Date: 04 April 2022
SSF Project Report

Tool development

Odyssey 2.1.1: Imputation of genomic data

Alina Orozco[1]

[1]Bioinformatics Core Facility, Sahlgrenska Academy, University of Gothenburg, Box 115, 405 30, Sweden.

Summary
Odyssey is a semi-autonomous workflow designed for the preparation, phasing and imputation of genomic data. The pipeline was originally published by authors Eller, Ryan J., Sarath C. Janga, and Susan Walsh. "Odyssey: a semi-automated pipeline for phasing, imputation, and analysis of genome-wide genetic data." BMC bioinformatics 20.1 (2019). Odyssey 2.1.1 is modified to run directly from the data folder on an HPC system or designated file system that contains the data of interest which can be specified in the configuration file (Setting.conf). Additionally, the option for imputation has been narrowed to using Minimac2 due to speed differences compared to Impute4. Other functionalities of Odyssey remain the same and can be reviewed in the modified documentation materials.

Contact: alina.orozco@gu.se

1 Introduction

With the growing accessibility of genome-wide data, there is a growing need for widely accessible genome-wide association tools for researchers, physicians and biologists that may not be possess the programming skills or knowledge of all the available tool options for processing the data. The usual steps for processing Genome Wide Associations studies include genome imputation, phasing, admixture resolution, result condensation and visualization.

While there are a number of Imputation tools available from online servers (Michigan Imputation Server and Sanger Imputation Server, https://imputationserver.sph.umich.edu/index.html) to offline solutions such as Michigan Imputation Server Docker (Das, 2016) and Genipe (Lemieux Perreault, 206), the advantage of Odyssey (Eller, 2019) over these other solutions is that the Odyssey pipeline does not limit the flexibility of reference panel choice in the way that online solutions might and does not require as much configuration as other offline sources. Genipe while providing the most similar imputation functions to Odyssey does require manual installation and configurations of dependencies.

The original Odyssey pipeline was developed to streamline the process of genome imputation admixture resolution and genome wide association analyses. Though the original pipeline structure was largely maintained, the need to locate large data sets in the pipeline folders themselves was identified as an area for improvement. Additionally, the removal of IMPUTE4/IMPUTE2 (Howie, 2011) was identified as an area for simplification for the program due to the superior performance of Minimac4 (https://genome.sph.umich.edu/wiki/Minimac4) compared to IMPUTE software. One tool comparison writes of these tools: “IMPUTE2 has the lowest concordance for the minor allele genotypes, while Beagle4.1, Beagle5.1, IMPUTE4, minimac3 and minimac4 have similar results with the minimac programs at the top.” (Stahl, 2021)

The main motivation for these modifications is for scenarios where there are a number of separate projects for which it would be cumbersome to move large amounts of data.

2 Implementation

2.1 Dependencies

This pipeline is designed to be run with Singularity (Singularity Developers, 2021) already installed on the system. Through the implementation of a singularity container, there are less dependencies that need to be available on the host system, however, bash command line tools such as tar and gzip should be readily installed and available to the pipeline.

2.1 Setup

Singularity was installed on a High Performance Cluster (HPC) at Gothenburg University. The tools and environment settings for Odyssey are included in a singularity build file *OdysseyContainer1.def.*

2.1 Pipeline

Odyssey integrates PLINK (Purcell, 2007), SHAPEIT (Delaneau, 2013), Eagle (Loh, P-R, 2016), Minimac4 and some R packages (data.table (Dowle, 2021), qqman (Turner, 2018), stringr (Wickham, 2019), manhattanly (Bhatnagar, 2021)) to create a modular workflow controlled via a single configuration file. The bash scripts are applied similarly to the original design where the separate bash scripts can be run together on an HPC system. The scripts are run in numerical order according to the script numbering starting with script *0_DataPrepCleanup.sh* followed by *1_ImputeProjectSetup.sh* then *2_PhaseScriptMaker.sh*, *3a_ImputeScriptMaker.sh*, *3b_ConcatConver.sh* and finally *4_AutomatePlink*. There are other scripts included for admixture analysis, these remain largely unchanged however, modifications allow this processing step to be run within the folder where data is located.

2.2 How it runs

Starting with the *Settings.conf* file, the user first specifies the **Configuration Variables** where the user indicates which method was used to set up Odyssey. "One" when Odyssey was set up to incorporate Singularity or "Two" when Odyssey was set up independently of Singularity.

The user then sets **Core Variables** including the location of the singularity installation as the 'singularityBinPath', the 'dataPath' which is the full path location of the data the user would like to analyze and the HPC Control Variables where the user can indicate if they want to run the pipeline on a HPC [HPS_Submit=(T/F)], their email and whether to use Lustre Stripped Directories [LustreStrip=(T/F)].

Users will also select in this file the options they would like to use ie., Shapeit2, minimac, eagle2. Other user inputs include the Project name (BaseName), whether to visualize data and whether to have X11 interface. There are a number of other input options that are detailed further in the OdysseyTutorialv6 document, however, an overview of inputs and outputs is shown in Figure 1.

