# Standardised and efficient querying of GWAS summary statistics by adapting the VCF format

Matthew Lyon1,2\*, Shea J Andrews3\*, Ben Elsworth2, Tom Gaunt1,2, Gibran Hemani2¥, Edoardo Marcora3¥

\* Joint first author

¥ Joint last author

Correspondence to Matthew Lyon (matt.lyon@bristol.ac.uk)

1. National Institute for Health Research Bristol Biomedical Research Centre
2. Medical Research Council (MRC) Integrative Epidemiology Unit (IEU), Bristol Medical School (Population Health Sciences), University of Bristol, Bristol, UK
3. Ronald M. Loeb Center for Alzheimer’s disease, Department of Neuroscience, Icahn School of Medicine at Mount Sinai, New York, NY, USA

## Abstract

## Introduction

The genome-wide association study (GWAS) is a powerful tool for identifying loci associated with traits, diseases and molecular phenotypes such as gene expression and biomarker concentration[1]. Sharing of non-identifiable test summary statistics (i.e. variant, effect size, standard error, p-value etc) has enabled a range of important secondary research applications including gene prioritization[2], causal inference (Mendelian randomization; MR)[3], risk prediction[1], genetic correlation[4] and heritability estimation[5]. However, data requirements for downstream analysis can vary widely, for example gene prioritization is undertaken with loci and trait association P-value while MR additionally demands allele and standard error information which is often unavailable. Even when the required data are present, existing tools have a plethora of input formats making analyses cumbersome and error prone. Storage and distribution of essential meta-data with summary statistics is also rarely practiced and can lead to errors if there is misinterpretation of the genome build, allele coding of the effect allele, or the trait and effect sizes units.

Lack of a common standard has led to GWAS analysis tools outputting results in different formats (e.g. plink[6], GCTA[7], BOLT-LMM[8], GEMMA[9] and meta-analysis tools e.g. METAL[10]). During secondary analyses processing issues are typically encountered: ambiguity in which allele relates to the effect size estimate, different field names and information, slow and unscalable queries. Some proposals have been made for standard formats. The EBI-NHGRI GWAS catalog developed a text format with standardized column names[11]. The SMR tool[12] proposed a way to store GWAS summary data in a binary format for rapid querying of quantitative trait loci. Learning from these examples and considering future needs i.e. GWAS of rare variants from genome sequencing data we have identified a set of requirements for a suitable universal format (Table 1).

We determined that adapting the widely used variant call format (VCF) [13], [14] was a convenient and constructive approach to meet these requirements. Here we describe the implementation, explain how it meets these requirements, describe existing and new software that creates and connects to the data format, and show results of query performance. Finally, we provide access to over 10,000 complete GWAS summary datasets that have been converted to this format as part of the IEU GWAS database, and are freely available for download: <https://gwas.mrcieu.ac.uk>.

## Implementation

The VCF format is organized into three units: flexible file header containing meta-data (lines beginning with ‘#’), variant information (one per row) and sample information (one per column). We adapt the format such that each sample data column represents GWAS of a single trait (Figure 1).

Meta-data define characteristics about the GWAS: trait description and units, genome build, number of variants, type of trait (continuous or case/control), sample size and study identifier. The VCF header is also mandatory for defining fields used in the file body including variable description, value requirements (i.e. number of values permitted and null values) and data type (i.e. string, number and boolean).

Each row of the file body contains a single variant position including contig (chromosome) name, base-pair position, variant identifier (i.e. dbSNP identifier), reference (major/non-effect allele) and alternative (minor/effect allele) alleles. The sample column is used to store allele-trait association metrics: marker identifier, allele frequency, regression coefficient, standard error and association P-value.

The full specification provides detailed information including reserved keys: <https://github.com/MRCIEU/gwas_vcf_spec>.

## How this format meets the specification

#### Human readable and easy to parse

The plain text file format can be easily read with any text viewer. Open-source parsing libraries are available in C (HTSLIB[15]) and Java (HTSJDK[15]) which can be implemented in most programming languages. Bcftools[16] is also available to provide user-friendly functionality accessible from the command line.

#### Unambiguous interpretation of the data

Data field descriptions and value types are required and defined in the file header. File validity is enforced during each read/write.

#### Unambiguous representation of bi-allelic, multi-allelic and insertion-deletion variants

Each locus (row) has capacity to store multiple alternative alleles as required. GWAS effect sizes are stored one per alternative allele allowing for bi/multi-allelic and insertion-deletion variants. HTSLIB[15] and HTSJDK[15] parsing libraries have routines for handling complex variants. Using this approach alternative allele(s) are always the regression coefficient effect allele allowing consistency between studies for ease of comparison.

#### Genomic information can be validated

The file header contains genome build, contig identifiers and sequence length. Reference alleles must match the specified reference FASTA. GATK[17] ValidateVariants can be used to verify file validity.

#### Flexibility on which GWAS fields are recorded and enforcement of essential fields

All fields are defined in the file header and can be set optional or required as desired.

#### Capacity to store meta-data about the study or studies

Each GWAS trait has a row in the file header to store trait description and units, number of variants, study type (case/control or continuous) and unique identifier.

