GWASLab: a Python package for processing and visualizing GWAS summary statistics

Yunye He[1,*], Masaru Koido[1], Yuka Shimmori[1], Yoichiro Kamatani[1,*]

[1] Laboratory of Complex Trait Genomics, Graduate School of Frontier Sciences, The University of Tokyo, Tokyo, Japan

*Corresponding authors: Yunye He, yunyehe.ctg@gmail.com; Yoichiro Kamatani, kamatani.yoichiro@edu.k.u-tokyo.ac.jp.

Keywords: GWAS, summary statistics, visualization, quality control, Python

Abstract

Summary

GWASLab is a comprehensive Python toolkit for processing and visualizing summary statistics (SumStats) derived from genome-wide association studies (GWAS). GWASLab provides functions including quality control (QC) of statistics, standardization of chromosome and allele notation, variant normalization, harmonization for meta-analysis, and data visualization. Modular implementation of functions allows users to customize their own pipelines for utilizing SumStats. An expandable formatting library and standalone utilities persistently ensure seamless compatibility with many post-GWAS tools.

Availability and implementation

GWASLab is implemented in Python; the source code is publicly and freely available at https://github.com/Cloufield/gwaslab, and the documentation is available at https://cloufield.github.io/gwaslab/.

1. Introduction

GWAS summary statistics (SumStats) are accumulating at a rapid speed. As of April 2023, SumStats for more than 6,000 publications are publicly available on GWAS Catalog (Buniello *et al.*, 2019). The availability of SumStats potentiates a wide range of post-GWAS analyses, such as LD Score regression (Bulik-Sullivan *et al.*, 2015), genome-wide meta-analysis, Mendelian randomization, and polygenic risk scores. SumStats sharing greatly enhanced research in genetics, and it was estimated that SumStats sharing led to around 75% more citations (Reales and Wallace, 2022).

Despite the efforts to develop a standard SumStats format (Hayhurst *et al.*, 2022), the large number of existing unprocessed SumStats remain challenges for data sharing and efficient reuse. Additionally, extensive discrepancies exist between the formats GWAS software generates and the required formats for each post-GWAS software. The missingness of certain information (such as rsID), the inconsistent representation of statistics, and the ambiguity of column headers (such as the headers for effect allele and non-effect allele) often hinder the direct reuse of SumStats, especially for beginners, and were error-prone during data and format conversions without careful reading the manual. Furthermore, unexpected failure in processing SumStats can lead to adverse impacts on downstream analyses, which could be avoided by checking detailed log messages.

Existing tools for handling or visualizing GWAS SumStats, mostly implemented in R (Murphy *et al.*, 2021; Yin *et al.*, 2021; Turner, 2018), focused on specific functionalities such as data munging or plotting. With the rapid increase in publicly available SumStats and post-GWAS tools, there is also an increasing need for a comprehensive and customizable tool that integrates functions for handling and visualizing GWAS summary statistics, which can serve as a bridge linking unprocessed SumStats to post-GWAS analysis tools seamlessly. Here, we present GWASLab, a user-friendly Python package for the manipulation and visualization of GWAS summary statistics. This package provides functions including quality control (QC) of statistics, standardization of chromosome and allele notation, variant normalization, harmonization for meta-analysis, and data visualization. We developed a summary statistics format conversion library along with standalone utilities, which ensure seamless compatibility with a wide range of post-GWAS tools. Moreover, the implementation of a logging system provided detailed reports on each process applied, which increased the interpretability and the reusability of SumStats.

