Recommendations from the KnowMore Team to increase FAIRness of SPARC datasets

Matthew A. Schiefer[1, 2, 3], Ryan Quey[4], Anmol Kiran[5,6], Bhavesh Patel[7,*]

Affiliations

Anant Corporation, Washington, D.C., USA

2 Malcom Randall VA Medical Center, Gainesville, FL, USA

3 Department of Biomedical Engineering, University of Florida, Gainesville, FL, USA

4 SimNeurix, LLC, Gainesville, FL, USA

5 Malawi-Liverpool-Wellcome Trust, Blantyre-3, Malawi

6 Institute of Infection, Veterinary & Ecological Sciences, University of Liverpool, Liverpool CH64 7TE, UK

7 California Medical Innovations Institute, San Diego, CA, USA

*Email: bpatel@calmi2.org

Introduction

As part of the SPARC Codeathon 2021, our team has developed KnowMore, an automated knowledge discovery tool for finding connections across datasets that could lead to new discoveries or at least guide researchers towards it. An important aspect of having a fully automated one-click discovery tool is datasets with standard data structure that are not only human but machine understandable such that cross-data analysis could be automated. SPARC Datasets, all complying with the SPARC Data Structure (SDS), are achieving that to a great extent. Yet, during the development of KnowMore, we have identified several areas of improvement that could enhance the capabilities of automated analysis tools for making discoveries from SPARC datasets. Several recommendations are provided accordingly in this document to increase the FAIRness of SPARC datasets, especially with regards to the Interoperability aspects. We aim that they will guide future updates of the SDS.

Overview of Recommendations

1. Standardize common data fields terminology in tabular data

Problem: Currently, there are no guidelines on labeling fields in a tabular data file nor there is guidance on how/where they should be defined. This limits critically automated data comparison across datasets.

Proposed solution: Standard terminology and units must be imposed for common labels + each nonstandard label must be defined in a csv file in the same folder location as the data and named ‘data_file_name_terminology.csv’ where ‘data_file_name’ is the name of the actual data. Further details follow this section.

2. Standardize tabular data format

Proposed solution: Currently, there is no guidelines on the files format to use for tabular data so all types are used in SPARC datasets (csv, json, xlsx, mat, etc.). This limits critically automated data comparison across datasets.

Proposed solution: Raw tabular data may be in any user-defined format, but a derivative should be included in a SPARC-imposed standard open source format (json).

3. **Prevent data enhancing with external data**

Problem: We noticed that some of the datasets include the same data several times. This leads to a false sense of strong data correlation during automated analysis.

Proposed solution: Data in a dataset must only contain novel data, not available in any other published datasets (especially other SPARC datasets), and correspond to the subjects and samples listed in the subjects/samples metadata files.

Details

Standardized common data fields terminology in tabular data

- Data that lends itself to tabular format should be stored as such:
    - Headers (Variable Names)
        - Headers should be able to be interpreted as variable names.
            - Headers should not contain spaces or special characters that would not be acceptable as variable names.
            - Headers should not start with numbers.
            - Headers should start with a capital letter and if words are joined together, each word should start with a capital letter. For example, NerveDiameter.
        - Headers should be standardized for recurring data types across SPARC. We recommend looking through current SPARC datasets to determine what data are common and advising the community on recommended names.
    - Header Metadata
        - Metadata for each header should exist that describes that header (variable).
            - **Data Type**: For every column in the table, there should be metadata that describes the type of data encountered in that column. We recommend the following strings to classify each column:
                - For quantitive data:
                    - "discrete": e.g. number of parts in order
                    - "continuous": e.g., length, weight
                - For qualitative/categorical data:
                    - "nominal": e.g., marital status, ethnicity
                    - "ordinal": e.g. letter grade
                    - "binary": e.g., pass/fail, yes/no
            - **Data Format**: For every column in the table, there should be a code that describes how the data in that column should be handled. We

recommend using common I/O formatting codes. but propose two additional codes to specify that data is in time format:

Value Type	Data Format	Details
Integer, signed	%d or %I	Base 10
Integer, unsigned	%u	Base 10
	%o	Base 8 (octal)
	%x or %X	Base 16 (hexadecimal)
Floating Point	%f	Fixed-point
	%e or %E	Exponential notation
	%g or %G	Compact exponential notation
Characters or String	%c	Single character
	%s	Character vector or string array
Time	%r	Relative time (from start)
	%t	Absolute time, stored as yyyy-mm-dd HH:MM:SS.FFF

- **Data Units**: For every column in the table, there should be a means of specifying units. We recommend that this be a string of commonly accepted units. For example, “mm” for millimeters or “um” for micrometers. Only SI units should be used. More complicated units resulting from mathematical manipulation of data can be custom strings.

- Observational data
  - This is the primary data table. It contains all observations of whatever was measured.
- Per-Subject data: Data for which there is only one observation per subject
  - This can be an optional table but it allows for comparisons of subjects across studies. For example, each subject, no matter how many measures were made nor when they were made, would only have one date of birth. Unless the date of birth is a critical factor for the observational data, it is probably more appropriate to list it in this table.
- Missing data
  - Recommendations on how to properly code missing data?