README

Background on the Genome Aggregation Database Canada (gnomAD-Canada)

Building on the success of the Genome Aggregation Database (gnomAD) from the Broad Institute, gnomAD-Canada aims to provide a comprehensive resource for population-wide allele frequencies specific to Canada. Managed by a dedicated Canadian team, this platform will compile genetic variation data from 60,000 Canadian genomes, including those from historically underrepresented populations. By adding phenotype information, increasing sample size and ancestral diversity, gnomAD-Canada will improve the diagnosis of rare conditions. The gnomAD-Canada database was created as part of the Canadian Genomic Data Commons (CGDC) and the Pan-Canadian Genome Library (PCGL).

As a participating site in the Federated gnomAD project, gnomAD-Canada will process and quality control its data according to gnomAD best practices and contribute aggregate allele frequency data to the global gnomAD resource, ensuring secure, policy-compliant sharing with the broader scientific and clinical community.

All of the raw data from contributing gnomAD-Canada project is processed through equivalent pipelines, quality controlled (QC), and jointly variant-called to increase consistency across projects. These pipelines are implemented in Python using the Hail framework, a scalable platform for genomic data analysis that leverages batch computing. The gnomAD-Canada v1.0 release uses 10,487 genomes from the HostSeq project. It is available publicly via download for the benefit of the wider biomedical community. For terms of use and other policies please see our policy page.

The aggregation and release of summary data from the genomes for gnomAD-Canada has been approved by the Sinai Health Research Ethics Board, Mount Sinai Hospital (Project: “The Canadian Genomics Data Commons”) and the HostSeq DACO.

Data description (HostSeq)

The Canadian COVID-19 HostSeq cohort[1] was launched in April 2020 from CGen, Canada’s national platform for genome sequencing and analysis. The study comprises fourteen clinical and epidemiological projects that recruited SARS-CoV-2 infected participants and controls over the course of the COVID-19 pandemic, and contains over 10,000 sequenced Canadian genomes geographically dispersed throughout the country.

Alignments, variant-calling, and joint-calling for HostSeq

Alignment and variant-calling pipeline overview:

- Dragmap align – Runs the Dragen alignment tool to align short sequencing reads against the reference (hg38) > Generates BAM
- Picard – Takes coordinate sorted BAM and calculates tags by comparing to reference
- GATK recalibration – Run Base Quality Score Recalibration (BQSR) to detect systematic errors made during sequencing and adjust quality scores accordingly
- Flagstats/bamstats – Generates text files with X-coverage and other statistics from the BAM file

- Samtools faidx – Index/query the reference fasta file
- GATK CreateSequenceDictionary – Generate sequence dictionary for reference sequence
- GATK HaplotypeCaller – Call germline SNVs and INDELs via local reassembly of haplotypes > Generates gVCF
- GATK index feature file – Creates an index for the gVCF file

Whole genome sequencing (WGS) was performed on a cohort (N = 10,487) across Canada enrolled in projects contributing data to HostSeq. Data processing and variant-calling were performed in-house at three CGEn sites using the Genome Analysis Toolkit (GATK) best practices.

The base calling was performed using the Illumina NovaSeq6000 platform at 30X depth of coverage. The resulting short-read sequences (FASTQ format) were aligned to the GRCh38 human reference genome using the DRAGEN mapper (DRAGMAP v1.3.0). The sequences were then de-duplicated using Picard tools (v2.25.0), and base quality scores were recalibrated using Base Quality Score Recalibration (BQSR) with the Genome Analysis Toolkit (GATK v4.2.5.0) to correct sequencing errors and adjust quality scores accordingly. Variant Call Format (VCF) files for single nucleotide variations (SNVs) and short insertions and deletions (indels) were generated using GATK HaplotypeCaller in DRAGEN mode on diploid samples for short variant discovery. Generated per-sample variant calls were stored in genomic VCF (gVCF) files, which are imported into a GATK GenomicsDB for joint calling using GATK GenotypeGVCFs, which further refines per-sample variant calls based on the observed data across all samples, generating a joint VCF[1].

Quality control procedures included assessing sample contamination using VerifyBamID2 (v2.0.1), with a contamination threshold of >3%; concordance between reported sex-at-birth and sex chromosome composition to eliminate possible sample swaps using PLINK (v1.90); and inference of genetic ancestry and relatedness using Genetic Relationship and Fingerprinting software (GRAF v2.4). Additional genetic data analyses are performed using PLINK (v2.00) and R (v3.6.3). Samples were also excluded based on genotyping call rate (<95%) and mean depth of coverage (<10X).