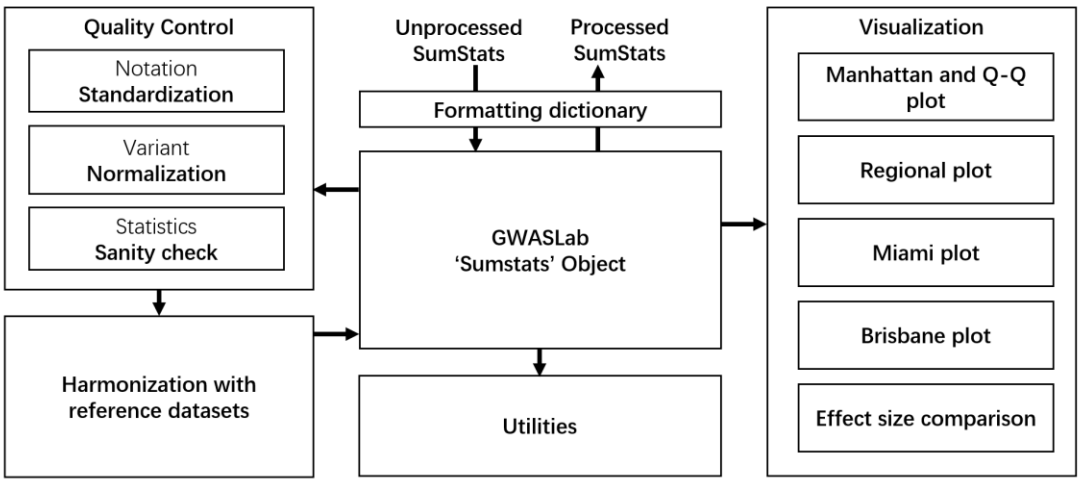

Fig. 1 Overview of GWASLab package design. Q-Q plot, quantile-quantile plot. SumStats, Summary statistics. Miami plot, mirrored Manhattan plots for two traits. Brisbane plot, Manhattan-like plot showing genomic density of independent genetic associations.

2. Implementation

GWASLab was designed based on the following principles: (1) to remove ambiguity and minimize uncertainty in SumStats by specifying column definition, standardizing notations, and aligning with reference data; (2) to provide flexible, customizable, and interpretable functions for each manipulation so that users can customize their own pipeline when the pre-defined pipeline does not suit for their SumStats; (3) to support detailed logging for every step of manipulation so that the pipeline is traceable and replicable; and (4) to provide compatibility and extensibility with existing tools and future ones.

GWASLab has been developed as an open-source Python package. Sumstats will be formatted and stored in the 'GWASLab.Sumstats' object. Functions to manipulate and visualize the summary statistics were implemented as methods of the 'GWASLab.Sumstats' object, enabling the construction of interpretable, flexible, and customizable SumStats processing pipelines.

GWASLab is composed of five main function groups (Fig.1): (1) SumStats formatting; (2) standardization and quality control without references, (3) harmonization with references, (4) visualization, and (5) standalone utilities. This package takes advantage of commonly used Python packages and can be readily integrated into any Python-based downstream analysis tools or pipelines. Additionally, we implemented a logging system along with an original status code system to log the manipulations applied to the SumStats and trace the status of each variant.

Columns of SumStats data in GWASLab consist of two parts, core columns, and additional columns. Core columns comprise key information for variants and statistics generated from GWAS, including rsID, SNPID (preferably in the format of chromosome: position: reference allele: alternative allele), CHR (chromosome number), POS (base-pair position), EA (effect allele), NEA (non-effect allele), EAF (effect allele frequency), BETA (effect size), SE (standard error of effect size), N (sample size), OR (odds ratio), OR_95L (lower bound of 95% confidence interval), OR_95U (upper bound of 95% confidence interval), P (p-value), MLOG10P (-log10(P-value)), Z (z score), CHISQ (chi-square statistic), DIRECTIONS (effect size directions). Additional columns consist of optional information for the variants such as annotation, providing extensibility for formatting or customized filtering.

3. Main Usage

3.1. Seamlessly importing and formatting files

Using a curated and expandable format conversion library ('formatbook': https://github.com/Cloufield/formatbook), GWASLab can read SumStats generated by widely used GWAS software such as PLINK (Purcell *et al.*, 2007), SAIGE (Zhou *et al.*, 2018) and REGENIE (Mbatchou *et al.*, 2020) and format SumStats to software-specific or widely accepted formats in line with sharing standards such as GWAS VCF (MacArthur *et al.*, 2021; Lyon *et al.*, 2021) and GWAS-SSF (Hayhurst *et al.*, 2022). Users can also load customized

formats by explicitly specifying the columns.

