allofus: An R package to facilitate use of the *All of Us* Researcher Workbench

Louisa H. Smith, PhD[1,2], Robert Cavanaugh, PhD[1,2]

[1]Roux Institute, Northeastern University, Portland, ME, USA

[2]Department of Public Health and Health Sciences, Bouvé College of Health Sciences, Northeastern University, Boston, MA, USA

Corresponding Author:

Louisa H. Smith
100 Fore St, Portland, ME 04101
l.smith@northeastern.edu
(617) 373-7008

Key Words: *All of Us*, R package, Common Data Model, OMOP CDM, OHDSI

Word Count: 4115

Conflict of interest: Robert Cavanaugh received a mentor stipend from the *All of Us* Research Scholar Program.

Author contributions:

Louisa Smith: Conceptualization, Methodology, Software, Visualization, Writing

Robert Cavanaugh: Conceptualization, Methodology, Software, Writing

Data availability: All source code and data for the package is available on GitHub (https://github.com/roux-ohdsi/allofus/). *All of Us* Research Program data is available to registered researchers at https://workbench.researchallofus.org/.

Funding statement: This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

ABSTRACT

Objectives: Despite easy-to-use tools like the Cohort Builder, using *All of Us* Research Program data for complex research questions requires a relatively high level of technical expertise. We aimed to increase research and training capacity and reduce barriers to entry for the *All of Us* community through an R package, `allofus`. In this article, we describe functions that address common challenges we encountered while working with *All of Us* Research Program data, and we demonstrate this functionality with an example of creating a cohort of *All of Us* participants by synthesizing electronic health record and survey data with time dependencies.

Target audience: *All of Us* Research Program data are widely available to health researchers. The `allofus` R package is aimed at a wide range of researchers who wish to conduct complex analyses using best practices for reproducibility and transparency, and who have a range of experience using R. Because the *All of Us* data are transformed into the Observational Medical Outcomes Partnership Common Data Model (OMOP CDM), researchers familiar with existing OMOP CDM tools or who wish to conduct network studies in conjunction with other OMOP CDM data will also find value in the package.

Scope: We developed an initial set of functions that solve problems we experienced across survey and electronic health record data in our own research and in mentoring student projects. The package will continue to grow and develop with the *All of Us* Research Program. The `allofus` R package can help build community research capacity by increasing access to the *All of Us* Research Program data, the efficiency of its use, and the rigor and reproducibility of the resulting research.

BACKGROUND AND SIGNIFICANCE

The *All of Us* Research Program is a groundbreaking initiative by the National Institutes of Health (NIH) to advance precision medicine using data from participants who reflect the diversity of the United States.[1] The program collects a wide variety of data, including health and lifestyle surveys, electronic health records (EHR), genetics, wearable, and physical measurement data. These cross-cutting data sources allow researchers to explore biological, clinical, social, and environmental determinants of health and advance medical research on a personalized basis.

The data are linked and transformed into the Observational Medical Outcomes Partnership Common Data Model (OMOP CDM), an open community data standard developed by the Observational Health Data Sciences and Informatics (OHDSI) community.[2] The OMOP CDM is designed to harmonize observational health data for efficient analyses and reliable real-world evidence.[3] The OMOP CDM allows for the transformation of disparate databases (e.g., data from many different hospital systems) and data types into a standard format, facilitating synthesis across data sources and creating the potential for network studies across many global OMOP CDM databases. Critically, the *All of Us* program data are widely available, making the *All of Us* research program data one of the most important research assets today.

After undergoing privacy protections, the transformed database, called the Curated Data Repository (CDR), is made available to researchers on the *All of Us* Researcher Workbench. Here, researchers can create collaborative workspaces, which hold data and analytical scripts and provide access to compute environments. Data are stored in a

Google BigQuery relational database hosted on the Workbench, accessible using SQL (structured query language), either with the statistical programming language R[4] or the general-purpose language Python[5] using Jupyter Notebooks [6] and RStudio,[7] or with the statistical software SAS Studio.[8] A graphical user interface tool called the *Cohort Builder* allows researchers to identify cohorts of interest and create datasets by extracting data from the CDR using automatically generated SQL code to be run in their choice of R, Python, or SAS.

While this setup has made the *All of Us* research program data widely available to researchers, a high technical burden for successful research remains. Studies beyond basic characterization often require intermediate to advanced programming skills in R, Python, SAS, and/or SQL and an understanding of the OMOP CDM, ultimately limiting the accessibility of the *All of Us* program. In our own experience using and mentoring students using the Researcher Workbench, this technical burden often slows research productivity and results in frustration. It can result in questionable research practices, and errors are often difficult to identify or fix. Similar experiences likely harm the research goals of the *All of Us* community and increase disparities in the successful use of the data between researchers with different training opportunities and backgrounds.

OBJECTIVES

In response, we have developed the `allofus` R package to facilitate successful, reproducible use of the *All of Us* research program data. By increasing program accessibility and researcher competency for a diverse community of researchers, the `allofus` R package aims to build research capacity, ultimately providing value to the wider

All of Us community. Our goals were to create a tool that would 1) make connecting to the database and managing files simple, 2) facilitate the use of the popular "tidyverse" ecosystem of R packages[9] on *All of Us* data, 3) help researchers more efficiently and accurately extract and synthesize survey data and EHR data, 4) increase the interoperability between fully-featured tools created by the OHDSI community and the Researcher Workbench. In addition, we have created a small but growing set of tutorials for researchers of all skills and backgrounds that demonstrate how to use the package.

We designed the `allofus` R package to be a complementary tool to be used in conjunction with the Researcher Workbench according to the needs and skills of the researcher. Our target audience includes students and researchers with some familiarity with R and the tidyverse data science ecosystem who are designing and implementing studies on Jupyter Notebooks or in RStudio, including those new to relational databases or the OMOP CDM. Also included in the target audience are researchers who are already familiar with OHDSI tools such as ATLAS or researchers with advanced R skills that might benefit from minimizing repetitive code needed for many common tasks. In this article, we review the functionality of the `allofus` R package and demonstrate how it builds on existing Researcher Workbench resources.

PACKAGE DESCRIPTION AND FUNCTIONALITY

Package design

The `allofus` R package provides functions that address common challenges we encountered when working with *All of Us* Research Program Data. Table 1 describes these functions with respect to how they facilitate the use of *All of Us* data. The package is specifically designed to leverage the tidyverse ecosystem of R packages,[9] including the dplyr package,[10] to interact with the *All of Us* database. The general workflow for using the package functions is depicted in Figure 1. In this section, we briefly describe some of the key technical barriers to the successful use of the Researcher Workbench and *All of Us* Program database that are addressed by the `allofus` R package.

Table 1: List of current functions and data available within the `allofus` R package

Function	**Description**	**Challenge addressed**
Connection helpers		
`aou_connect()`	Connects to the *All of Us* database and returns a BigQueryConnection object, which can be referenced to query the database using R and or SQL code	Avoids long code snippet with every connection
`aou_session_info()`	Returns a table of information that is necessary to fully reproduce an analysis, including the R version, the packages loaded and their versions, and the *All of Us* CDR release	Improves reproducibility by making documentation simple
`aou_sql()`	Executes a SQL query on the *All of Us* database and returns the result as a data frame	Allows for running SQL queries without bringing the data into local memory immediately
Analysis helpers		
`aou_atlas_cohort()`	Retrieves a cohort definition from ATLAS and generates the cohort in *All of Us*	Increased interoperability with OHDSI toolstack allows for easier collaboration and better reproducibility
`aou_compute()`	Computes a temporary table from a dplyr chain that returns an SQL query (e.g., `tbl(con, "person") %>% select(person_id)`)	Overly long SQL queries take a long time to run repeatedly when iteratively working on code, or can even cause errors
`aou_concept_set()`	Retrieves occurrences of a concept set from the *All of Us* database for a given cohort	Makes extracting and summarizing EHR data across multiple OMOP CDM tables straightforward
`aou_create_temp_table()`	Generates a temporary table from a local data frame	Allows for merging remote tables with user-provided data, e.g., to subset before bringing into local memory
`aou_join()`	Joins two tables in the *All of Us* database	Provides safeguards compared to `dplyr::*_join()`, e.g., when

		joining two tables with the same column names
`aou_observation_period()`	Generates a temporary observation period table based the first and last event in the electronic medical record data	Allows for cohort creation based on duration or timing of EHR data
`aou_survey()`	Extracts survey responses in a tidy format that also includes 'skip' responses and collapses across all versions of the person health / personal medical history surveys	Multiple versions of the health history survey make it difficult to combine data, and the nested nature of the family health questions can make it challenging to identify skips and "no" responses
Workspace helpers		
`aou_bucket_to_workspace()`	Retrieves a file from the workspace bucket and moves it into the current workspace where it can be read into R, e.g., using a function like `read.csv()`	Clarify and facilitate movement between the two storage locations on the Researcher Workbench
`aou_ls_bucket()`	Lists all files in the bucket or files matching a certain pattern	
`aou_ls_workspace()`	Lists all data files in the workspace or files matching a certain pattern	
`aou_workspace_to_bucket()`	Moves a file saved in on the persistent disk to the workspace bucket	
Data dictionaries		
`aou_table_info`[a]	A data frame containing tables and columns in the *All of Us* Controlled Tier Dataset v7 CDR Data Dictionary	Provide curated information about the data and serve as the backend for interactive data tables on the package website (https://github.com/roux-ohdsi/allofus/)
`aou_codebook`	A data frame with rows from the *All of Us* Survey Codebook mapped to the *All of Us* PPI Vocabulary available on Athena	
`aou_health_history`	A data frame with rows of the *All of Us* codebook pertaining to the health history questions	