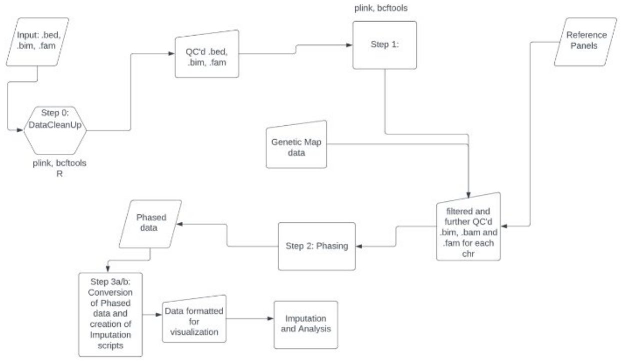

Figure 1. Overview of inputs and outputs.

3 Results

All outputs are created within the folder that the data for the pipeline is located.

From the Data Cleanup step 0 the output includes a corrected .bim, .bed and .fam files to be used in the Imputation step 1, so these output are moved to both TargetData (within the folder where the data is located) as well as a Temp directory. Step 1 further cleans the data from Step 0 by filtering for genotype and individual missingness, minor allele frequency, and Hardy-Weinberg equilibrium. The resulting .bed, .bam, .fam files can be found in a new folder that is created within the user data folder and that is named according to the ´BaseName´ specified within the *Settings.Conf* file. The Phasing Step 2 entails the phasing of the chromosomal separated data that has gone through the previously mentioned steps by Shapeit2, which is based on 1000 Genome Haplotypes from Phase 3 (GRCh37-hg19). Step 3a and 3b work with the phased and imputed data created in the previous section. Finally, step 4 performs the imputation analysis and further visualization.

4 Documentation

The original documentation is published on Github [https://github.com/Orion1618/Odyssey/blob/master/3a_ImputeScriptMaker.sh] and the updated work is published as a fork of the master repository [https://github.com/AlinaO311/Odyssey].

5 Conclusions

A future improvement could be implementing another imputation option such as the previously mentioned Beagle program. Another improvement might be to resolve for unusual data formats or in pgen format. Though more functionality was added for Plink2, it could be that a complete replacement of plink commands to plink 2 would improve functionality further.

Acknowledgements

The author thanks the Bioinformatics Core Facility at the University of Gothenburg for providing computational resources for data analysis and storage. Special thanks to Tara Stanne (University of Gothenburg) for kindly providing datasets for case testing, and Björn Andersson (Bioinformatics Core Facility) for valuable analytical input during the development of the pipeline.

Funding

This study was supported by the Swedish Foundation for Strategic Research (RIF14–0081). The funding body did not play any role in the design of the study, or collection, analysis, or interpretation of data, or in writing the manuscript.

References

Bhatnagar, Sahir (2021). manhattanly: Interactive Q-Q and Manhattan Plots Using 'plotly.js'. R package version 0.3.0. https://CRAN.R-project.org/package=manhattanly

Das, S., Forer, L., Schönherr, S. et al. Next-generation genotype imputation service and methods. Nat Genet 48, 1284–1287 (2016). https://doi.org/10.1038/ng.3656

Delaneau, O., Zagury, JF. & Marchini, J. Improved whole-chromosome phasing for disease and population genetic studies. Nat Methods 10, 5–6 (2013). https://doi.org/10.1038/nmeth.2307

Dowle, Matt and Arun Srinivasan (2021). data.table: Extension of `data.frame`. R package version 1.14.2. https://CRAN.R-project.org/package=data.table

Eller, R.J., Janga, S.C. & Walsh, S. Odyssey: a semi-automated pipeline for phasing, imputation, and analysis of genome-wide genetic data. BMC Bioinformatics 20, 364 (2019). https://doi.org/10.1186/s12859-019-2964-5

Howie B, J. Marchini, and M. Stephens (2011) Genotype imputation with thousands of genomes. G3: Genes, Genomics, Genetics 1(6): 457-470

Lemieux Perreault, Louis-Philippe, Marc-André Legault, Géraldine Asselin, Marie-Pierre Dubé, genipe: an automated genome-wide imputation pipeline with automatic reporting and statistical tools, Bioinformatics, Volume 32, Issue 23, 1 December 2016, Pages 3661–3663, https://doi.org/10.1093/bioinformatics/btw487

Loh, P.-R. et al. Reference-based phasing using the Haplotype Reference Consortium panel. Nature Genetics 48, 1443–1448 (2016).

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ & Sham PC (2007) PLINK: a toolset for whole-genome association and population-based linkage analysis. American Journal of Human Genetics, 81. http://pngu.mgh.harvard.edu/purcell/plink/

Singularity Developers (2021) Singularity. 10.5281/zenodo.1310023 https://doi.org/10.5281/zenodo.1310023

Stahl, K., Gola, D., & König, I. R. (2021). Assessment of Imputation Quality: Comparison of Phasing and Imputation Algorithms in Real Data. Frontiers in genetics, 12, 724037. https://doi.org/10.3389/fgene.2021.724037

Turner, Stephen D (2018). qqman: an R package for visualizing GWAS results using Q-Q and manhattan plots. Journal of Open Source Software, 3(25), 731 doi:10.21105/joss.00731

Wickham, Hadley (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr