: benign, likely_benign, likely_pathogenic, pathogenic	
go	List of GO (Gene Ontology) terms.	GO:0002020
expression	List of tissues of interest.	lung
proteinKeyword	List of Uniprot protein variant annotation keywords	
drug	List of drug names	
functionalScore	Functional score: {functional_score}[<|>|<=|>=]{number}	cadd_scaled>5.2 , cadd_raw<=0.3
customAnnotation	Custom annotation: {key}[<|>|<=|>=]{number} or {key}[~=|=]{text}	
annotationExists	Return only annotated variants	

Parameter combination

When using together two or more filters of **gene**, **ct**, **biotype** or **transcriptFlag**, all filters will need to match within the same transcript. [#1214]

Query Options

Modifies over the variants to return.

Key	Description	Example
limit	Number of elements to return.	100
skip	Number of elements to skip.	100
sort	Sort variants by position	true

include	Fields from the Variant's model to be included in the response See Variant Fields.	chromosome,start, reference,alternate
exclude	Fields from the Variant's model to be excluded in the response. Ignored if "include" is present. See Variant Fields.	studies.stats, annotation. expression
summary	Selects an small amount of fields to return. Ignored if "include" or "exclude" are present. See Variant Fields.	true
includeFormat	List of FORMAT names from Samples Data to include in the output. Accepts "all" and "none".	AD,DP
includeGenotype	Include genotypes, apart of other formats defined with include If "GT" is not provided in "includeFormat" or this parameter is false, genotypes won't be returned.	true
includeStudy	List of studies to be included in the result. Accepts "all" and "none".	
includeFile	List of files to be included in the result. Accepts "all" and "none".	
includeSample	List of samples to be included in the result. Accepts "all" and "none".	
unknownGenotype	Returned genotype for unknown genotypes. Common values: [0/0, 0|0, ./.]	
sampleLimit	Limit the number of samples to be included in the result	
sampleSkip	Skip some samples from the result. Useful for sample pagination.	
sampleMetadata	Return the samples metadata group by study. Sample names will appear in the same order as their corresponding genotypes.	

Variant Fields

The parameters **include** and **exclude** accepts a list of Variant Fields. This is a list with all the accepted values. Some short alias to those fields are listed in *italic*.

- id
- chromosome
- start
- end
- reference
- alternate
- length
- type
- studies
    - studies.samplesData | *samples* | *samplesData*
    - studies.files | *files*
    - studies.stats | *stats*
    - studies.secondaryAlternates
    - studies.studyId
- annotation
    - annotation.ancestralAllele
    - annotation.id
    - annotation.xrefs
    - annotation.hgvs
    - annotation.displayConsequenceType
    - annotation.consequenceTypes
    - annotation.populationFrequencies
    - annotation.minorAllele
    - annotation.minocohortStatsRefrAlleleFreq

- annotation.conservation
- annotation.geneExpression
- annotation.geneTraitAssociation
- annotation.geneDrugInteraction
- annotation.variantTraitAssociation
- annotation.functionalScore
- annotation.additionalAttributes