vcf2gwas - PYTHON API FOR COMPREHENSIVE GWAS ANALYSIS USING GEMMA

A PREPRINT

Frank Vogt *
Department of Molecular Biology
Max Planck Institute for Developmental Biology
Tuebingen, Germany 72076

Gautam Shirsekar †
Department of Molecular Biology
Max Planck Institute for Developmental Biology
Tuebingen, Germany 72076

Detlef Weigel
Department of Molecular Biology
Max Planck Institute for Developmental Biology
Tuebingen, Germany 72076

May 29, 2021

ABSTRACT

We present a new software package *vcf2gwas* to perform reproducible genome-wide association studies (GWAS). *vcf2gwas* is a Python API for bcftools, PLINK and GEMMA. A traditional GWAS workflow requires the user to edit and format the genotype information from commonly used Variant Call Format (VCF) file and phenotype information, before running the analysis in the software of choice. Post-processing steps involve summarizing and visualizing the analysis results. This workflow requires a user to utilize the command-line, manual text-editing and knowledge of one or more programming/scripting languages which can prove to be time-consuming especially when analyzing multiple phenotypes. Our package provides a convenient pipeline performing all of these steps, reducing the GWAS workflow to a single command-line input without the need to edit or format the VCF file beforehand or installing and using additional software. In addition, features like reducing the dimensionality of the phenotypic space and performing analyses on the reduced dimensions or comparing the significant variants from the results to specific genes/regions of interest are implemented. By integrating different tools to perform GWAS under one workflow, the package ensures reproducible GWAS while reducing the user efforts significantly.
Availability and implementation: The source code of *vcf2gwas* is available under the GNU General Public License. The package can be easily installed using conda. Installation instructions and a manual including tutorials can be accessed on the package website at `https://github.com/frankvogt/vcf2gwas`

1 Introduction

Genome-wide association study (GWAS) has been proven to be an extremely useful tool to find an association between genetic variants, typically single-nucleotide polymorphisms (SNPs) with a given trait in many organisms. To carry out a GWAS, the traits under investigation are analyzed by recording the phenotypes of different individuals and sequencing their DNA. After the SNPs are read from the genotype information, these genetic variants are tested for the association with the phenotypes using various computational methods implemented in a wide range of software. As a result, the likelihood of each SNP to be associated with the trait is calculated and subsequently used to identify SNPs with significant association.

*fvogt@tuebingen.mpg.de
†gshirsekar@tuebingen.mpg.de

While there are many algorithms implemented in various softwares to perform the association analysis, Genome-wide Efficient Mixed Model Association (GEMMA) ([1]) stands out because of its versatility and efficiency in handling large scale data. GEMMA can fit a univariate linear mixed model ([1]), a multivariate mixed model ([2]), and a Bayesian sparse linear mixed model ([3]) for testing marker associations with a trait of interest in different organisms.

Although GEMMA has a very straightforward command-line interface to carry out the actual association analysis, it requires users to first install necessary softwares with required dependencies on their machines. Subsequently, users need to prepare the inputs (genotype and phenotypes) in a proper format before execution of GEMMA. Similarly, the outputs generated need post-processing of the results for better interpretation and presentation. The entire workflow can be overwhelming especially for inexperienced users. Briefly, this workflow starts with the genotype information in a VCF file. This file has to be converted to the PLINK[4] BED format with the phenotype information that needs to be manually edited into associated .FAM file. Once the analysis is complete the user is left with the outputs which need to be summarized and plotted. If multiple phenotypes are to be analyzed, the analysis has to be repeated for every phenotype. Thus, performing GWAS in this way using GEMMA is very time-consuming and makes it very challenging to conduct reproducible research while keeping all the analyses well-organized.

The *vcf2gwas* package aims to facilitate performing a GWAS with GEMMA by automating all the phases, beginning with the installation of all the required softwares, to input preparations, to carrying out the analysis, and finally to processing the results. This reduces overhead for the user to a minimum. *vcf2gwas* is especially helpful when analyzing large numbers of phenotypes or different sets of individuals because it can perform the analyses in parallel. Additionally, the package offers features like analyzing reduced phenotypic space and comparing the resulting relevant SNPs to genes or regions of interest. But most importantly, *vcf2gwas* is easily installable as a conda package and therefore makes it possible to reproduce each GWAS on any compatible machine.

2 Implementation

The *vcf2gwas* package is implemented using the Python programming language. It contains multiple Python3 scripts with all the functions required for the program to execute the analysis. The core functionality of the package consists of the ability to perform automated GWAS analysis with a single command-line input, requiring minimal overhead by the user. It has multiple wrapper functions calling bcftools ([5]), PLINK ([4]), and GEMMA ([1]) from Python3 as well as complementary Python3 functions performing both pre- and post-analysis tasks, all arranged by the pipeline scripts. Pre-analysis includes functions for formatting, trimming and filtering the input files for the association analysis with GEMMA. The different analysis modes of GEMMA can be utilized and are also specified with the command-line input. For the post-analysis, additional functions are used to summarize and visualize the resulting data and perform additional operations specified by the command-line options. *vcf2gwas* uses Python libraries numpy[6], matplotlib[7], pandas[8] for preparing input/output files and plotting while scikit-learn[9] and umap-learn libraries are used for dimensionality reduction.

3 Example usage

Performing a GWAS using GEMMA needs a user to execute a number of steps. First, the genotype information (usually available as VCF file) has to be converted to the PLINK BED format. Second, phenotype information must be provided in a specific format as well, for e.g. by manually editing FAM files. Furthermore, the user has to trim both the genotype and phenotype information to contain the matching individuals in correct order. And finally, additional analysis such as visualizing and summarizing the results requires the user to utilize additional software or programming languages. In addition, analyzing more than one phenotype results in re-running GEMMA for each phenotype.

In contrast, by running *vcf2gwas*, the pipeline will perform all of the above steps in a fully automated fashion. The input files will be trimmed, filtered and converted to the format required by GEMMA. After the analysis is executed, the results will be summarized and visualized by drawing publication-ready Manhattan (Figure 1) and QQ plots; optionally additional operations as explained in the next section will be carried out. A basic bash command example to analyze a single phenotype using GEMMA's linear mixed model providing only the VCF and phenotype file is:

```
$ vcf2gwas -v <input.vcf> -pf <inputpheno.csv> -p 1 -lmm
```

Internally, the pipeline converts and formats the input files utilizing bcftools and PLINK while the post-analysis steps are performed using custom Python functions. We have presented an example Manhattan plot of a GWA analysis performed with *vcf2gwas* in figure 1.

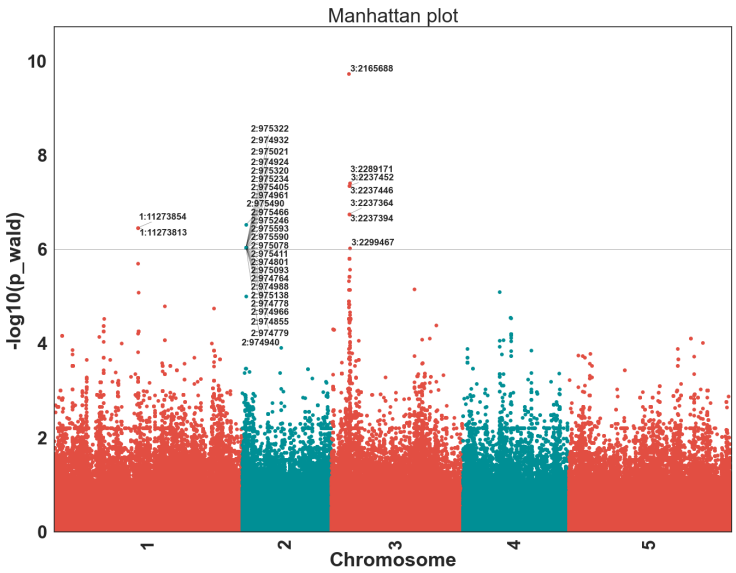

Figure 1: **Manhattan plot produced by *vcf2gwas* on *avrRpm1* recognition in *Arabidopsis thaliana***.
Linear mixed model analysis on a hypersensitive response phenotype observed in 58 *A. thaliana* host lines in response to *Pseudomonas syringe* expressing *avrRpm1* gene. The most significant SNP is 700 bp upstream of the *A. thaliana Rpm1* resistance gene. Description of the original experiment can be found at `https://arapheno.1001genomes.org/phenotype/17/`

4 Additional functionality

Beyond the basic necessities of summarizing and plotting the results of the GEMMA analysis, it might be desirable to perform the GWAS in reduced phenotypic space to account for underlying phenotypic structures or to compare the resulting relevant SNPs to genes/regions of interest. Below are the most important additional features implemented in *vcf2gwas*:

- Dimensionality reduction via PCA or UMAP can be performed on phenotypes and used for analysis.
- Relevant SNPs can be compared to genes/regions of interest listed in additionally specified file.
- *vcf2gwas* is able to analyze several input files with different sets of individuals and multiple phenotypes in a efficient manner due to parallelization, saving the user a lot of time compared to standard GWAS procedure.
- Results are reproducible on any Unix-based machine (macOS / Linux).

Acknowledgements

We thank Hajk-Georg Drost and Max Collenberg for useful comments and suggestions during the software development.

Funding

This work has been supported by Max Planck Society.

References

[1] Xiang Zhou and Matthew Stephens. Genome-wide efficient mixed-model analysis for association studies. *Nat. Genet.*, 44(7):821–824, June 2012.

[2] Xiang Zhou and Matthew Stephens. Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nat. Methods*, 11(4):407–409, April 2014.

[3] Xiang Zhou, Peter Carbonetto, and Matthew Stephens. Polygenic modeling with bayesian sparse linear mixed models. *PLoS Genet.*, 9(2):e1003264, February 2013.

[4] Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel A R Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul I W de Bakker, Mark J Daly, and Pak C Sham. PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, 81(3):559–575, September 2007.

[5] Petr Danecek, James K Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A McCarthy, Robert M Davies, and Heng Li. Twelve years of SAMtools and BCFtools. *Gigascience*, 10(2), February 2021.

[6] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[7] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[8] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.