The Journal of Open Source Software

REDCapTidieR: Extracting complex REDCap databases into tidy tables

Richard Hanna[1], **Ezra Porter**[1], **Stephany Romero**[1], **Paul Wildenhain**[6], **William Beasley**[7], and **Stephan Kadauke**[2,3,4,5]

1 Division of Oncology, Children's Hospital of Philadelphia, Philadelphia, Pennsylvania **2** Department of Biomedical and Health Informatics, Children's Hospital of Philadelphia, Philadelphia, Pennsylvania **3** Department of Pathology and Laboratory Medicine, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, Pennsylvania **4** Division of Transfusion Medicine, Children's Hospital of Philadelphia, Pennsylvania **5** Division of Pathology Informatics, Children's Hospital of Philadelphia, Pennsylvania **6** Division of Pediatrics, Children's Hospital of Philadelphia, Philadelphia, Pennsylvania **7** Department of Pediatrics, The University of Oklahoma Health Sciences Center, College of Medicine, Oklahoma City, Oklahoma, USA

DOI: 10.21105/joss.06277

Software

- Review
- Repository
- Archive

Editor: Michael Mahoney

Reviewers:

- @RhysPeploe
- @spgarbet

Submitted: 05 December 2023
Published: 14 February 2024

Summary

Capturing and storing electronic data is integral in the research world. REDCap (Harris et al., 2009, 2019) offers a secure web application that lets users build databases and surveys with a robust front-end interface that can support data of any type, including data requiring compliance with standards for protected information.

Many REDCap users use the R programming language (R Core Team, 2020) to extract and analyze their data. The `REDCapR` (Beasley, 2023) and `redcapAPI` (Garbett et al., 2024) packages allow R users to extract data directly into their programming environment. While this works well for simple REDCap databases, it becomes cumbersome for complex databases, because the REDCap API outputs a "block matrix"–a single table with varied granularity levels, which conflicts with the "tidy data" framework (Wickham, 2014) that advocates for standardized data organization.

To address this, we introduce `REDCapTidieR`, an open-source package that streamlines data extraction and restructures it into an intuitive format compatible with the tidy data principles. This facilitates seamless data analysis in R, especially for complex longitudinal studies.

While there are several tools available for REDCap data management, REDCapTidieR introduces a unique solution by transforming the challenging block matrix into a standardized tidy data structure that we term the "supertibble". This approach not only aligns with good data science practice but also caters to databases of any complexity. By providing a suite of utility functions to work with the supertibble, REDCapTidieR provides a complete framework for extracting REDCap data designed with user-friendliness at its core.

Statement of Need

As of 2023, the REDCap Consortium boasts nearly 3 million users across over 150 countries. REDCap databases range from single-instrument projects to complex builds that use both repeating instruments and repeating events. These data structures are needed to capture multiple items related to a specific visit, such as concomitant medications, or events that cannot be planned ahead of time, such as adverse events.

REDCap databases that contain repeating events and instruments require significant manual pre-processing, a major pain point for researchers and analysts. This is because the REDCap API returns a single table (Figure 1) that includes data from instruments that record data at different levels of granularity.

While several existing REDCap packages are available (Table 1), REDCapTidieR distinguishes itself by offering analysts a unique framework that returns a tidy data structure regardless of the size or complexity of the extracted database. Packages such as tidyREDCap (Balise et al., 2023) and REDCapDM (Carmezim et al., 2023) also offer tools for data processing, while redcapAPI gives a wealth of options for data export in addition to features that break apart the block matrix using a base R framework. However, only REDCapTidieR deconstructs the block matrix into easily joinable tidy tables that form their own composite primary keys to preserve the relationships between each other in accordance with their unique granularity.

