**Title:** dbGaPCheckup: pre-submission checks of dbGaP-formatted subject phenotype files

**Authors:** Lacey W. Heinsberg, PhD, RN1; Daniel E. Weeks, PhD1,2

1Department of Human Genetics, School of Public Health, University of Pittsburgh, Pittsburgh, Pennsylvania, USA

2Department of Biostatistics, School of Public Health, University of Pittsburgh, Pittsburgh, Pennsylvania, USA

**Authors Email addresses and ORCID iDs**

Lacey W. Heinsberg: law145@pitt.edu, 0000-0002-7690-5485

Daniel E. Weeks: weeks@pitt.edu, 0000-0001-9410-7228

**Corresponding authors**: Lacey W. Heinsberg, PhD, RN

Department of Human Genetics

School of Public Health

University of Pittsburgh

Public Health 3102A

130 De Soto Street

Pittsburgh, PA 15261

E-mail: law145@pitt.edu

Daniel E. Weeks, PhD

Department of Human Genetics

School of Public Health

University of Pittsburgh

Public Health 3119

130 De Soto Street

Pittsburgh, PA 15261

E-mail: weeks@pitt.edu

**Word count (including abstract, headings, and declarations):** 981

**NOTE:** This is the submitted version of this manuscript (i.e., pre-peer review version) self-archived on L. Heinsberg’s personal website/repository to support accessibility until our paper has been published. A link to the peer reviewed and published version will be added upon publication.

**ABSTRACT**

**Summary:** We developed dbGaPCheckup, an R package which implements a series of check, awareness, reporting, and utility functions to help ensure your subject phenotype data set and data dictionary meet formatting requirements for submission to the National Library of Medicine’s database of Genotypes and Phenotypes (dbGaP).

**Availability:** <https://CRAN.R-project.org/package=dbGaPCheckup>; <https://github.com/lwheinsberg/dbGaPCheckup>

**Contact:** law145@pitt.edu or weeks@pitt.edu.

**Keywords:** Database architecture; data quality control; data sharing; data distribution; repository; database of genotypes and phenotypes

**1. INTRODUCTION**

Making research data publicly available is important to both promote scientific rigor/reproducibility and preserve data beyond the life of the study through which it was originally generated. In recent years, many major funding agencies, including the National Institutes of Health (NIH), have moved toward requiring data sharing across scientific disciplines. A variety of publicly available data repositories exist to facilitate this, including the Database of Genotypes and Phenotypes (dbGaP).1,2 To support curation of thousands of complex data sets, dbGaP has very detailed submission instructions.3 To help researchers meet these multifaceted formatting requirements, and to further support data integrity, we developed dbGaPCheckup, an R package which implements a series of check, reporting, awareness, and utility functions. The package is publicly available at <https://CRAN.R-project.org/package=dbGaPCheckup> and <https://github.com/lwheinsberg/dbGaPCheckup>.

**2. FEATURES**

This package focuses on two required dbGaP subject phenotype files, the data set and the corresponding data dictionary; the functions in our package cover a range of data integrity and formatting requirement checks, many of which are illustrated in Figure 1. For example, our checks ensure that the dbGaP- and package-required data dictionary fields are present; the number and names of variables match between the data set and data dictionary; observed data values are not more extreme than the minimum and maximum values stated in the data dictionary; and more. To execute the package, users simply call the check\_report() function which implements a panel of dbGaPCheckup checks. A complete description of the functions is available in the package documentation at <https://lwheinsberg.github.io/dbGaPCheckup/index.html>.



**Figure 1.** Exemplar dbGaPCheckup pre-submission checks.

Beyond the formatting and data checks, dbGaPCheckup includes several functions that implement minor/scalable fixes. For example, if full application of all the available checks is limited because the input data dictionary is missing the dbGaP-optional but package-required data dictionary fields of TYPE, MIN, and MAX, they can easily be added using the add\_missing\_fields() function. Similarly, the reorder\_dictionary()function reorders the variables in the data dictionary to match the order listed in the data set. Most problems the package identifies, however, will need to be manually corrected by the user as issues are most often study-specific (e.g., coding mismatches; typographical errors; etc.).