Following joint calling and cohort-level quality control, the dataset was adapted to meet the input requirements of the gnomAD QC v3 open-source pipeline[2]. The gnomAD QC v3 workflow was deployed within the BC Cancer Genome Sciences Centre (GSC) server environment, with HostSeq serving as the demonstration dataset for the implementation, benchmarking, and validation of the pipeline infrastructure as part of the gnomAD-Canada activity. The Broad Institute's "gnomad_qc" pipeline was adapted for use and operation with on-premises resources, including various stages for sample QC, variant QC, and annotation of the final release that are described in some detail below.

Creation of VDS from g.vcf files

The "gnomad_qc" compatible VariandDataset (VDS) was generated using the Hail/VDS combiner (v0.2.108). The HostSeq dataset was processed in batches of ~100 samples per VDS segment, which were ultimately merged into a final ~10,000 sample dataset. Criteria for inclusion in the dataset included: (i) mean depth of coverage (>10X), (ii) per chromosome coverage >70% (refers to the fraction of bases on a chromosome covered by at least one read), (iii) mapped fraction per chromosome >0.85 (refers to the ratio of unmapped to mapped read segments per chromosome), (iv) number of variants per chromosome $>2.5 \times 10^5$ (threshold was applied as a flat per-chromosome floor to exclude samples with evidence of failed or incomplete variant calling), (v) total reads $>5.5\times10^8$ (extracted as a raw sequencing output metric to identify failures in sequencing prior to alignment). These cutoffs were determined empirically based on the distributions of each metric in the HostSeq dataset. Where applicable checks were restricted to autosomal chromosomes (1-22).The variant count threshold was additionally evaluated for chromosome X, while chromosome Y was excluded from all chromosome-specific metrics.

The generated VDS comprises 10,487 samples with an on-disk volume of ~19 TB. The final release is provided as a table of allele frequencies across the entire dataset and various population subsets, inferred through principal component analysis (PCA) implemented in the gnomAD pipeline. Genetic ancestry labels used in HostSeq are pre-defined by the gnomAD pipeline, which divides samples into continental population groups: African/African- American, Latino/ Admixed-American, European (non-Finnish), Ashkenazi Jewish, East Asian, European (Finnish), Middle Eastern, South Asian, and Remaining individuals (formerly the Other category). The final allele frequency table in Hail format is ~104 Gb in size for the HostSeq dataset.

Quality Control (gnomAD QC)

load_data/

`create_last_END_positions.py`

Compute the genomic position of the most upstream reference block overlapping each row on the raw sparse MatrixTable. This computation is used in downstream steps to filter to only relevant rows before a densification to only a subset of sites using (densify_sites).

sample_qc/

`sample_qc.py`

- `--sample_qc` - Compute Hail's sample QC metrics on the raw split MatrixTable stratified by bi-allelic and multi-allelic variants.
- `--impute_sex` - Impute chromosomal sex karyotype annotation.
- `--compute_hard_filters` - Determine the samples that fail hard filtering thresholds.
- `--compute_qc_mt` - Filter the full sparse MatrixTable to a smaller dense QC MatrixTable based on liftover of gnomAD v2 QC sites and Purcell 5k sites.
- `--run_pc_relate` - Run Hail's implementation of PC-relate to compute relatedness estimates among pairs of samples in the callset.
- `--run_pca` - Perform genotype PCA on unrelated samples and project related samples onto PCs.
- `--assign_pops` - Fit random forest (RF) model on PCs for samples with known ancestry labels and apply that RF model to assign ancestry to remaining samples.
- `--calculate_inbreeding` Calculate sample level inbreeding coefficient. This is not currently recommended for use because of ancestral differences that will impact this calculation.
- `--calculate_clinvar` Calculate counts of ClinVar and ClinVar Pathogenic/Likely Pathogenic variants per sample. Used to investigate potential project specific differences.
- `--apply_stratified_filters` - Evaluate sample quality metrics by population and flag outlier samples.
- `--apply_regressed_filters` - Regress out the PCs used for the ancestry assignment (`--run_pca`) and flag samples that are outliers based on the residuals for each of the QC metrics.
- `--compute_related_samples_to_drop` - Determine related samples to drop by computing global sample ranking (based on hard-filters, releasable and coverage) and then using Hail's maximal_independent_set.
- `--generate_metadata` - Combine project specific metadata, sample_qc (`--sample_qc`), sex imputation (`--impute_sex`), hard filter (`--compute_hard_filters`), relatedness (`--run_pc_relate` and `--compute_related_samples_to_drop`) and ancestry inference (`--run_pca` and `--assign_pops`) into a unified metadata Table. Define the release sample set.