#### Allows multiple studies to be stored together

The sample column was chosen to store GWAS association metrics to allow for mmultiple traits in a single file enabling distribution of related phenotypes or individually as desired.

#### Rapid querying by dbSNP identifier, genomic position range or GWAS summary data values

The file is sorted karyotypically and indexed to allow rapid queries by genomic position. Refer to query performance section for comparison with standard UNIX tools.

#### File compression

VCF files may be compressed with block GZIP[16] or converted to binary BCF file[16].

#### Readable by existing open-source tools

A large number of tools support VCF files including: GATK[17], Picard[18], bcftools[16], bedtools[19], vcftools[14] and plink[6]. Bcftools[16] can also provide a tabular extract for use with non-compatible tools.

#### Amenable to cloud-based streaming and database storage

Genomic intervals may be extracted over a network using a range-request without transferring the whole file. This is enables rapid streaming queries over the internet.

For high-throughput storage and querying, VCF files can be easily imported into GenomicsDB[20].

## Open source tools that use the format

We have developed open-source tools for mapping GWAS to and reading from VCF according to our specification (Table 2). We encourage users to provide feedback via the issue pages.

## Query performance

To assess query performance for extracting a single variant using chromosome position from either unprocessed tab-separated text or VCF (Figure 2), we obtained GWAS summary statistics from a large metanalysis of body mass index containing 2,336,269 variants [21]. Through each of the 100 repetitions we randomly chose a variant to query and then measured retrieval time using UNIX awk or bcftools[16] (v1.10).

The mean query time using gzip compressed VCF was 0.07 seconds (95% CI 0.07, 0.07) compared with 0.91 seconds for compressed text using gzip and awk (95% CI 0.91, 0.91) or 0.75 seconds for uncompressed text using awk directly (95% CI 0.74, 0.75). These tests suggest query time with VCF is around ten times quicker than using awk.

Can you also compare querying by

* Rsid
* Chrom position ranges
* P-value

## Conclusion

The VCF format is a robust solution to storing and distributing GWAS summary statistics. We find the query performance to be around ten times faster than reading the whole dataset with standard UNIX tools. To facilitate adoption, we have developed tools for mapping data from tabular data to VCF and reading using R or Python.

## Specification

Available from: <https://github.com/MRCIEU/gwas_vcf_spec>

## Data availability

Full summary statistics for over 10,000 GWAS in VCF format available from the IEU GWAS Database (<https://gwas.mrcieu.ac.uk>)

## Acknowledgments

This study was funded by the NIHR Biomedical Research Centre at University Hospitals Bristol National Health Service Foundation Trust and the University of Bristol. The views expressed are those of the author(s) and not necessarily those of the NIHR or the Department of Health and Social Care.

M.L., B.E., T.R.G. work in the Medical Research Council Integrative Epidemiology Unit at the University of Bristol, which is supported by the Medical Research Council and the University of Bristol (MC\_UU\_00011/4). G.H. is supported by the Wellcome Trust and Royal Society [208806/Z/17/Z].

## References

[1] P. M. Visscher *et al.*, “10 Years of GWAS Discovery: Biology, Function, and Translation,” *American Journal of Human Genetics*, vol. 101, no. 1. Cell Press, pp. 5–22, 06-Jul-2017.

[2] L. Hou and H. Zhao, “A review of post-GWAS prioritization approaches,” *Front. Genet.*, vol. 4, no. DEC, p. 280, Dec. 2013.

[3] G. D. Smith and S. Ebrahim, “‘Mendelian randomization’: Can genetic epidemiology contribute to understanding environmental determinants of disease?,” *International Journal of Epidemiology*. 2003.

[4] B. Bulik-Sullivan *et al.*, “LD score regression distinguishes confounding from polygenicity in genome-wide association studies,” *Nat. Genet.*, 2015.

[5] J. Yang, J. Zeng, M. E. Goddard, N. R. Wray, and P. M. Visscher, “Concepts, estimation and interpretation of SNP-based heritability,” *Nature Genetics*, vol. 49, no. 9. Nature Publishing Group, pp. 1304–1310, 01-Sep-2017.

[6] S. Purcell *et al.*, “PLINK: A tool set for whole-genome association and population-based linkage analyses,” *Am. J. Hum. Genet.*, 2007.

[7] J. Yang, S. H. Lee, M. E. Goddard, and P. M. Visscher, “GCTA: A tool for genome-wide complex trait analysis,” *Am. J. Hum. Genet.*, vol. 88, no. 1, pp. 76–82, Jan. 2011.

[8] P. R. Loh *et al.*, “Efficient Bayesian mixed-model analysis increases association power in large cohorts,” *Nat. Genet.*, vol. 47, no. 3, pp. 284–290, Feb. 2015.

[9] X. Zhou and M. Stephens, “Genome-wide efficient mixed-model analysis for association studies,” *Nat. Genet.*, vol. 44, no. 7, pp. 821–824, Jul. 2012.

[10] C. J. Willer, Y. Li, and G. R. Abecasis, “METAL: fast and efficient meta-analysis of genomewide association scans,” *Bioinforma. Appl. NOTE*, vol. 26, no. 17, pp. 2190–2191, 2010.

[11] A. Buniello *et al.*, “The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D1005–D1012, Jan. 2019.

[12] Z. Zhu *et al.*, “Integration of summary data from GWAS and eQTL studies predicts complex trait gene targets,” *Nat. Genet.*, vol. 48, no. 5, pp. 481–487, May 2016.

[13] “The Variant Call Format (VCF) Version 4.2 Specification,” 2019.

[14] P. Danecek *et al.*, “The variant call format and VCFtools,” *Bioinformatics*, vol. 27, no. 15, pp. 2156–2158, Aug. 2011.

[15] H. Li *et al.*, “The Sequence Alignment/Map format and SAMtools,” *Bioinforma. Appl. NOTE*, vol. 25, no. 16, pp. 2078–2079, 2009.

[16] H. Li, “A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data.,” *Bioinformatics*, vol. 27, no. 21, pp. 2987–93, Nov. 2011.

[17] A. McKenna *et al.*, “The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data,” *Genome Res.*, 2010.

[18] “broadinstitute/picard: A set of command line tools (in Java) for manipulating high-throughput sequencing (HTS) data and formats such as SAM/BAM/CRAM and VCF.” [Online]. Available: https://github.com/broadinstitute/picard. [Accessed: 25-Feb-2020].

[19] A. R. Quinlan and I. M. Hall, “BEDTools: a flexible suite of utilities for comparing genomic features,” *Bioinforma. Appl. NOTE*, vol. 26, no. 6, pp. 841–842, 2010.

[20] “GenomicsDB/GenomicsDB: Highly performant data storage in C++ for importing, querying and transforming variant data with Java/Spark. Used in gatk4.” [Online]. Available: https://github.com/GenomicsDB/GenomicsDB. [Accessed: 25-Feb-2020].

[21] L. Yengo *et al.*, “Meta-analysis of genome-wide association studies for height and body mass index in ~700 000 individuals of European ancestry,” *Hum. Mol. Genet.*, vol. 27, no. 20, pp. 3641–3649, 2018.

[22] K. Voss, J. Gentry, and G. Van Der Auwera, “GATK4 + WDL + Cromwell,” *F1000Research*, vol. 6, p. 4, Aug. 2017.

Table 1. List of requirements for GWAS summary statistics storage format

|  |  |
| --- | --- |
| # | Requirement |
| 1 | Human readable and easy to parse |
| 2 | Unambiguous interpretation of the data |
| 3 | Unambiguous representation of bi-allelic, multi-allelic and insertion-deletion variants |
| 4 | Genomic information can be validated |
| 5 | Flexibility on which GWAS fields are recorded and enforcement of essential fields |
| 6 | Capacity to store meta-data about the study or studies |
| 7 | Allows multiple studies to be stored together |
| 8 | Rapid querying by dbSNP identifier, genomic position range or GWAS summary data values |
| 9 | File compression |
| 10 | Readable by existing open-source tools |
| 11 | Amenable to cloud-based streaming and database storage |

Table 2. Open-source tools for working with the GWAS VCF format

|  |  |  |  |
| --- | --- | --- | --- |
| Program | Purpose | Implementation | Link |
| gwas2vcf | Mapping tab separated GWAS summary statistics and EBI format to VCF | Python3 (Docker) | <https://github.com/mrcieu/gwas2vcf> |
| gwas2vcfweb | Front-end and queue schedular for gwas2vcf | Python3, Cromwell[22]  (Docker) | <https://github.com/mrcieu/gwas2vcfweb> |
| gwasvcf | Library for querying and reading GWAS VCF files | R | <https://github.com/mrcieu/gwasvcf> |
| pygwasvcf | Library for querying and reading GWAS VCF files | Python3 | <https://github.com/mrcieu/pygwasvcf> |
| gwasglue | Library for processing GWAS summary statistics ready for secondary analysis | R | <https://github.com/mrcieu/gwasglue> |
| LD Score Regression[4] | Estimating genetic correlation and heritability | Python | <http://github.com/explodecomputer/ldsc> |

Figure 1. VCF format adapted to store GWAS summary statistics

Figure 2. Performance comparison for querying GWAS summary statistics in plain text and VCF format using chromosome position

A screenshot of a cell phone

Description automatically generated

Query time for extracting a single variant using chromosome position with tab-separated text or VCF format. GWAS summary statistics were obtained from a large metanalysis of body mass index containing 2,336,269 variants [21]. The simulation was performed on Ubuntu v18.04 running an Intel Xeon(R) 2.0 Ghz processor with 100 repetitions. During each repetition a variant was chosen at random and queried using awk or bcftools[16] v1.10. There were robust differences (paired t-test VCF vs compressed text mean query time [sec] P < 2.2 x 10-16) in mean query time between compressed VCF (0.07s [95% CI 0.07, 0.07]) and compressed (0.91s [95% CI 0.91, 0.91]) or uncompressed text (0.75s [95% CI [0.74, 0.75]).