Connecting to the database

Because *All of Us* data are stored centrally in a BigQuery cloud database,[11] any analysis must connect to the database in some way. The *All of Us* Cohort Builder generates a SQL query, wrapped in R, Python, or SAS code (depending on the selected environment), which, when executed, executes the query and downloads the resulting table to the "workspace bucket" (a remote shared storage space for a collaborative workspace). Future analyses must move this file from the bucket to the “persistent disk” (a more temporary user-specific storage space) and load the data into local memory for analysis. Another option for interacting with BigQuery cloud database in R is via the `dbplyr`[12] package, which allows researchers to use `dplyr`[10] and other R functions directly on the database by translating them to SQL code. The `allofus` package takes advantage of this functionality with the `aou_connect()` function, which creates a BigQuery database connection using the `DBI` package[13] that stays open as long as a notebook is in use or RStudio session is running. Many of the package functions work on the database via this connection but researchers can also directly use the database querying tools provided by `dbplyr` and `DBI`.

Writing and executing queries

A typical `dbplyr` query consists of a reference to one or more remote tables and a series of `dplyr` functions to manipulate data from the tables (e.g., `mutate()`, `filter()`, and `select()`) connected using a pipe operator. The SQL query it represents is only executed when the R object is printed or when the `collect()` function is used to bring the data into local memory. The `allofus` package provides several functions to facilitate this process within the constraints of the BigQuery database. When queries get too long or complex, they can result in SQL errors. The `aou_compute()` function forces early execution of the query, storing the results in a temporary table for additional manipulation. The `aou_collect()` function, which executes a query and downloads its result, is a wrapper for `dplyr::collect()`that deals explicitly with integer-valued columns that may be too large to be represented with base R’s integer class.

Along with the option to write code using dplyr syntax, users can also use the aou_sql() function to easily send a SQL query to the database. References to database tables in the form of “{CDR}.table” will be automatically completed, and other objects or R code can be included within curly braces, allowing for parametrized queries. As with most of the package functions, the collect argument allows the user to choose to leave the query unexecuted on the database or download it into local memory.

Some queries rely on user-provided data. For example, filtering data for a large number of codes can be made more efficient by joining the data with a table of those codes. However, due to bigrquery limitations on temporary tables and lack of write access on the CDR, joining remote tables with local data is not straightforward. To get around these limitations, we wrote aou_create_temporary_table(), which creates a new temporary table from a local dataframe.

EHR and survey data tools

Survey and EHR data are standardized according to the OMOP CDM. This data model has been described in detail elsewhere;[3] briefly, codes (e.g., ICD-10 codes) are mapped to standard “concepts”. Electronic health record research using OMOP often involves creating a list of the standardized codes (a “concept set”) that define a certain condition. Most concepts can be found by searching the public data browser, though some (e.g., those for devices) cannot. Alternatively, all concepts can be found using publicly available tools such as Athena (https://athena.ohdsi.org/). After a researcher has defined a concept set, the aou_concept_set() searches for instances of the selected codes across OMOP tables. Data can either be returned in full (i.e., all observations of the codes) or in summary (as a count of codes or indicator that at least 1 was present). Arguments to set the start and end date for each individual can restrict the extracted codes to a specific period.

The survey data are currently stored in two places in the database: the custom "ds_survey" table, which supports the Researcher Workbench GUI, and the OMOP "observation" table. Although the "ds_survey" table is more intuitively structured, it does not contain complete skip and non-response information for the health/medical history surveys. Therefore, our survey tools are based on the observation table and are designed to address the complex nested nature of these data and encourage researchers to make explicit choices for dealing with missing data.

The `aou_survey()` function returns a table with a column for the response to each requested survey question and the response date. To find concepts representing questions of interest for the function, researchers can use two codebooks (`aou_codebook`, `aou_health_history`) in the allofus package (also available on the allofus package website) or the aforementioned publicly available tools. The codebooks are based on CDR v7.0 and contain information on the survey questions, the response options, and the codes used to represent those response options in the database.

