**The variant call format provides efficient and robust storage of GWAS summary statistics**

Matthew Lyon (0000-0002-2500-1013)1,2\*, Shea J Andrews3\*, Ben Elsworth2, Tom R Gaunt1,2, Gibran Hemani2¥, Edoardo Marcora3¥

1. National Institute for Health Research Bristol Biomedical Research Centre, University of Bristol, Bristol, UK
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

\* These authors contributed equally to this work

¥ These authors contributed equally to this work

Word count:

**Genome-wide association study (GWAS) summary statistics are a fundamental resource for a variety of secondary research applications 1–6. Yet despite their widespread utility, no common storage format has been widely adopted, hindering tool development, data sharing, analysis and integration. Existing tabular formats 7,8 often ambiguously or incompletely store information about genetic variants and their associations, and also lack essential metadata increasing the possibility of errors in data interpretation and post-GWAS analyses. Additionally, data in these formats are typically not indexed, requiring the whole file be read which is slow and computationally inefficient. To address these issues, we propose an implementation of the variant call format (VCF) 9 and a suite of open-source tools for creating and using this format in downstream analyses. Simulations of query performance determine VCF is 9-46x faster than tabular alternatives when extracting variant(s) by genomic position. Our results demonstrate the VCF provides a robust and performant solution for analysis and sharing of GWAS data. To encourage adoption, we provide free access to over 10,000 complete GWAS summary datasets converted to this format (available from:** [**https://gwas.mrcieu.ac.uk**](https://gwas.mrcieu.ac.uk)**).**

**Main**

The GWAS is a powerful tool for identifying genetic loci associated with any trait, including diseases and clinical biomarkers, as well as non-clinical and molecular phenotypes such as height and gene expression 3 (eQTLs). Sharing of GWAS results as summary statistics (i.e. variant, effect size, standard error, p-value etc.) has enabled a range of important secondary research applications including: causal gene and functional variant prioritisation 1, causal cell/tissue type nomination 2, pathway analysis 3, causal inference (Mendelian randomization; MR) 4, risk prediction 3, genetic correlation 5 and heritability estimation 6. However, the utility of GWAS summary statistics is hampered by the absence of a universally adopted storage format and associated tools.

Historic lack of a common standard has resulted in GWAS analysis tools outputting results in different formats (e.g. plink 10, GCTA 11, BOLT-LMM 12, GEMMA 13 and meta-analysis tools e.g. METAL 14). As a consequence, various processing issues are typically encountered during secondary analysis. First, inconsistency and ambiguity of which allele relates to the effect size estimate (the “effect” allele). Confusion over the effect allele can have disastrous consequences on the interpretation of GWAS findings and the validity of post-GWAS analyses. For example MR studies may provide causal estimates with incorrect effect directionality 15. Likewise, prediction models based on polygenic risk scores might predict disease groups wrongly or suffer reduced power if some of the effect directionalities are incorrect. Related issues are encountered with naïve usage of dbSNP 16 identifiers which identify the variant base-position but do not distinguish between multiple alternative alleles (multiallelic variant). Second, summary statistic files are provided with a variety of content and field (column) names. Absent fields can limit analyses and although approaches exist to estimate the values of some of these missing columns (e.g. standard error from P value) imprecision is introduced reducing subsequent test power. Varying field names are easily addressed in principle but can be cumbersome and error prone. Third, data are frequently distributed with no or insufficient metadata describing the trait, measurement units, publication source, etc. which can lead to errors and impede research reproducibility. Fourth, querying plain text tabular files is slow and memory inefficient, making some potential applications computationally infeasible (e.g. systematic hypothesis-free analyses).

Some proposals for a standard format have been made. The EBI-NHGRI GWAS catalog developed a text format with uniform column names 7. The SMR tool 8 introduced a binary format for rapid querying of quantitative trait loci. These approaches are adequate for storing variant level summary statistics but do not enforce allele consistency or have capacity for essential metadata. Learning from these examples and our experiences performing high-throughput analyses across two research centres, we developed a set of requirements for a suitable universal format (Table 1). These features place emphasis on consistency and robustness, capacity for metadata to provide a full audit trail, efficient querying and file storage, ensuring data integrity, interoperability with existing open-source tools and across multiple datasets to support data sharing and integration. We determined that adapting the variant call format (VCF) 9 was a convenient and constructive solution to address these issues. We provide evidence demonstrating how the VCF meets our requirements and showcase capabilities of this medium (Table 1).