3.2. Standardization, normalization, quality control, and harmonization

GWASLab provides functions to standardize the notations and data types of variant ID, chromosomes, base pair positions, and alleles. After standardization, variants will be checked for normalization to ensure they are left aligned and parsimonious (Tan *et al.*, 2015). For statistics, GWASLab can filter extreme values, remove missing or duplicated records, and perform sanity checks as implemented in existing tools (Murphy *et al.*, 2021; Matushyn *et al.*, 2022). GWASLab provides functions for SumStats harmonization with references, which include allele alignment with a reference genome, converting genome coordinates (liftover) using appropriate chain file, strand checking of palindromic SNPs, and annotating rsID using reference files downloaded from commonly used sources such as 1000 Genomes Project (1000 Genomes Project Consortium *et al.*, 2015) and dbSNP (Sherry *et al.*, 2001), or self-prepared files for quick annotation.

3.3. Highly customized visualization and other utilities

Visualization functions were integrated into the GWASLab framework. GWASLab can create Q-Q plots, Manhattan plots, Miami plots (Winkler *et al.*, 2015), Regional plots, and Brisbane plots (Yengo *et al.*, 2022). GWASLab also supports effect size comparison, data conversion, lead variant extraction, and position-based novel loci determination (Zhou *et al.*, 2022). For example, users can specify an EFO ID (Malone *et al.*, 2010), and GWASLab will extract known associations for the trait through GWAS Catalog API (Buniello *et al.*, 2019), and compare the base pair distances between known loci and lead variants in user-provided SumStats, allowing users to check whether the known loci are potentially novel based on distances.

Acknowledgements

This work was supported by the Japan Agency for Medical Research and Development (AMED) under grant number JP19km0405215 (to Y.K.), JP23tm0624002 to Y.K. (to Y.K.) and JP223fa627011 (to Y.K.).

References

1000 Genomes Project Consortium *et al.* (2015) A global reference for human genetic variation. *Nature*, **526**, 68–74.

Bulik-Sullivan,B.K. *et al.* (2015) LD Score regression distinguishes confounding from polygenicity in genome-wide association studies. *Nat Genet*, **47**, 291–295.

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

Hayhurst,J. *et al.* (2022) A community driven GWAS summary statistics standard. 2022.07.15.500230.

Lyon,M.S. *et al.* (2021) The variant call format provides efficient and robust storage of GWAS summary statistics. *Genome Biology*, **22**, 32.

MacArthur,J.A.L. *et al.* (2021) Workshop proceedings: GWAS summary statistics standards and sharing. *Cell Genomics*, **1**, 100004.

Malone,J. *et al.* (2010) Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics*, **26**, 1112–1118.

Matushyn,M. *et al.* (2022) SumStatsRehab: an efficient algorithm for GWAS summary statistics assessment and restoration. *BMC Bioinformatics*, **23**, 443.

Mbatchou,J. *et al.* (2020) Computationally efficient whole genome regression for quantitative and binary traits. *Nature Genetics*.

Murphy,A.E. *et al.* (2021) MungeSumstats: a Bioconductor package for the standardization and quality control of many GWAS summary statistics. *Bioinformatics*, **37**, 4593–4596.

Purcell,S. *et al.* (2007) PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am J Hum Genet*, **81**, 559–575.

Reales,G. and Wallace,C. (2022) Sharing GWAS summary statistics results in more citations: evidence from the GWAS catalog. 2022.09.27.509657.

Sherry,S.T. *et al.* (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res*, **29**, 308–311.

Tan,A. *et al.* (2015) Unified representation of genetic variants. *Bioinformatics*, **31**, 2202–2204.

Turner,S.D. (2018) qqman: an R package for visualizing GWAS results using Q-Q and manhattan plots. *Journal of Open Source Software*, **3**, 731.

Winkler,T.W. *et al.* (2015) EasyStrata: evaluation and visualization of stratified genome-wide association meta-analysis data. *Bioinformatics*, **31**, 259–261.

Yengo,L. *et al.* (2022) A saturated map of common genetic variants associated with human height. *Nature*, **610**, 704–712.

Yin,L. *et al.* (2021) rMVP: A Memory-efficient, Visualization-enhanced, and Parallel-accelerated Tool for Genome-wide Association Study. *Genomics Proteomics Bioinformatics*, **19**, 619–628.
Zhou,W. *et al.* (2018) Efficiently controlling for case-control imbalance and sample relatedness in large-scale genetic association studies. *Nat Genet*, **50**, 1335–1341.
Zhou,W. *et al.* (2022) Global Biobank Meta-analysis Initiative: Powering genetic discovery across human disease. *Cell Genom*, **2**, 100192.