Our package also includes reporting functions (i.e., create\_report() and create\_awareness\_report()) which generate graphical and textual descriptives of the data to support more detailed and subjective interrogation. The creation of these report functions was motivated by challenges of (1) checking for consistent use of user-defined missing value codes (e.g., -4444, -9999) and (2) attempting to check for missing value=meaning map elements in dbGaP’s unique format (e.g., -9999=missing value; 0=no).

Finally, we have created the label\_data() function which can be utilized to add information from the data dictionary as attributes to the data set stored as a unified R data frame for ease of future use.4,5 This function even enables SPSS-style encoding/handling of user-defined missing value codes.4,5 Once data are labelled, variable names and codes are automatically displayed within certain views of the data, leading to more human-interpretable and less error-prone workflows and analyses.

**3. DISCUSSION**

Our R package, dbGaPCheckup, is an innovative assistive and timesaving tool that fills an important gap for NIH researchers as it will make dbGaP submission of large and complex data sets less error prone. In fact, through both the objective checks, as well as the more subjective awareness reports, we have caught pre-submission errors in our own data sets that we likely would not have caught otherwise. Not only is our package simple to use but it also eases the burden of complying with the many dbGaP formatting requirements. Further, as many of us have experienced in our prior workflows and collaborations, having the data set and data dictionary as separate files adds a modicum of difficulty in looking up information in the data dictionary. Unfortunately, even small hurdles such as this often deters analysts from looking up required information. The label\_data() function, in particular, brings huge advantages in addressing this issue by merging the data dictionary and data set so that the data dictionary information moves with the data and is readily and easily available.

Finally, beyond its original intended purpose, dbGaPCheckup provides important extrinsic merit. Given the latest data sharing policies from the NIH, which will require all researchers (and not just those generating large scale genomic data) to make their data publicly available as of January 20236, we believe universal adoption of dbGaP formatting and database management styles should be considered by all academic institutions and investigators for database harmonization. A standardized database format of this nature would not only make submissions to data repositories simpler and faster, but would also make internal data management, curation, merging, and sharing across research groups easier. If such an approach is taken, dbGaPCheckup will have important utility across the database architecture phases preceding data submission, further supporting NIH researchers during this important transition into an era of more rigorous and reproducible science.

**DECLARATIONS:**

**Acknowledgements:** The ‘eval\_function’ and ‘dat\_function’ functions that form the backbone of the awareness reports were inspired by an elegant 2016 homework answer submitted by Tanbin Rahman in our HUGEN 2070 course ‘Bioinformatics for Human Genetics’. We would also like to thank Nick Moshgat for testing and providing feedback on our package during development.

**Funding:** This work was partially supported by the National Institutes of Health [grant numbers R01HL133040, R01HL093093, and K99HD107030].

**Conflict of Interest:** The funders had no role in the design of the study or development of the package. As such, the authors declare no conflict of interest.

**REFERENCES**

1. Tryka KA, Hao L, Sturcke A, et al. NCBI’s Database of Genotypes and Phenotypes: dbGaP. *Nucleic Acids Res*. 2014;42(Database issue):D975-979. doi:10.1093/nar/gkt1211

2. Mailman MD, Feolo M, Jin Y, et al. The NCBI dbGaP database of genotypes and phenotypes. *Nat Genet*. 2007;39(10):1181-1186. doi:10.1038/ng1007-1181

3. dbGaP Study Submission Guide. https://www.ncbi.nlm.nih.gov/gap/docs/submissionguide/

4. Wickham H, Miller E, Smith D. haven: Import and Export “SPSS”, “Stata” and “SAS” Files. Published online 2022. https://CRAN.R-project.org/package=haven

5. Larmarange J. labelled: Manipulating Labelled Data. Published online 2022. https://CRAN.R-project.org/package=labelled

6. *Final NIH Policy for Data Management and Sharing*. https://grants.nih.gov/grants/guide/notice-files/NOT-OD-21-013.html