The VCF is organised into three components: a flexible file header containing metadata (lines beginning with ‘#’), variant information (one locus per row) and sample information (one sample per column). We adapt this format to include GWAS-specific metadata (Supplementary Table 1) and utilise the sample column to store variant-trait association data (Figure 1; Supplementary Table 2).

The VCF header is mandatory for defining fields throughout 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). Metadata define important characteristics of the GWAS including trait description(s), units and identifiers, genome build, chromosome lengths and number, number of variants, type of trait (continuous or binary), sample size and study identifier.

Each row of the file body contains a single variant position describing chromosome name, base-pair position, variant identifier (i.e. dbSNP identifier), reference (non-effect allele) and alternative/effect allele(s), quality and filter. The latter can be used to flag poor quality variants for exclusion in downstream analyses. Additionally, the INFO column is a flexible store for variant-level data such as population frequency, genomic annotations and functional effects. Finally, the sample column is used to store allele-trait association metrics: effect size, standard error and association P-value. Cohort variant frequency and genotype marker identifier are also stored in the sample field to accommodate multiple GWAS from different populations and/or genotyping technology in a single VCF file.

This format has a number of advantages over existing solutions. First, established VCF parsing libraries (HTSLIB 17 & HTSJDK 17) provide robust methods for handling complex variation such as multiallelic and insertion-deletion variants which are typically discarded during analyses. Second, the VCF provides format validation ensuring values are of the appropriate data type and without missing entries unless explicitly permitted. These features reduce parsing errors and prevent unexpected program operation. Third, file headers support embedding custom study/variant/trait-level metadata. Fourth, the VCF is well established and supported by existing tools providing a range of analytical functions. Finally, VCF can store individual or multiple GWAS traits in a single file which is beneficial for data integration and distribution.

Simulations of query performance suggest compressed VCF is substantially quicker than unindexed and uncompressed flat files for querying by genomic position. On average VCF was 16x faster to extract a single variant using chromosome position (0.08 seconds [95% CI 0.08, 0.08]) than unindexed text (1.29 seconds [95% CI 1.29, 1.30]) and 9x quicker using the dbSNP identifier (0.09 seconds [95% CI 0.09, 0.09] vs 0.81 seconds [95% 0.80, 0.82]). Using a 1Mb window of variants VCF was 46x quicker (0.11 seconds [95% CI 0.11, 0.11] vs 5.02 seconds [95% CI 4.99, 5.04]). Although querying on association P value was faster using unindexed text (7.18 seconds [95% CI 7.09, 7.26] vs 18.04 seconds [95% CI 17.92, 18.16]) VCF could be improved by using variant flags (i.e. in the INFO filed) to highlight records below prespecified thresholds if the exact value is unimportant. For example, all variants at genome-wide significance (P=5e-8) or a more relaxed threshold (e.g. P=5e-5). Alternatively, VCF files could be read entirely into memory (if possible) or loaded into a dedicated database such as GenomicsDB 18 which might offer better query performance.

To automate mapping tabular summary statistics to VCF, we developed open-source Python3 software (Gwas2VCF; Table 2). The application reads in metadata and variant-trait associations using a user-defined schema. During processing, variants are harmonised using a supplied reference genome file to ensure the non-effect allele matches the reference sequence enabling consistent inter-study comparisons. Insertion-deletion variants are subsequently left-aligned and trimmed using bcftools 17 to ensure consistent representation. Finally, the VCF is indexed using tabix 17 and rsidx 19 which enable rapid queries by genomic position and dbSNP identifier, respectively. We have developed a freely available web application providing a user-friendly interface for this implementation and encourage other centres to deploy their own instance (Table 2).

Once in GWAS-VCF, summary statistics can be read and queried using R or Python programming languages with our open-source libraries (Table 2). Further, gwasglue provides convenient R programming functions to automate preparation of genetic association data for downstream analysis (Table 2). Currently, methods exist for streamlining variant fine-mapping, colocalisation, MR 4 and data visualisation. New methods are being actively added and users may request new features via the repository issues page.

To encourage adoption, we made freely available over 10,000 complete GWAS summary statistics in VCF. These studies include a broad range of traits, diseases and molecular phenotypes initially collected as part of the MR Base platform 20.

A limitation of the GWAS-VCF is a lack of stable variant identifier for labelling individual variant-trait associations which is especially problematic at multiallelic loci where multiple association statistics are placed on a single VCF row. In version 1.0.0 of the GWAS-VCF specification we propose querying by chromosome and base-position or dbSNP identifier and filtering to retain the target variant, but this acknowledge this approach can be cumbersome. One solution is to store each alternative allele on a single row and use the VCF ID field to store a unique variant identifier. We put forward three suggestions to address this issue for the next version of the specification (Supplementary Table 3), but successful implementation will require consultation from all stakeholders.

Here we present a specification for GWAS summary statistics storage amenable to high-throughput analyses and robust data sharing. We implement open-source tools to convert summary statistics to VCF and libraries for reading or querying the format and integrating with existing analysis tools. Finally, we provide complete GWAS summary statistics for over 10,000 traits in GWAS-VCF. These resources enable convenient and efficient secondary analyses of summary statistics and support future tool development.

**Code availability**

Query performance evaluation source code available from GitHub (https://github.com/MRCIEU/gwas-vcf-performance) or pre-built image available from DockerHub (mrcieu/gwas-vcf-performance)

**Data availability**

Version 1.0.0 of the GWAS summary statistics VCF specification is available from: <https://github.com/MRCIEU/gwas-vcf-spec/releases/tag/1.0.0>

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

**Method**

**Specification**

The specification was developed through experience of collecting and harmonising GWAS summary data across two research centres at scale 20 and performing a range of representative high throughput analyses on these data (for example LD score regression 22, MR 23, genetic colocalisation analysis 24 and polygenic risk scores 25).

**Query performance simulation**

Densely imputed summary statistics (13,791,467 variants) for a large GWAS of body mass index data were obtained from Neale et al 26. The data were mapped to VCF using Gwas2VCF v1.1.1 and processed using bcftools v1.10 17 to remove multiallelic variants or records with missing dbSNP identifiers. A tabular (unindexed) file was prepared from the VCF to replicate a typical storage medium currently used for distributing summary statistics. Query runtime performance was compared between tabix and standard UNIX commands under the following conditions: single variant selection using dbSNP identifier or chromosome position, multi-variant selection by association P value (thresholds: P < 5e-8, 0.2, 0.4, 0.6, 0.8) or 1 Mb genomic interval. Tests were undertaken with 100 repetitions using VCF or unindexed text formats with and without GZIP compression on an Ubuntu v18.04 server with Intel Xeon(R) 2.0 Ghz processor. All comparisons were performed using singled thread operations and therefore differences in runtime performance were due to tool and/or file index usage.

**References**

1. Hou, L. & Zhao, H. A review of post-GWAS prioritization approaches. *Front. Genet.* **4**, 280 (2013).

2. Finucane, H. K. *et al.* Partitioning heritability by functional annotation using genome-wide association summary statistics. *Nat. Genet.* **47**, 1228–1235 (2015).

3. Visscher, P. M. *et al.* 10 Years of GWAS Discovery: Biology, Function, and Translation. *American Journal of Human Genetics* **101**, 5–22 (2017).

4. Smith, G. D. & Ebrahim, S. ‘Mendelian randomization’: Can genetic epidemiology contribute to understanding environmental determinants of disease? *International Journal of Epidemiology* (2003). doi:10.1093/ije/dyg070

5. Bulik-Sullivan, B. *et al.* LD score regression distinguishes confounding from polygenicity in genome-wide association studies. *Nat. Genet.* (2015). doi:10.1038/ng.3211

6. Yang, J., Zeng, J., Goddard, M. E., Wray, N. R. & Visscher, P. M. Concepts, estimation and interpretation of SNP-based heritability. *Nature Genetics* **49**, 1304–1310 (2017).

7. Buniello, A. *et al.* The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Res.* **47**, D1005–D1012 (2019).

8. Zhu, Z. *et al.* Integration of summary data from GWAS and eQTL studies predicts complex trait gene targets. *Nat. Genet.* **48**, 481–487 (2016).

9. Danecek, P. *et al.* The variant call format and VCFtools. *Bioinformatics* **27**, 2156–2158 (2011).

10. Purcell, S. *et al.* PLINK: A tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.* (2007). doi:10.1086/519795

11. Yang, J., Lee, S. H., Goddard, M. E. & Visscher, P. M. GCTA: A tool for genome-wide complex trait analysis. *Am. J. Hum. Genet.* **88**, 76–82 (2011).