Cohort creation

ATLAS[14] is an open-source web app created by the OHDSI community that connects to an OMOP database and includes a vast array of tools for working with real-world data. Atlas' concept set tools allow users to search for and get counts for concepts, much like the *All of Us* data browser. It also has functionality for creating complex cohort definitions (e.g., based on time-varying criteria), which are often shared among researchers to conduct network studies and for research consistency and reproducibility. Although ATLAS can't be directly linked to the *All of Us* database, it can be used to define cohorts to be executed on *All of Us* data. Researchers without access to an ATLAS instance can use the free demo version (https://atlas-demo.ohdsi.org/) to define a cohort to be executed using the `aou_atlas_cohort()` function on the Researcher Workbench.

Generating ATLAS cohorts requires an "observation period" table, which contains the dates under which individuals were "under observation", i.e., when their health data would have been likely to be captured. The *All of Us* observation period table is not set up for this purpose, so we designed the `aou_observation_period()` function to create a table with the earliest and latest date each participant's EHR data appear in the database. The function can also be used on its own to facilitate manual cohort construction. Because some EHR sites have contributed data from several decades ago, researchers might want to consider further constraining this table to reasonable date ranges of interest.

File management

The `allofus` package also includes helper functions to simplify file management. On the Researcher Workbench, files are stored in a Google Cloud bucket and must be moved via command line prompts. We designed a set of functions to list the files in the bucket or workspace associated with the current project and to copy files between the two locations, simplifying the Workbench code snippets. A comparison between the provided code snippets and the file management functions in the `allofus` R package is shown in Table 2.

Table 2: Comparison between provided command line code snippets and allofus R package functions for file management

Task	Provided code snippet	allofus function

List files in the bucket	`# Get the bucket name`
	`my_bucket <- Sys.getenv('WORKSPACE_BUCKET')`
	`# List objects in the bucket`
	`system(paste0("gsutil ls -r ", my_bucket), intern=T)`
		`aou_ls_bucket()`

Move a file from the bucket to workspace disk	`# replace 'test.csv' with the name of the file in your google bucket (don't delete the quotation marks)`
	`name_of_file_in_bucket <- 'test.csv'`
	`# Get the bucket name`
	`my_bucket <- Sys.getenv('WORKSPACE_BUCKET')`
	`# Copy the file from current workspace to the bucket`
	`system(paste0("gsutil cp ", my_bucket, "/data/", name_of_file_in_bucket, " ."), intern=T)`
	`# Load the file into a dataframe`
	`my_dataframe  <- read_csv(name_of_file_in_bucket)`
		`aou_bucket_to_workspace("test.csv")`

Write a file to disk and move it to the bucket	`# Replace df with THE NAME OF YOUR DATAFRAME`
	`my_dataframe <- df`
	`# Replace 'test.csv' with THE NAME of the file you're going to store in the bucket (don't delete the quotation marks)`
	`destination_filename <- 'test.csv'`
	`# store the dataframe in current workspace`
	`write_excel_csv(my_dataframe, destination_filename)`
	`# Get the bucket name`
	`my_bucket <- Sys.getenv('WORKSPACE_BUCKET')`
	`# Copy the file from current workspace to the bucket`
	`system(paste0("gsutil cp ./", destination_filename, " ", my_bucket, "/data/"), intern=T)`
	`# Check if file is in the bucket`
	`system(paste0("gsutil ls ", my_bucket, "/data/*.csv"), intern=T)`
		`write.csv(df, "test.csv")`
		`aou_workspace_to_bucket(df, "test.csv")`

Package implementation and availability

We wrote the allofus R package for use on the Researcher Workbench (current R version 4.3.1) The primary dependencies are to the bigrquery R package,[15] which facilitates connection to the Google BigQuery database, and to `dbplyr`, `dplyr` and other tidyverse packages. The package is designed specifically for use on the Researcher Workbench, includes unit tests for key functions, and is licensed under the MIT License.[16] It is available for download on the R package repository CRAN (https://cran.r-project.org/web/packages/allofus/index.html), and a website with documentation, searchable reference tables, and tutorials is available at https://roux-ohdsi.github.io/allofus/. All source code and data for the package are available on GitHub (https://github.com/roux-ohdsi/allofus/), and *All of Us* Research Program data are available to registered researchers at https://workbench.researchallofus.org/.

EXAMPLE

To illustrate the basic functions of the `allofus` R package, the following will describe the preparation of a simplified cohort of *All of Us* participants who do not have Type 2 Diabetes (T2DM), e.g., to look at risk factors for new diagnoses. We will demonstrate performing the data manipulation directly in the database rather than downloading and wrangling local data in the workspace. A Jupyter Notebook with the code is provided as Supplementary Material.