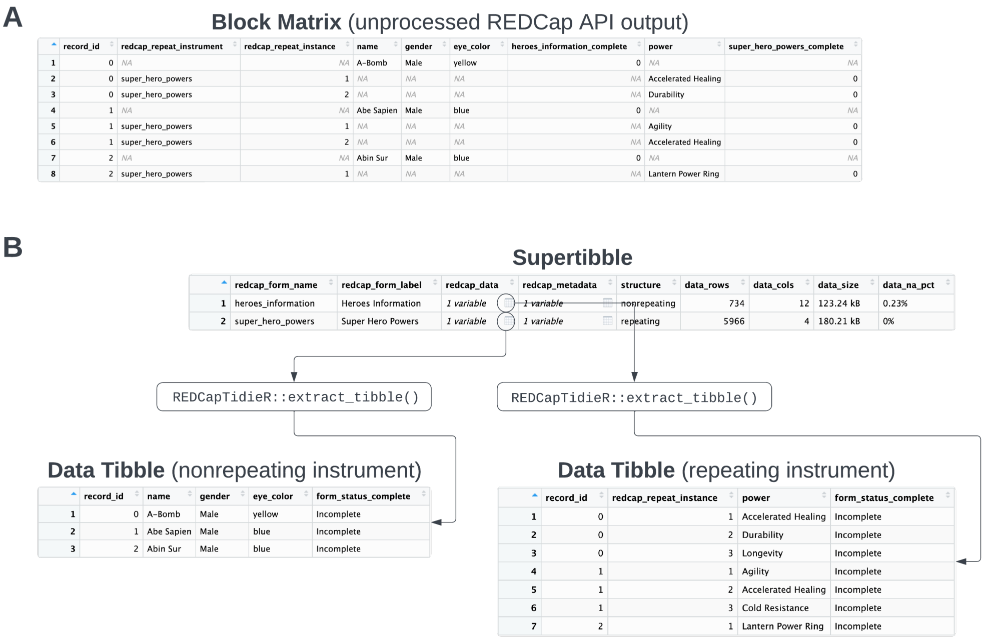
REDCapTidieR is built with production readiness in mind. In addition to an extensive 98% coverage test suite, REDCapTidieR execution is evaluated against 15 test databases that cover many complex configuration scenarios. It also provides ample documentation through a pkgdown site (Hanna et al., 2023). It is also built on top of REDCapR, which contains its own extensive test suite, and evaluated against an additional 26 test databases. REDCapTidieR meets the rigorous requirements of the OpenSSF Best Practices Badge (Open Source Security Foundation, 2023), which certifies open-source projects that adhere to criteria for delivering high-quality, robust, and secure software.

Package	Exports from REDCap	Imports into REDCap	Tidy Reformatting	Extensive Test Suite
redcapAPI	x	x		x
REDCapR	x	x		x
tidyREDCap	x			
REDCapDM	x			
REDCapTidieR	x		x	x

Table 1: Comparative breakdown of the landscape for REDCap tools in R.

Design

The REDCapTidieR::read_redcap() function leverages REDCapR to make API calls to query the data and metadata of a REDCap project and returns the supertibble (Figure 1). The supertibble, named after the tibble package (Müller & Wickham, 2023), is an alternative presentation of the data in which multiple tables are linked together in a single object in a fashion consistent with tidy data principles. Specific data tibbles within the supertibble, representing the data of individual REDCap instruments, can be easily joined using their composite primary keys.


A
B
REDCapTidieR::extract_tibble()
REDCapTidieR::extract_tibble()

Block Matrix (unprocessed REDCap API output)

	record_id	redcap_repeat_instrument	redcap_repeat_instance	name	gender	eye_color	heroes_information_complete	power	super_hero_powers_complete
1	0	NA	NA	A-Bomb	Male	yellow	0	NA	NA
2	0	super_hero_powers	1	NA	NA	NA	NA	Accelerated Healing	0
3	0	super_hero_powers	2	NA	NA	NA	NA	Durability	0
4	1	NA	NA	Abe Sapien	Male	blue	0	NA	NA
5	1	super_hero_powers	1	NA	NA	NA	NA	Agility	0
6	1	super_hero_powers	2	NA	NA	NA	NA	Accelerated Healing	0
7	2	NA	NA	Abin Sur	Male	blue	0	NA	NA
8	2	super_hero_powers	1	NA	NA	NA	NA	Lantern Power Ring	0

Supertibble

	redcap_form_name	redcap_form_label	redcap_data	redcap_metadata	structure	data_rows	data_cols	data_size	data_na_pct
1	heroes_information	Heroes Information	*1 variable*	*1 variable*	nonrepeating	734	12	123.24 kB	0.23%
2	super_hero_powers	Super Hero Powers	*1 variable*	*1 variable*	repeating	5966	4	180.21 kB	0%

Data Tibble (nonrepeating instrument)

	record_id	name	gender	eye_color	form_status_complete
1	0	A-Bomb	Male	yellow	Incomplete
2	1	Abe Sapien	Male	blue	Incomplete
3	2	Abin Sur	Male	blue	Incomplete

Data Tibble (repeating instrument)

	record_id	redcap_repeat_instance	power	form_status_complete
1	0	1	Accelerated Healing	Incomplete
2	0	2	Durability	Incomplete
3	0	3	Longevity	Incomplete
4	1	1	Agility	Incomplete
5	1	2	Accelerated Healing	Incomplete
6	1	3	Cold Resistance	Incomplete
7	2	1	Lantern Power Ring	Incomplete