`create_fam.py`

- `--find_dups` - Create a table with duplicate samples indicating which one is the best to use.
- `--infer_families` - Infer all complete trios from kinship coefficients (`sample_qc.py --run_pc_relate`) and sex imputation annotations (`sample_qc.py --impute_sex`), including duplicates.
- `--run_mendel_errors` - Calculate Mendelian violations on the inferred complete trios (`--infer_families`) and a random set of trios.
- `--finalize_ped` - Create a final ped file by excluding families where the number of Mendel errors or de novos are higher than those specified in `--max_dnm` and `--max_mendel`.

annotations/

`generate_qc_annotations.py`

- `--compute_info` - Compute a Table with the typical GATK allele-specific (AS) and site-level info fields as well as ACs and lowqual fields.
- `--split_info` - Split the alleles of the info Table (`--compute_info`).
- `--export_info_vcf` - Export the split info Table (`--split_info`) as a VCF.

variant_qc/

`random_forest.py`

- `--annotate_for_rf` - Gather variant- and allele-level annotations used as features for training the variant QC random forests model, impute any missing entries.
- `--train_rf` - Select variants for training examples and train random forests model using specified parameters for depth and number of trees. If specified, test resulting model on a pre-selected region. Save the training data with metadata describing random forest parameters used.
- `--apply_rf` - Apply random forest model to full variant set.

`evaluation.py`

- `--create_bin_ht` - Create Table with bin annotations based on the ranking of random forest and/or VQSR variant quality scores, with SNVs and indels handled separately. Additional bin annotations are added for the following stratifications: bi-allelic variants, singletons, and bi-allelic singletons.
- `--create_aggregated_bin_ht` - Compute aggregated metrics for each bin that are useful for comparison of variant filtering performance across multiple random forest models (and VQSR if used).
- `--extract_truth_samples` - Extract truth samples (NA12878 and synthetic diploid sample) from the full callset MatrixTable for comparison to their truth data.
- `--merge_with_truth_data` - Compute a table for each truth sample (NA12878 and synthetic diploid sample) comparing the truth sample in the callset to the truth data.
- `--bin_truth_sample_concordance` - Create a concordance Table of the filtering model (e.g., VQSR, random forest) against truth data (NA12878 and synthetic diploid sample) binned by rank (both absolute and relative). Used for evaluating the variant filtering models.

final_filter.py

Create Table containing the variant filter status based on SNP and indel quality cutoffs and an inbreeding coefficient threshold.

Making the Release

annotations/

generate_freq.py

Compute frequencies of variants in gnomAD for various sample groupings (e.g., ancestry, sex, subsets) and for downsampled sets and add filtering allele frequency annotations for ancestry sample groupings.

generate_qc_annotations.py

- `--generate_allele_data` - Determine the following annotations for each variant in the split info Table: variant type (SNV, indel, multi-SNV, multi-indel, mixed), allele type (SNV, insertion, deletion, complex), and the total number of alleles present at the site.
- `--generate_ac` - Allele count per variant (raw and adj filtered genotypes) of high quality samples, high quality unrelated samples, and release samples.
- `--generate_fam_stats` - Calculate transmission and de novo statistics using trios (`sample_qc.py --finalize_ped`).
- `--vep` - Get the Ensembl Variant Effect Predictor (VEP) annotation for each variant.

create_release/

create_release_sites_ht.py

Combine frequency, filtering allele frequency, variant QC, VEP, dbSNP, in silico scores, and variant QC metric histogram annotations into unified table. Add frequency annotations for each sample subset.