To define this cohort, we decide that we can be reasonably confident that participants have not been previously diagnosed with T2DM if they have at least 1 year of EHR data prior to joining *All of Us* without any codes related to T2DM. To identify those eligible, we identify each participant's earliest survey date, subtract 1 year, and join the data to the table created by `aou_observation_period().`

```r
library(allofus)
library(dplyr)

# connect to the database
con <- aou_connect()

# create an observation period table for all participants
observation_period_tbl <- aou_observation_period()

# find the first survey date and the date exactly one year before
survey_date_tbl <- tbl(con, "ds_survey") %>%
  summarize(first_survey = as.Date(min(survey_datetime)),
            .by = "person_id") %>%
  mutate(year_before_survey = DATE_ADD(first_survey,
                                       sql(paste("INTERVAL", -1, "year"))))

# use join_by(person_id, within(…)) to keep only participants for whom the
# full year before their survey falls within the start and end of their
# observation period.
eligible <- survey_date_tbl %>%
  aou_join(observation_period_tbl, type = "inner",
           by = join_by(person_id,
                        within(year_before_survey, first_survey,
                               observation_period_start_date,
                               observation_period_end_date)))
```

Next, we will consider someone to have a diagnosis of T2DM if they have at least 2 of: a diagnosis code for Type 2 diabetes, a prescription for metformin, a lab test for hemoglobin A1c measuring above 6.5%; or if they have a device for home blood glucose monitoring. In this toy example, we will use a non-exhaustive list of related concepts in Table 3, acknowledging that a real analysis should be more thorough and thoughtful about these choices.

Table 3: Example concept set with diabetes-related concepts from various domains, used to create a simplified cohort definition of All of Us participants with Type 2 Diabetes.

Concept ID	**Concept Name**	**Domain**	**Rule**
201826	Type 2 diabetes mellitus	Condition	At least 2
4193704	Type 2 diabetes mellitus without complication	Condition	
40164929	metformin hydrochloride 500 MG Oral Tablet	Drug	
40164897	metformin hydrochloride 1000 MG Oral Tablet	Drug	
3004410	Hemoglobin A1c/Hemoglobin.total in Blood	Measurement	At least 2 > 6.5
3005673	Hemoglobin A1c/Hemoglobin.total in Blood by HPLC	Measurement	
2616672	Home blood glucose monitor	Device	At least 1

3034639	Glucose monitoring system	Device

We can extract the data using `aou_concept_set()`, combine it into a single table, and let it temporarily remain in the database so that we can perform more SQL manipulations. By default, `aou_concept_set()` will pull all matching codes from all OMOP tables across the entire timespan of data. However, to look for people with pre-existing T2DM, we need to specify `start_date` and `end_date` arguments to refer to the year before the initial *All of Us* survey. To identify people with sufficient evidence of T2DM, we use the function to request an indicator variable for the presence of at least 1 device code during that time period, a variable indicating whether a person has at least two of the condition or drug codes, and all of the related measurement data so that we can identify values above the minimum. Then we will combine these datasets into a single table containing participants with a previous T2Dm diagnosis (i.e., ineligible) and then remove them from our initial cohort.

```
codes_condition_drug <- c(201826, 4193704, 40164929, 40164897)
codes_lab_test <- c(3004410, 3005673)
codes_device <- c(2616672, 3034639)

# request variable called device with 1 if any concepts, 0 otherwise
device <- aou_concept_set(eligible, codes_device,
                          start_date = "year_before_survey",
                          end_date = "first_survey",
                          output = "indicator",
                          concept_set_name = "device",
                          domains = "device") %>%
  filter(device == 1) %>%
  distinct(person_id)

# request variable called condition_drug with 1 if >= 2 concepts, 0 otherwise
condition_drug <- aou_concept_set(eligible, codes_condition_drug,
                                  start_date = "year_before_survey",
                                  end_date = "first_survey",

                                  output = "indicator",
                                  min_n = 2,
                                  concept_set_name = "condition_drug",
                                  domains = c("condition", "drug")) %>%
  filter(condition_drug == 1) %>%
  distinct(person_id)

# request all data on these concepts for filtering above 6.5, then to people
# with at least 2
lab_test <- aou_concept_set(eligible, codes_lab_test,
                            start_date = "year_before_survey",
                            end_date = "first_survey",
                            output = "all",
                            concept_set_name = "lab_test",
                            domains = "measurement") %>%
  filter(value_as_number > 6.5) %>%
  summarise(n = n(), .by = "person_id") %>%
  filter(n >= 2) %>%
  distinct(person_id)

# combine tables and create intermediate temporary table (the queries
# resulting in eligible and ineligible are otherwise too complex to merge)
ineligible <- list(condition_drug, lab_test, device) %>%
  purrr::reduce(union_all) %>%
  aou_compute()

# remove from eligible cohort
eligible <- eligible %>%
  anti_join(ineligible, by = "person_id")
```

To illustrate how the `allofus` R package mitigates challenges related to the *All of Us* health-related surveys, we will also confirm participants reported no prior T2DM diagnosis in the *All of Us* Personal and Family Health History survey (currently administered) or in the Personal Medical History (administered until 2021).[17] Changes in the survey design make it difficult to correctly classify participants over time; however, the `allofus` R package facilitates the creation of straightforward “Yes”/”No” /”Skip”/NA variables for health history questions.

For example, concept ID 43529932 stores the self-reported T2DM response (“Including yourself, who in your family has had type 2 diabetes? – Self”). In the initial Personal Medical History survey, the question that would elicit a respondent’s personal history of T2DM was “Has a doctor or health care provider ever told you that you have___?” with a list of endocrine conditions. The currently administered Personal and Family Health History survey also asks about family medical history: “Have you or anyone in your family ever been diagnosed with the following hormone and endocrine conditions?” When a respondent selects a condition such as T2DM, they are then asked, “Including yourself, who in your family has had Type 2 diabetes?” with options including “Self” and other family members. This multistage question makes it difficult to correctly combine data across both surveys and identify missing data and “No” responses. For example, the lack of a “Self” response to the more recent question format could mean that the respondent didn’t answer any questions about endocrine conditions, that someone in their family does have T2DM but that individual does not, or that no one has any endocrine conditions in their family.

Along with “Yes” (i.e., “Self”) responses identified by the concept ID, the `aou_survey()` function handles the ambiguity of other responses by classifying participants as “No” (assumed when someone answers that, for example, no one in their family has an endocrine condition, or they report that someone else but not themselves has type 2 diabetes or another endocrine condition), “Don’t know”/“Skip”/“Prefer not to answer” (to either of the parent questions), or NA, which indicates that a person never completed a survey with that question. Other survey questions, e.g., about education and income (both from the Basics survey), are more straightforward.

To identify participants who self-reported T2DM (and extract information about education and income), we provide `aou_survey()`with our initial cohort of participants without an EHR indication of T2DM, concept IDs for self-report of T2DM, education, and income, and the desired names of the columns for each of these concept IDs. Joining back to our eligible table produces a final cohort of people who had one year of EHR data prior to the first survey, who had no indication of Type 2 diabetes in that period, and who took one of the two versions of the personal health history survey and reported that they did not have Type 2 diabetes.

```
survey_data <- aou_survey(eligible,
                          questions = c(43529932, 1585940, 1585375),
                          question_output = c("t2dm", "edu", "income"))

eligible <- survey_data %>%
  filter(t2dm == "No") %>%
  aou_join(eligible, type = "inner", by = "person_id")
```

All data manipulation has taken place directly on the remote database, and this final table remains there. This minimizes the need to extract information on participants who have already been found ineligible in earlier steps. It also allows us to efficiently revise code if we make different decisions about defining the cohort. To save the data, make a table describing the cohort, or fit a regression model, we can bring it into memory using `aou_collect()`, which will convert integer-valued data to avoid errors in certain settings. This object is now a data frame that could be saved, e.g., as a .csv file. After saving, we can move it to the bucket for permanent storage and to share with collaborators with access to the project with `aou_workspace_to_bucket()`. Because the entire script was written in R with tidyverse-style code, rather than in SQL code generated by the Cohort Builder, sharing the script can allow researchers across levels of coding skill to easily examine our analytic choices.

DISCUSSION

In creating the `allofus` R package, we have attempted to reduce the technical burden and statistical programming knowledge required to conduct rigorous research on the *All of Us* research program database. The package stemmed from our experiences using the Researcher Workbench and mentoring students whose research projects use *All of Us* data. The functions in the package range from abstracting away bash commands and complex SQL code to extracting complex survey data, increasing compatibility with a key OHDSI tool, and allowing for easy linking of external data with the database. These tools enable R users to work more seamlessly with the *All of Us* program data directly and improve accessibility for beginner R users without threatening the rigor and validity of research. "Tidyverse-style" is often the preferred syntax for teaching new R users as it comprises a fully-featured data science ecosystem intentionally designed to ease the learning process with improved readability.[19]

By creating methods to make this ecosystem more compatible with the *All of Us* Researcher Workbench, data science tasks, such as creating a cohort and extracting data from various database tables, become more accessible to a wider range of researchers, who benefit from a consistent toolkit.[20] We have published with the package a series of tutorials with commented code and explanations for extracting and wrangling the data in different ways. We plan to continue building these tutorials and invite other researchers to contribute their own via the package GitHub repository. Although we have a number of unit tests designed to detect bugs and have extensively tested package functions in our own research, bugs are inevitable, and we also invite users to report these via GitHub as they arise.

Future directions

We have identified several key future directions for the package. First, we will continue to work on increasing interoperability with OHDSI tools such as ATLAS and Phenotype Library to allow researchers to leverage these tools in *All of Us* data. Additionally, while the current R package has focused primarily on the survey and EHR data, we aim to expand our attention to the other data sources: physical measurement, Fitbit, and genomics data.

Secondly, we aim to add methods that assess and mitigate biases resulting from missing data, selection bias, and lack of representativeness in *All of Us*. Although the *All of Us* mission is to ensure representation from groups historically underrepresented in research, the study has no sampling frame. Participation bias is thus difficult to assess but can threaten the validity of *All of Us* findings. In addition, there are varying degrees of missing data throughout the *All of Us* surveys (e.g., “skip” responses) that require thoughtful methods to address. Future package functionality will support addressing these biases.

CONCLUSION

The `allofus` R package aims to return value to communities by contributing to the training and building of community research capacity and by supporting rigorous, reproducible research. We hope that it will increase the accessibility of the *All of Us* research program for a diverse community of researchers with a range of existing skills, in concert with existing tools. Other researchers are encouraged to participate in this community project by suggesting and contributing extensions to the package.

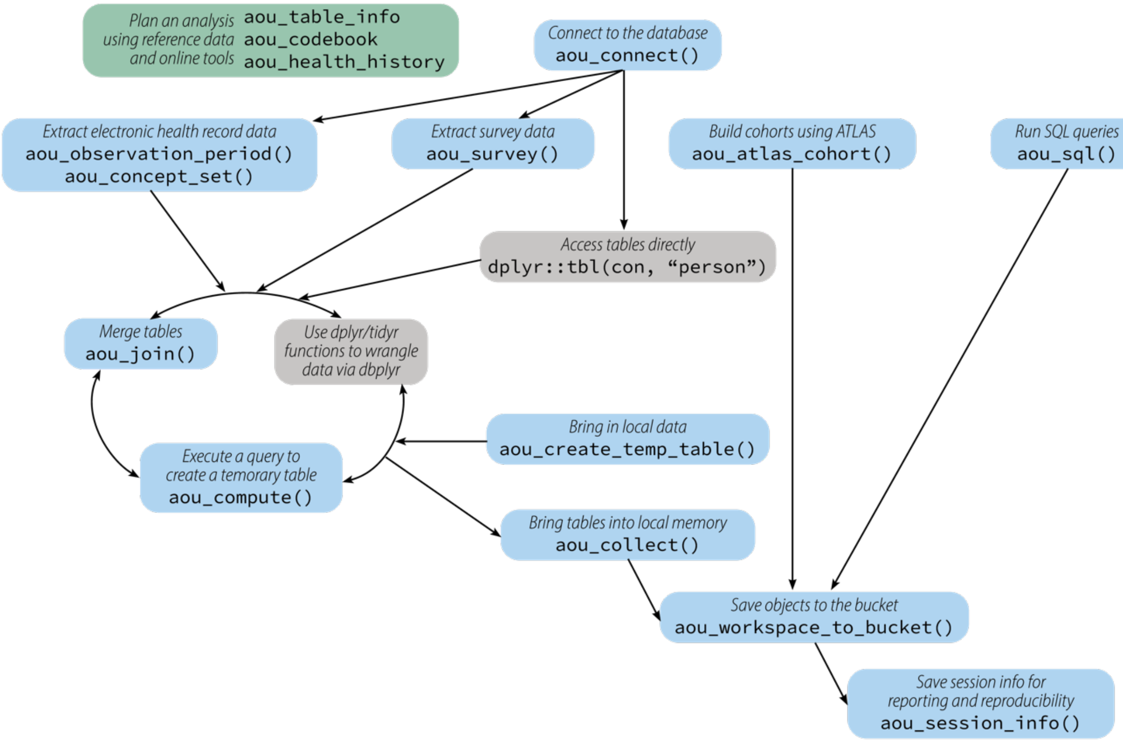
Acknowledgments

We gratefully acknowledge *All of Us* participants for their contributions, without whom this research would not have been possible, as well as the National Institutes of Health's *All of Us* Research Program for making available the data and research platform on which this work was conducted. The All of Us Research Program is supported by the National Institutes of Health, Office of the Director: Regional Medical Centers: 1 OT2 OD026549; 1 OT2 OD026554; 1 OT2 OD026557; 1 OT2 OD026556; 1 OT2 OD026550; 1 OT2 OD 026552; 1 OT2 OD026553; 1 OT2 OD026548; 1 OT2 OD026551; 1 OT2 OD026555; IAA #: AOD 16037; Federally Qualified Health Centers: HHSN 263201600085U; Data and Research Center: 5 U2C OD023196; Biobank: 1 U24 OD023121; The Participant Center: U24 OD023176; Participant Technology Systems Center: 1 U24 OD023163; Communications and Engagement: 3 OT2 OD023205; 3 OT2 OD023206; and Community Partners: 1 OT2 OD025277; 3 OT2 OD025315; 1 OT2 OD025337; 1 OT2 OD025276.

We also wish to thank Chloe Bennett, Narcissa Plummer, and Chelsea Wong for helping test the package's functionality.

Figure captions

Figure 1: A simplified schematic of possible workflows on the *All of Us* Researcher Workbench using the `allofus` R package. Nodes in blue are package functions (a non-exhaustive list). The package is designed to integrate seamlessly with dbplyr (gray nodes), enabling dplyr functions on database backends. The package also provides several data tables, which are also available online, to help plan analyses.


Plan an analysis using reference data and online tools
aou_table_info
aou_codebook
aou_health_history
Connect to the database
aou_connect()
Extract electronic health record data
aou_observation_period()
aou_concept_set()
Extract survey data
aou_survey()
Build cohorts using ATLAS
aou_atlas_cohort()
Run SQL queries
aou_sql()
Access tables directly
dplyr::tbl(con, “person”)
Merge tables
aou_join()
Use dplyr/tidyr functions to wrangle data via dbplyr
Bring in local data
aou_create_temp_table()
Execute a query to create a temorary table
aou_compute()
Bring tables into local memory
aou_collect()
Save objects to the bucket
aou_workspace_to_bucket()
Save session info for reporting and reproducibility
aou_session_info()

References

1 The “All of Us” research program. *N Engl J Med*. 2019;381:668–76.

2 Hripcsak G, Duke JD, Shah NH, *et al.* Observational health data sciences and informatics (OHDSI): Opportunities for observational researchers. *Stud Health Technol Inform*. 2015;216:574–8.

3 Reich C, Ostropolets A, Ryan P, *et al.* OHDSI Standardized Vocabularies—a large-scale centralized reference ontology for international data harmonization. *J Am Med Inform Assoc*. 2024;31:583–90.

4 R Core Team. R: A language and environment for statistical computing. 2023. https://www.R-project.org/

5 van Rossum G. Python. CRC Press 1998. https://doi.org/10.1201/9780849331350.ch23

6 Kluyver T, Ragan-Kelley B, Pérez F, *et al.* Jupyter Notebooks: a publishing format for reproducible computational workflows. In: Loizides F, Scmidt B, eds. *Positioning and power in academic publishing: Players, agents and agendas*. IOS Press 2016:87–90. https://eprints.soton.ac.uk/403913/

7 Posit Team. RStudio: Integrated development environment for R. 2024. http://www.posit.co/

8 SAS Institute Inc. SAS302-2014.pdf. https://support.sas.com/en/software/studio-support.html

9 Wickham H, Averick M, Bryan J, *et al.* Welcome to the Tidyverse. *J Open Source Softw*. 2019;4:1686.

10 Wickham H, François R, Henry L, *et al.* dplyr: A grammar of data manipulation. 2023. https://CRAN.R-project.org/package=dplyr

11 BigQuery Enterprise Data Warehouse. Google Cloud. https://cloud.google.com/bigquery

12 Wickham H, Girlich M, Ruiz E. dbplyr: A “dplyr” back end for databases. 2024. https://CRAN.R-project.org/package=dbplyr

13 R Special Interest Group on Databases (R-SIG-DB), Wickham H, Müller K. DBI: R database interface. 2024. https://CRAN.R-project.org/package=DBI

14 Home · OHDSI/Atlas Wiki. https://github.com/OHDSI/Atlas/wiki (accessed 4 April 2024)

15 Wickham H, Bryan J. bigrquery: An interface to google's "BigQuery" "API." 2024. https://CRAN.R-project.org/package=bigrquery

16 The MIT License. Open Source Initiat. 2006. https://opensource.org/license/MIT (accessed 3 April 2024)

17 All of Us Survey Changes Over Time. User Support. 2023. https://support.researchallofus.org/hc/en-us/articles/14935566710804-All-of-Us-Survey-Changes-Over-Time (accessed 5 April 2024)

18 All of Us Public Data Browser. Pers. Fam. Health Hist. https://databrowser.researchallofus.org/survey/personal-and-family-health-history/diabetes (accessed 5 April 2024)

19 Çetinkaya-Rundel M, Hardin J, Baumer BS, *et al.* An educator's perspective of the tidyverse. *Technol Innov Stat Educ*. 2022;14. doi: 10.5070/T514154352

20 Pruim R, Gîrjău M-C, Horton NJ. Fostering Better Coding Practices for Data Scientists. *Harv Data Sci Rev*. 2023;5. doi: 10.1162/99608f92.97c9f60f