Figure 1: The REDCapTidieR Supertibble

Figure 1: The REDCapTidieR supertibble shown in the Data Viewer of the RStudio IDE. The "Superhero database" (ter Lingen, 2023) contains two instruments, one nonrepeating and one repeating. A. The REDCap API outputs a "Block Matrix". Note an abundance of `NA` values, which do not represent missing values but rather fields that do not apply due to the data structure. B. The `read_redcap()` function returns a "Supertibble". Note that each row represents one instrument, identified by the `redcap_form_name` column. The `redcap_data` column is a list column that links to tibbles containing the data from a specific instrument. The Data Viewer allows drilling down into individual tibbles by clicking on the table icon, allowing for rapid and intuitive data exploration without any preprocessing. Since each instrument has a consistent granularity, these tibbles can be tidy. Two data tibbles are shown, one from a nonrepeating and one from a repeating instrument. Note the differences in granularity between the instruments.

`REDCapTidieR` provides utility functions to work with the supertibble, all designed to work with the R pipe operator `|>`. The `extract_tibble()` function takes a supertibble object and returns a specific data tibble. The `make_labelled()` function leverages the `labelled` package (Larmarange, 2023) to apply variable labels to the supertibble. The `add_skimr_metadata()` function uses the `skimr` package (Waring et al., 2023) to add summary statistics. Using the `write_redcap_xlsx()` function, which leverages the `openxlsx2` (Barbone & Garbuszus, 2023) package, users can easily export an the supertibble into a collaborator-friendly Excel document, in which each Excel sheet contains the data for an instrument.

`REDCapTidieR` cannot be used to write data to a REDCap project. We refer the reader to an excellent guide of how to accomplish this using `REDCapR` (Beasley & Balise, 2023).

Installation

`REDCapTidieR` is available on GitHub and CRAN and works on all major operating systems.

Acknowledgements

We would like to thank Jan Marvin and Raymond Balise for their feedback and support in development.

This package was developed by the Cell and Gene Therapy Informatics Team of the Children's Hospital of Philadelphia.

Conflict of interest

The authors declare no financial conflicts of interest.

References

Balise, R., Odom, G., Calderon, A., Bouzoubaa, L., DeFreitas, W., & Grealis, K. (2023). *tidyREDCap: Helper functions for working with 'REDCap' data*. https://raymondbalise.github.io/tidyREDCap/index.html

Barbone, J. M., & Garbuszus, J. M. (2023). *openxlsx2: Read, write and edit 'xlsx' files*. https://janmarvin.github.io/openxlsx2/

Beasley, W. (2023). *REDCapR: Interaction Between R and REDCap*. https://ouhscbbmc.github.io/REDCapR/

Beasley, W., & Balise, R. (2023). *Writing to a REDCap project*. https://ouhscbbmc.github.io/REDCapR/articles/workflow-write.html

Carmezim, J., Peñafiel, J., Satorra, P., García, E., Pallarés, N., & Tebé, C. (2023). *REDCapDM: 'REDCap' data management*. https://bruigtp.github.io/REDCapDM/

Garbett, S., Nutter, B., Lane, S., Beasley, W., Horner, J., Stephens, J., Lehr, M., Beck, C., & Obregon, S. (2024). *redcapAPI: Accessing data from REDCap projects using the API*. https://doi.org/10.5281/zenodo.10564837

Hanna, R., Porter, E., & Kadauke, S. (2023). *REDCapTidieR*. https://chop-cgtinformatics.github.io/REDCapTidieR/index.html

Harris, P. A., Taylor, R., Minor, B. L., Elliott, V., Fernandez, M., O'Neal, L., McLeod, L., Delacqua, G., Delacqua, F., Kirby, J., & Duda, S. N. (2019). The REDCap consortium: Building an international community of software platform partners. *Journal of Biomedical Informatics*, *95*, 103208. https://doi.org/10.1016/j.jbi.2019.103208

Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., & Conde, J. G. (2009). Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of Biomedical Informatics*, *42*(2), 377–381. https://doi.org/10.1016/j.jbi.2008.08.010

Larmarange, J. (2023). *labelled: Manipulating labelled data*. https://larmarange.github.io/labelled/

Müller, K., & Wickham, H. (2023). *tibble: Simple data frames*. https://tibble.tidyverse.org/

Open Source Security Foundation. (2023). *OpenSSF Best Practices badge program*. The Linux Foundation. https://www.bestpractices.dev/

R Core Team. (2020). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

ter Lingen, J. (2023). *Superhero database*. https://www.superherodb.com/

Waring, E., Quinn, M., McNamara, A., Arino de la Rubia, E., Zhu, H., & Ellis, S. (2023). *skimr: Compact and flexible summaries of data*. https://docs.ropensci.org/skimr/

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10), 1–23. https://doi.org/10.18637/jss.v059.i10