Downloading the Release

The release sites Hail table is available as `release_sites_hostseq.tar.gz` (~104 GB). It can be downloaded from the portal using `wget`:

```
wget https://gnomad.bcgsc.ca/gnomad/download/
  release_sites_hostseq.tar.gz
```

To extract:

```
tar -xzf release_sites_hostseq.tar.gz
```

The extracted Hail table can then be loaded in Python:

```
import hail as hl
hl.init()
```

```
ht = hl.read_table("release_sites_hostseq.ht")
ht.describe()
```

Release Sites Table: Field Descriptions

The release sites Hail table contains global and per-variant (row) fields described below.

Global Fields

- `freq_meta` — Allele frequency metadata. An ordered list containing the frequency aggregation group for each element of the `freq` array row annotation.
- `freq_index_dict` — Dictionary keyed by label grouping combinations (group: adj/raw, gen_anc: gnomAD inferred global population, sex: sex karyotype), with values describing the corresponding index of each grouping entry in the `freq` array row annotation.
- `faf_index_dict` — Dictionary keyed by label grouping combinations (group: adj/raw, pop: gnomAD inferred global population, sex: sex karyotype), with values describing the corresponding index of each grouping entry in the filtering allele frequency (`faf`) row annotation.
- `faf_meta` — Filtering allele frequency metadata. An ordered list containing the frequency aggregation group for each element of the `faf` array row annotation.
- `VEP version` — VEP version that was run on the callset.
- `vep_csq_header` — VEP header for VCF export.
- `dbsnp_version` — dbSNP version used in the callset.
- `freq_sample_count` — A sample count per sample grouping defined in the `freq_meta` global annotation.
- `filtering_model` — The variant filtering model used and its specific cutoffs.
    - `model_name` — Variant filtering model name used in the `filters` row annotation, indicating the variant was filtered by this model during variant QC.
    - `score_name` — Annotation name of the score used for variant filtering.
    - `snv_cutoff.bin` — Filtering percentile cutoff for SNVs.
    - `snv_cutoff.min_score` — Minimum score at SNV filtering percentile cutoff.
    - `indel_cutoff.bin` — Filtering percentile cutoff for indels.
    - `indel_cutoff.min_score` — Minimum score at indel filtering percentile cutoff.
    - `model_id` — Variant filtering model ID for score data (used for internal specification of the model).
    - `snv_training_variables` — Variant annotations used as features in the SNV filtering model.
    - `indel_training_variables` — Variant annotations used as features in the indel filtering model.
- `age_distribution` — Callset-wide age histogram calculated on release samples.
    - `bin_edges` — Bin edges for the age histogram.
    - `bin_freq` — Bin frequencies for the age histogram. Number of records found in each bin.
    - `n_smaller` — Count of age values falling below the lowest histogram bin edge.
    - `n_larger` — Count of age values falling above the highest histogram bin edge.

Row Fields

Variant Identity

- `locus` — Variant locus. Contains contig and position information.

- alleles — Variant alleles.
- rsid — dbSNP reference SNP identification (rsID) numbers.
- a_index — The original index of this alternate allele in the multiallelic representation (1 is the first alternate allele or the only alternate allele in a biallelic variant).
- was_split — True if this variant was originally multiallelic, otherwise False.

Allele Frequencies

- freq — Array of allele frequency information (AC, AN, AF, homozygote count) for each frequency aggregation group in the release.
    - AC — Alternate allele count in release.
    - AF — Alternate allele frequency (AC/AN) in release.
    - AN — Total number of alleles in release.
    - homozygote_count — Count of homozygous alternate individuals in release.
- grpmax — Allele frequency information for the non-bottlenecked genetic ancestry group with the maximum allele frequency. Excludes Amish (ami), Ashkenazi Jewish (asj), European Finnish (fin), Middle Eastern (mid), and "Other" (oth).
    - AC — Alternate allele count in the population with the maximum allele frequency.
    - AF — Maximum alternate allele frequency (AC/AN) across populations.
    - AN — Total number of alleles in the population with the maximum allele frequency.
    - homozygote_count — Count of homozygous individuals in the population with the maximum allele frequency.
    - gen_anc — Population with maximum allele frequency.
    - faf95 — Filtering allele frequency (Poisson 95% CI) for the population with the maximum allele frequency.