12. Loh, P. R. *et al.* Efficient Bayesian mixed-model analysis increases association power in large cohorts. *Nat. Genet.* **47**, 284–290 (2015).

13. Zhou, X. & Stephens, M. Genome-wide efficient mixed-model analysis for association studies. *Nat. Genet.* **44**, 821–824 (2012).

14. Willer, C. J., Li, Y. & Abecasis, G. R. METAL: fast and efficient meta-analysis of genomewide association scans. *Bioinforma. Appl. NOTE* **26**, 2190–2191 (2010).

15. Hartwig, F. P., Davies, N. M., Hemani, G. & Smith, G. D. Two-sample Mendelian randomization: avoiding the downsides of a powerful, widely applicable but potentially fallible technique. *Int. J. Epidemiol.* 1717–1726 (2016). doi:10.1093/ije/dyx028

16. Home - SNP - NCBI. Available at: https://www.ncbi.nlm.nih.gov/snp/. (Accessed: 16th March 2020)

17. Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987–93 (2011).

18. GenomicsDB/GenomicsDB: Highly performant data storage in C++ for importing, querying and transforming variant data with Java/Spark. Used in gatk4. Available at: https://github.com/GenomicsDB/GenomicsDB. (Accessed: 25th February 2020)

19. bioforensics/rsidx: Library for indexing VCF files for random access searches by rsID. Available at: https://github.com/bioforensics/rsidx. (Accessed: 5th March 2020)

20. Hemani, G. *et al.* The MR-base platform supports systematic causal inference across the human phenome. *Elife* **7**, (2018).

21. Karczewski, K. J. *et al.* Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes. *bioRxiv* 531210 (2019). doi:10.1101/531210

22. Zheng, J. *et al.* Databases and ontologies LD Hub: a centralized database and web interface to perform LD score regression that maximizes the potential of summary level GWAS data for SNP heritability and genetic correlation analysis. *Bioinformatics* **33**, 272–279 (2017).

23. Hemani, G. *et al.* Automating Mendelian randomization through machine learning to construct a putative causal map of the human phenome. *bioRxiv* 173682. (2017). doi:10.1101/173682

24. Richardson, T. G., Hemani, G., Gaunt, T. R., Relton, C. L. & Davey Smith, G. A transcriptome-wide Mendelian randomization study to uncover tissue-dependent regulatory mechanisms across the human phenome. *Nat. Commun.* **11**, 1–11 (2020).

25. Richardson, T. G., Harrison, S., Hemani, G. & Smith, G. D. An atlas of polygenic risk score associations to highlight putative causal relationships across the human phenome. *Elife* **8**, (2019).

26. UK Biobank — Neale lab. Available at: http://www.nealelab.is/uk-biobank/. (Accessed: 25th February 2020)

27. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinforma. Appl. NOTE* **25**, 2078–2079 (2009).

28. Obenchain, V. *et al.* Sequence analysis VariantAnnotation: a Bioconductor package for exploration and annotation of genetic variants. **30**, 2076–2078 (2014).

29. Gentleman, R. C. *et al.* *Open Access Bioconductor: open software development for computational biology and bioinformatics*. *Genome Biology* **5**, (2004).

30. Huber, W. *et al.* Orchestrating high-throughput genomic analysis with Bioconductor. *Nat. Methods* **12**, 115–121 (2015).

31. Bioconductor - Home. Available at: https://www.bioconductor.org/. (Accessed: 27th March 2020)

32. pysam-developers/pysam: Pysam is a Python module for reading and manipulating SAM/BAM/VCF/BCF files. It’s a lightweight wrapper of the htslib C-API, the same one that powers samtools, bcftools, and tabix. Available at: https://github.com/pysam-developers/pysam. (Accessed: 10th March 2020)

33. McKenna, A. *et al.* The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* (2010). doi:10.1101/gr.107524.110

34. 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. Available at: https://github.com/broadinstitute/picard. (Accessed: 25th February 2020)

35. Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinforma. Appl. NOTE* **26**, 841–842 (2010).

36. Voss, K., Gentry, J. & Auwera, G. Van Der. GATK4 + WDL + Cromwell. *F1000Research* **6**, 4 (2017).

**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].

E.M. and S.J.A. are supported by the JPB foundation and by the National Institute of Health (U01AG052411 and U01AG058635; principal investigator Alison Goate).

**Author contributions**

All authors contributed the manuscript and storage format specification. G.H. and E.M. designed the research. M.L. and G.H. wrote software packages and performed query performance simulations. B.E. and G.H. prepared the GWAS data.

**Competing interest**

None.

**Correspondence**

Matthew Lyon ([matt.lyon@bristol.ac.uk](mailto:matt.lyon@bristol.ac.uk))

Population Health Sciences

Bristol Medical School

University of Bristol

Oakfield House

Oakfield Grove

Bristol

BS8 2BN

Table 1. Requirements for a summary statistics storage format and solutions offered by the VCF

|  |  |
| --- | --- |
| Requirement | Solution using the variant call format |
| Human readable and easy to parse | Easily read with any text viewer. Mature open-source parsing libraries are available (HTSLIB 27 and HTSJDK 27) which are implemented in most modern programming languages, for example: VariantAnnotation 28 R-package is available from Bioconductor 29–31 and python package pysam 32. Bcftools 17 also provides user-friendly functionality from the command line. |
| Unambiguous interpretation of the data | Data field descriptions, value types and number of values 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. Established parsing libraries 27 have routines for handling complex variants. Using this approach alternative allele(s) are always the 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 reference FASTA. GATK 33 ValidateVariants can verify file format validity and compare reference allele information against the corresponding human genome reference sequence. |
| 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. The supplement contains essential fields and reserved keys. |
| Capacity to store metadata about the study and trait(s) | Each GWAS trait has a row in the file header to store phenotype description and units, outcome type (binary or continuous), trait unique identifier (e.g., EFO term, Ensembl Gene ID [eQTLs] or UniProt protein accession [pQTLs]), publication identifier and population ethnicity. |
| Allows multiple traits or studies to be stored together | The VCF SAMPLE column was chosen to store GWAS association metrics to allow for storage of multiple traits in a single VCF file, or as individual files if desired. |
| Rapid querying by variant identifier, genomic position interval or GWAS summary data value | The file is sorted karyotypically and indexed by chromosome position using tabix to enable fast queries by genomic position. Secondary indexing on dbSNP identifier is also provided using rsidx 19. Refer below for performance comparison of indexed VCF files and standard UNIX tools. |
| File compression | VCF files may be compressed with block GZIP 17 or converted to a binary call file which is a binary VCF companion format 17. |
| Readable by existing open-source tools | A large number of tools support VCF files including: GATK 33, Picard 34, bcftools 17, bedtools 35, vcftools 9 and plink 10. Bcftools 17 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 range-requests which extracts file segments without transferring the whole file. This enables rapid streaming of queries over the internet. For high-throughput and distributed storage and querying, VCF files can be easily imported into GenomicsDB 18. |

GWAS, genome-wide association study. dbSNP, database of single-nucleotide polymorphisms. HTSLIB, high-throughput sequencing data library. HTSJDK, high-throughput sequencing data java development kit. GATK, genome-analysis toolkit. dbSNP, single nucleotide polymorphism database. EFO, Experimental Factor Ontology. eQTL, expression quantitative trait loci. pQTL, protein quantitative trait loci.

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

|  |  |  |  |
| --- | --- | --- | --- |
| Program | Purpose | Implementation | Source code link |
| gwas2vcf | Mapping tab separated GWAS summary statistics and EBI format to VCF | Python3 (Docker) | <https://github.com/mrcieu/gwas2vcf> |
| gwas2vcfweb <http://64.227.44.193> | Front-end and queue schedular for gwas2vcf | Python3, Cromwell36  (Docker) | <https://github.com/mrcieu/gwas2vcfweb> |
| R/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> |
| R/gwasglue | Library for processing GWAS summary statistics ready for secondary analysis | R | <https://github.com/mrcieu/gwasglue> |
| LD Score Regression 5 (patch) | Estimating genetic correlation and heritability | Python | <http://github.com/explodecomputer/ldsc> |

GWAS, genome-wide association study. LD, linkage disequilibrium. VCF, variant call format. EBI, European Bioinformatics Institute.

Figure 1. VCF adapted to store GWAS summary statistics



Variant call file format for storing GWAS summary statistics. The file contains study and trait(s) metadata, variant-level content and variant-trait association statistics. Each field is defined in the file header including variable type and number of values. The format can accommodate one or multiple traits per file as required.

Figure 2. Performance comparison for querying summary statistics in plain text and VCF



Mean query time (log milliseconds [lower is quicker]; repetitions n=100) to extract either: a single variant using the chromosome position or dbSNP identifier or multiple variants using a 1 Mb interval or association P value. AWK, grep, bcftools 17 and rsidx 19 were evaluated using uncompressed and GZIP/BGZIP 17 compressed unindexed text and VCF. Error bars represent the 95% confidence interval.

Supplementary Table 1. VCF study and trait fields for GWAS summary statistics

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| Trait | String | Phenotype description |
| Unit | String | Phenotype units |
| PMID | Integer | PubMed publication identifier |
| Population | String | Participant ethnicity |
| TotalControls\* | Integer | Total number of controls in the association study (or total sample size if continuous outcome) |
| TotalCases | Integer | Total number of cases in the association study |
| StudyType\* | String | Type of GWAS study [Continuous or Binary] |
| TotalVariants\* | Integer | Total number of variants in input |
| VariantsNotRead\* | Integer | Number of variants that could not be read |
| HarmonisedVariants\* | Integer | Total number of harmonised variants |
| VariantsNotHarmonised\* | Integer | Total number of variants that could not be harmonised |
| SwitchedAlleles\* | Integer | Total number of variants strand switched |

Reserved fields for storing essential metadata in the VCF header. PMID, PubMed identifier. \* Required fields.

Supplementary Table 2. VCF variant fields for GWAS summary statistics

|  |  |  |
| --- | --- | --- |
| Field | Type | Description |
| ES\* | Float | Effect size estimate relative to the alternative allele |
| SE\* | Float | Standard error of effect size estimate |
| LP\* | Float | -log10 p-value for effect estimate |
| AF | Float | Alternate allele frequency in the association study |
| SS | Float | Sample size used to estimate genetic effect |
| EZ | Float | Z-score provided if it was used to derive the ES and SE fields |
| SI | Float | Accuracy score of summary data imputation |
| NC | Float | Number of cases used to estimate genetic effect |
| ID | String | Study variant identifier |

Reserved fields for storing essential variant contextual data in the VCF sample column. ES, effect size (beta or log-odds). SE, standard error. LP, -log10 P value. AF, allele frequency. SS, sample size. EZ, effect Z-score. SI, summary imputation. NC, number of cases. ID, identifier. \* Required fields.

Supplementary Table 3. Proposed variant identifiers for the VCF file ID column

|  |  |  |
| --- | --- | --- |
| VCF row identifier (ID column) | Advantages | Disadvantages |
| dbSNP rsID (i.e. rs376272854) | * No duplication of information already in the row * Rsidx provides fast rsID queries * Widely used * Short length * Compatibility with existing tools (rsID is encouraged by VCF v4.2 specification) | * Complex to manipulate multiallelic rows * Does not distinguish between multiple alternative alleles * Multiple rsIDs can point to a single position (e.g. new dbSNP entries awaiting merge with existing records) |
| HGVS DNA nomenclature (i.e. chr2:g.84918761\_84918811del) | * Unique identifier for every variant * Supports one variant per row in the VCF which is easier to parse * Short insertion-deletion encoding * Known format | * Duplicates information already stored in the row * Not stable between genome builds * Comparing between builds is difficult * Not widely used for GWAS |
| Concatenation of chromosome, position and alleles (i.e. chr2:84918760:  CCCAACCCTGCTGTCAT  AATGCATAAGCAGCCAC  AGACAGTAAGTGAATGAA:C) | * Unique identifier for every variant * Supports one variant per row in the VCF which is easier to parse * Known format | * Duplicates information already stored in the row * Not stable between genome builds * Comparing between builds is difficult * Long insertion-deletion coding |
| Concatenation of chromosome, position and alleles using MD5 hash to shorten long alleles (i.e. chr2:84918760-7c43e7284b58ba06e  7438bff62376edf:C) | * Unique (almost) identifier for every variant * Supports one variant per row in the VCF which is easier to parse * Short insertion-deletion coding | * Duplicates information already stored in the row * Not stable between genome builds * Comparing between builds is difficult * Cannot reverse hash without database * Not widely used * Small chance of a hash collision |

GWAS, genome-wide association study. VCF, variant call format. Rsidx, file index using the dbSNP identifier. MD5, message-digest algorithm. HGVS, Human Genome Variation Society.