Filtering Allele Frequency

- faf — Filtering allele frequency.
    - faf95 — Filtering allele frequency (Poisson 95% CI).
    - faf99 — Filtering allele frequency (Poisson 99% CI).
- gen_anc_faf_max — The genetic ancestry population in which the variant has its highest filtering allele frequency.
- faf95_max — Maximum filtering allele frequency (Poisson 95% CI).
- faf95_max_gen_anc — Genetic ancestry group with the maximum filtering allele frequency (95% CI).
- faf99_max — Maximum filtering allele frequency (Poisson 99% CI).
- faf99_max_gen_anc — Genetic ancestry group with the maximum filtering allele frequency (99% CI).

Variant Filters

- filters — Variant filter status.
    - AC0 — Allele count is zero after filtering out low-confidence genotypes (GQ < 20; DP < 10; AB < 0.2 for het calls).
    - AS_VQSR — Failed allele-specific VQSR thresholds (-2.7739 for SNVs, -1.0606 for indels).
    - InbreedingCoeff — GATK InbreedingCoeff < -0.3.
    - PASS — Passed all variant filters.
- InbreedingCoeff — Inbreeding coefficient — excess heterozygosity at a variant site, computed as 1 – (observed heterozygous genotypes) / (expected heterozygous genotypes under HWE).
- Inbreeding_coeff_cutoff — Inbreeding coefficient threshold used to hard filter variants.

info Struct

Typical GATK allele-specific (AS) info fields and additional variant QC fields.

- `RF` — Likelihood of the variant being real (random forest score).
- `SB` — Per-sample strand bias components for Fisher's exact test: ref-fwd, ref-rev, alt-fwd, alt-rev read depths.
- `MQ` — Root mean square mapping quality of reads across all samples.
- `MQRankSum` — Z-score from Wilcoxon rank sum test of alt vs. reference read mapping qualities.
- `AS_ReadPosRankSum` — Allele-specific z-score from Wilcoxon rank sum test of alt vs. reference read position bias.
- `AS_pab_max` — Maximum p-value over callset for binomial test of observed allele balance for a heterozygous genotype (expected 0.5).
- `AS_MQ` — Allele-specific root mean square mapping quality across all samples.
- `AS_MQRankSum` — Allele-specific z-score from Wilcoxon rank sum test of alt vs. reference read mapping qualities.
- `FS` — Phred-scaled p-value of Fisher's exact test for strand bias.
- `AS_FS` — Allele-specific phred-scaled p-value of Fisher's exact test for strand bias.
- `ReadPosRankSum` — Z-score from Wilcoxon rank sum test of alt vs. reference read position bias.
- `AS_SB_TABLE` — Allele-specific forward/reverse read counts for strand bias tests.
- `AS_SOR` — Allele-specific strand bias estimated by the symmetric odds ratio test.
- `SOR` — Strand bias estimated by the symmetric odds ratio test.

Variant Annotations

- `singleton` — Variant is seen once in the callset.
- `transmitted_singleton` — Variant was a callset-wide doubleton transmitted within a family from parent to child (i.e., a singleton among unrelated samples).
- `omni` — Variant is present on the Omni 2.5 genotyping array and found in 1000 Genomes data.
- `mills` — Indel is present in the Mills and Devine data.
- `monoallelic` — All samples are homozygous alternate for the variant.
- `vep` — Consequence annotations from Ensembl VEP (with LOFTEE plugin).
- `region_flag` — Flags for problematic genomic regions.
    - `lcr` — Variant falls within a low complexity region.
    - `segdup` — Variant falls within a segmental duplication region.
- `allele_info` — Allele-level annotations.
    - `variant_type` — Variant type: snv, indel, multi-snv, multi-indel, or mixed.
    - `allele_type` — Allele type: snv, insertion, deletion, or mixed.
    - `n_alt_alleles` — Total number of alternate alleles observed at the variant locus.
    - `was_mixed` — Variant type was mixed.

Genotype Quality Histograms

The table contains two sets of genotype quality histograms: `raw_qual_hists` (computed on all genotypes) and `qual_hists` (computed on high quality genotypes only). Each contains the following fields:

- `gq_hist_all` — Histogram for genotype quality (GQ) across all samples. Bin edges: 0|5|10|…|100.
    - `bin_edges`, `bin_freq`, `n_smaller`, `n_larger`
- `gq_hist_alt` — Histogram for GQ in heterozygous individuals.
    - `bin_edges`, `bin_freq`, `n_smaller`, `n_larger`

- `dp_hist_all` — Histogram for read depth (DP) across all samples. Bin edges: 0|5|10|…|100.
    - `bin_edges`, `bin_freq`, `n_smaller`, `n_larger`
- `dp_hist_alt` — Histogram for DP in heterozygous individuals.
    - `bin_edges`, `bin_freq`, `n_smaller`, `n_larger`
- `ab_hist_alt` — Histogram for allele balance (AB) in heterozygous individuals. Bin edges: 0.00|0.05|…|1.00.
    - `bin_edges`, `bin_freq`, `n_smaller`, `n_larger`

Age Histograms

- `age_hist_het` — Age histogram for heterozygous release samples (high quality genotypes).
    - `bin_edges`, `bin_freq`, `n_smaller`, `n_larger`
- `age_hists.age_hist_hom` — Age histogram for homozygous release samples (high quality genotypes).
    - `bin_edges`, `bin_freq`, `n_smaller`, `n_larger`

Credits

Project Leadership

Jordan Lerner-Ellis (PI) · 1, 2, 3
Steven Jones (PI) · 4
Daniel Taliun (PI) · 5

gnomAD-Canada Development

Jordan Lerner-Ellis (PI) · 1, 2, 3
Steven Jones (PI) · 4
Daniel Taliun (PI) · 5
Rohan Abraham · 4
Eric Chuah · 4
Erika Frangione · 1, 2
Xu Xinyi · 1, 2
Navneet Aujla · 1, 2

We are grateful for the support of the CGDC and PCGL team leadership, users and collaborators, advisors and research teams:

https://genomicdatacommons.ca/team/ https://genomelibrary.ca/about-us/pcgl-working-groups/

HostSeq Implementation Committee

Naveed Aziz · 6
Steven Jones · 4
Bartha Knoppers · 5
Mark Lathrop · 5
Stephen W. Scherer · 7
Lisa Strug · 7
Stuart Turvey · 4

Contributing Organizations

1. Mount Sinai Hospital, Sinai Health
2. Lunenfeld-Tanenbaum Research Institute, Sinai Health
3. University of Toronto
4. BC Cancer Research Centre
5. McGill University
6. Genome Canada
7. The Hospital for Sick Children

Acknowledgements

This work was supported by the Canadian Foundation for Innovation (CFI; Project ID 43084) through the Canadian Genomic Data Commons (CGDC) and the Canadian Institutes of Health Research (CIHR; Project ID 190675; McGill 263096) through the Pan-Canadian Genome Library (PCGL).

This research has been conducted using CGEn's HostSeq Databank, funded by the Government of Canada through Genome Canada, under Project ID 'DACO-18'.

This study was approved by the Research Ethics Board of Mount Sinai Hospital, Sinai Health, Toronto (Project ID: 0250).

All gnomAD-Canada analyses were conducted in an appropriate manner consistent with the regulations of the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS 2), following the Chapter 9 policy and guidelines for research involving First Nations, Inuit, and Métis peoples[3], and with reference to the Indigenous Background Variant Library (IBVL)[4], a resource developed through the Silent Genomes Project that provides aggregated variant frequency data to support genomic interpretation for Indigenous populations while maintaining Indigenous data governance and sovereignty. **No attempts should be made to relabel gnomAD-Canada populations.**

References

1. Yoo S, Garg E, Elliott LT, et al. HostSeq: a Canadian whole genome sequencing and clinical data resource. *BMC Genom Data*. 2023;24(1):26. Published 2023 May 2. doi: 10.1186/s12863-023-01128-3

2. Chen, S.*, Francioli, L. C.*, Goodrich, J. K., et al. A genomic mutational constraint map using variation in 76,156 human genomes. *Nature*. 625, 92–100 (2024). https://doi.org/10.1038/s41586-023-06045-0 PMID: 38057664

3. Canadian Institutes of Health Research; Natural Sciences and Engineering Research Council of Canada; Social Sciences and Humanities Research Council of Canada. Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS 2) — Chapter 9: Research Involving the First Nations, Inuit and Métis Peoples of Canada. https://ethics.gc.ca/eng/tcps2-eptc2_2022_chapter9-chapitre9.html

4. BC Children's Hospital Research Institute. Indigenous Background Variant Library (IBVL). Silent Genomes Project. https://www.bcchr.ca/silent-genomes-project/ibvl