SOFTWARE TOOL ARTICLE

KnowMore: an automated knowledge discovery tool for the FAIR SPARC datasets [version 1; peer review: 2 approved with reservations]

Ryan Quey¹, Matthew A. Schiefer²⁻⁴, Anmol Kiran^{5,6}, Bhavesh Patel ⁷

¹Anant Corporation, Washington, DC, USA

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

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

⁴SimNeurix, LLC, Gainesville, FL, USA

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

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

⁷FAIR Data Innovations Hub, California Medical Innovations Institute, San Diego, CA, 92122, USA

V1 First published: 09 Nov 2021, 10:1132
<https://doi.org/10.12688/f1000research.73492.1>
 Latest published: 09 Nov 2021, 10:1132
<https://doi.org/10.12688/f1000research.73492.1>

Abstract

Background: This manuscript provides the methods and outcomes of KnowMore, the Grand Prize winning automated knowledge discovery tool developed by our team during the 2021 NIH SPARC FAIR Data Codeathon. The National Institutes of Health Stimulating Peripheral Activity to Relieve Conditions (NIH SPARC) program generates rich datasets from neuromodulation researches, curated according to the Findable, Accessible, Interoperable, and Reusable (FAIR) SPARC data standards. Currently, the process of simultaneously comparing and analyzing multiple SPARC datasets is tedious because it requires investigating each dataset of interest individually and downloading all of them to conduct cross-analyses. It is crucial to enhance this process to enable rapid discoveries across SPARC datasets.

Methods: To fill this need, we created KnowMore, a tool integrated into the SPARC Portal that only requires the user to select their datasets of interest to launch an automated discovery process. KnowMore uses several SPARC resources (Pennsieve, o²S²PARC, SciCrunch, protocols.io, Biolucida), data science methods, and machine learning algorithms in the back end to generate various visualizations in the front end intended to help the user identify potential similarities, differences, and relations across the datasets. These visualizations can lead to a new discovery, new hypothesis, or simply guide the user to the next logical step in their discovery process.

Results: The outcome of this project is a SPARC portal-ready code architecture that helps researchers to use SPARC datasets more efficiently and fully leverages their FAIR characteristics. The tool has been built and documented such that more data analysis methods

Open Peer Review

Approval Status ? ?

	1	2
version 1
09 Nov 2021	view	view
1. Thomas Donoghue , Columbia University, New York, USA 2. Ammar Ammar , Maastricht University, Maastricht, The Netherlands		
Any reports and responses or comments on the article can be found at the end of the article.		

and visualization items could be easily added.

Conclusions: The potential for automated discoveries from SPARC datasets is huge given the unique SPARC data ecosystem promoting FAIR data practices, and KnowMore has only demonstrated a small highlight of what could be achieved to speed up discoveries from SPARC datasets.

Keywords

Data standards, Data Science, Metadata, Natural Language Processing, Knowledge graph, Cloud computing, Python

This article is included in the **Research on Research, Policy & Culture** gateway.

This article is included in the **Data: Use and Reuse** collection.

This article is included in the **Hackathons** collection.

This article is included in the **Python** collection.

Corresponding author: Bhavesh Patel (bpatel@calmi2.org)

Author roles: **Quey R:** Data Curation, Investigation, Methodology, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Schiefer MA:** Investigation, Methodology, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Kiran A:** Investigation, Methodology, Software, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Patel B:** Conceptualization, Data Curation, Investigation, Methodology, Project Administration, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2021 Quey R *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Quey R, Schiefer MA, Kiran A and Patel B. **KnowMore: an automated knowledge discovery tool for the FAIR SPARC datasets [version 1; peer review: 2 approved with reservations]** F1000Research 2021, 10:1132 <https://doi.org/10.12688/f1000research.73492.1>

First published: 09 Nov 2021, 10:1132 <https://doi.org/10.12688/f1000research.73492.1>

Introduction

The National Institutes of Health's (NIH's) Stimulating Peripheral Activity to Relieve Conditions (SPARC) program seeks to accelerate the development of therapeutic devices that modulate electrical activity in nerves to improve organ function.¹ A major focus of the SPARC program is to generate rich datasets that provide resources for understanding nerve-organ interactions and guiding the development of neuromodulation therapies. These datasets are publicly available through an open data platform, the SPARC Data Portal.² As of July 2021, 115 datasets are available spanning multiple scales (cellular, tissue, organ level), organs (stomach, large intestine, small intestine, heart, bladder, urinary tract, lung, pancreas, spleen), species (pig, human, rat, mouse, dog), and data types (scaffold data, histology, immunohistochemistry, electrical impedance tomography, 3D microscopy, morphometric analyses, computer simulations of single axons or populations of axons, electrophysiological responses to electrical stimulation, etc.).

To ensure SPARC datasets are findable, accessible, interoperable, and reusable (FAIR), they are curated according to the SPARC Data Structure (SDS), the data standards designed by the SPARC Data Curation Team to capture the large variety of data generated by SPARC investigators.^{3,4} Accordingly, many resources are made available to SPARC researchers for making their data FAIR, such as the cloud data platform Pennsieve, the curated vocabulary selector and annotation platform SciCrunch, the open source computational modeling platform Open Online Simulations for Stimulating Peripheral Activity to Relieve Condition (o²S²PARC), the online microscopy image viewer Biolucida, and the data curation Software to Organize Data Automatically (SODA).^{5,6} As a result, the SPARC program provides a wealth of open and well-curated datasets that are accessible via the SPARC Data Portal. The portal provides several means of accessing data. A standard portal search feature is available. Alternatively, the user can find datasets by browsing through data categorized by organ system. The user can also use an interactive map to click on organs or nerves in animal models of interest and the portal provides links to associated datasets. These pathways make it easy to find datasets. Clicking a link to a dataset provides the user with details about the study and options to download all or subsets of the data files.

While it is very easy to look at the details of any single SPARC dataset on the portal, there is currently no easy way to rapidly compare multiple datasets. Typically, a researcher wanting to find relations across datasets would have to do so manually by going through each dataset individually, i.e., read the description of each dataset, go through each protocol, browse files that are accessible from the browser, etc. Datasets that warrant further investigation must be downloaded for offline analyses and payment may be required for access to large datasets, according to Amazon Web Services (AWS) pricing. Depending on the formats of the data, this may require programming skills beyond that of many users. After spending time collating data in a form that allows comparison across the different datasets, the user may find that, in fact, the datasets did not contain the information they needed. This process of analyzing multiple datasets together is tedious, which ideally should not be the case since the SDS is designed to facilitate such analysis. Therefore, this process needs to be urgently improved to 1) enable rapid discoveries across SPARC datasets and 2) encourage more researchers to use the SPARC Data Portal.

To address this shortcoming, we developed KnowMore during the 2021 SPARC FAIR Codeathon⁷ (July 12th, 2021 – July 26th, 2021). KnowMore is an automated knowledge discovery tool integrated within the SPARC Portal. With minimal clicks, the user selects datasets of interest and KnowMore allows the user to visualize potential relations, similarities, differences, and correlations between the studies and associated datasets. This process, illustrated in **Figure 1**, is achieved by leveraging our knowledge of the SPARC data structure and metadata that allows us to perform text mining, generate a summary table, and plot data that is common across all selected datasets. The results are presented as several visualization items that provide the user with a quick means of identifying potential relations across the datasets.

Figure 1. Illustration of the simple user side workflow of KnowMore. Note that the tool is not currently integrated in the official SPARC Portal, but accessible through our fork of the sparc-app. It could be included into the official SPARC Portal after future consultation with SPARC.

This manuscript describes the structure of KnowMore and provides an example of knowledge provided by the tool when applied to a set of three sample datasets that constituted our use case for demonstrating KnowMore during the 2021 SPARC FAIR Codeathon.

Methods

Software architecture

The overall workflow of KnowMore is shown in Figure 2. Our architecture consists of three main blocks that are independent:

1. The front end of our app is based on a fork of the sparc-app (i.e. the front end of sparc.science) where we have integrated additional user interface elements and front end logic for KnowMore.⁸
2. The back end consists of a Flask (a micro web framework written in Python programming language) application that listens to front end requests and launches the data processing jobs.
3. The data processing and result generation are done through a MATLAB code (for ‘MAT’ data files) and a Python code (all other data types) that both run on the o²S²PARC platform, the SPARC supported cloud computing platform.⁹

In our front end of the sparc-app, we have integrated an “Add to KnowMore” button that is visible in the search result for each dataset and also available on the dataset page. By clicking on this button, the user can add their desired datasets for the analysis. Once all the datasets have been added, the user can go to the “KnowMore” tab we have included. On that page, the user can see a list of the selected datasets as well as a “Discover” button. A click on that button initiates the discovery process, where the Pennsieve IDs (i.e., the unique ID attributed to each dataset on sparc.science) of the selected datasets are sent to the Flask server, which then sends the IDs and our data processing Python script to o²S²PARC, using the o²S²PARC application programming interface (API).¹⁰ Once the script is fully executed, the results are sent back to the Flask server, which then transfers them to the front end where visualization items are generated. More details about the visualization items are provided in the next section.

The software architecture shown in Figure 2 was motivated by our aim of making KnowMore ready to on-board the SPARC Data Portal:

- Integrating the front end of KnowMore will only require merging our fork of the sparc-app with the main branch sparc-app.

Figure 2. Illustration of the overall technical workflow of KnowMore. The red rectangles highlight the major code blocks of KnowMore that were developed during the 2021 SPARC FAIR Codeathon.

- The back end of the sparc-app, the sparc-api, is built with Flask so the KnowMore back end is readily integrable.¹¹
- The data processing jobs are designed to run on o²S²PARC and do not require any type of integration as our back end ensures communication with o²S²PARC.

Moreover, each of the three main elements of KnowMore is fully independent. While the front end will not be of much use on its own, having the back end fully interoperable is very valuable as our Flask application can be connected to any front end if needed (another analysis tool, website, software, etc.). The data processing and results generation jobs are also independent such that they can be used directly to get the visualization items. We have demonstrated that by developing a Jupyter Notebook that communicates directly with o²S²PARC to run the knowledge discovery jobs based on user-specified dataset IDs. Note that the data for the Knowledge Graph is obtained from Pennsieve/Scicrunch on the front end for efficiency but the same results can be generated in the back end as well. Thorough details for using the source code are available on the GitHub repository for this project.¹²

Data processing and outputs

The output of KnowMore consists of multiple interactive visualization items displayed to the user so that they can progressively gain knowledge on the potential similarities, differences, and relations across the datasets. This output is intended to provide foundational information to the user so that they can rapidly make novel discoveries from SPARC datasets, generate new hypotheses, or simply decide on their next step (assess each dataset individually on the portal, download and analyze the datasets further, remove/add datasets to their analysis pool, etc.). A list of the visualization items is provided in Table 1, along with the potential knowledge that could be gained from each of them.

The process of getting these outputs starts by getting the IDs of the datasets selected by the user, which are obtained using the Pennsieve API¹³ in the front end. From there, we leverage several SPARC-supported and recommended resources in our data processing Python Script to collect the raw data required to generate the above-mentioned outputs. These resources include the Pennsieve API,¹³ the Scicrunch Elasticsearch API,¹⁴ the protocols.io API,¹⁵ and the Biolumina API.¹⁶ We refer to the paper on the SPARC Data Resource Center (DRC) for more details about these resources and their role in the SPARC data ecosystem.⁶ Details about each of the visualization items are provided below. Each of these items can be easily saved from the front-end interface.

Knowledge graph

Using the Pennsieve ID of each dataset, the following items are queried from SciCrunch Elasticsearch API¹⁴: Person (authors of the dataset), Affiliation (affiliation of the authors), and Award (funding source for the dataset). The visualization library Vega is used in the front end to display this information in an interactive knowledge graph, which instantly highlights high-level relations amongst the datasets.

Table 1. Table listing the visualization items automatically generated by KnowMore. The source of the raw data for generating the visualization items are also listed.

Visualization item	Knowledge gained across the datasets	Raw data used for generating the visualization and how it was obtained
Knowledge graph	High-level connections (authors, institutions, funding organisms, etc.)	Dataset metadata obtained with the Pennsieve API and the SciCrunch Elasticsearch API
Summary table	Similarities/differences in the study design	Dataset metadata from the metadata.json file obtained with the Pennsieve API
Common keywords	Common themes	Dataset metadata from the metadata.json file and all dataset text files obtained with the Pennsieve API. Protocol text obtained with the protocols.io API
Abstract	Common study design and findings	Dataset metadata from the metadata.json file and all dataset text files obtained with the Pennsieve API. Protocol text obtained with the protocols.io API
Data plots	Comparison between measured numerical data (if any)	MAT files in the derivative folder of the datasets obtained with the Pennsieve API

Summary table

A summary table is built with information collected from the metadata.json file of each dataset, which is a standard file generated for each SPARC dataset when published, and the subjects and samples metadata files, which are standard metadata files prescribed for SPARC datasets by the SDS. The files are retrieved from the Pennsieve API within our Python code. The following items are parsed from the metadata.json file for each dataset: title of the dataset, subtitle of the dataset, publication date. The following items are parsed from the subjects' metadata file for each dataset: number of subjects, species, age, sex. The following items are parsed from the samples metadata file: number of samples, specimen, type, specimen anatomical locations. The visualization library Plotly is used in the front end to display the results in an interactive table, which shows this information side-by-side for each dataset, thus enabling quick comparison in the study design of each dataset.

Keywords

Text is obtained for each dataset from the description included in metadata.json file and the text from all the text files in the dataset using the Pennsieve API, and the text from the protocol on protocols.io associated with the dataset using the protocols.io API. The link to the protocol.io protocol is extracted from the metadata.json file of the dataset. All text is combined to create a paragraph for each dataset. The Natural Language Processing (NLP) Python library NLTK¹⁷ is then used to clean the text (e.g., remove stopwords). Biological keywords are identified using the spaCy python module and ScispaCy models.^{18,19} The frequency of biological words is counted for each dataset. The final frequency of the keywords is assigned based on lowest occurrence among the datasets and the twenty most frequent words are selected and displayed as a word cloud using the visualization library Vega. The minimum frequency of a keyword across the dataset is displayed when the cursor hovers over the word. These keywords conveniently allow the user to identify common themes across the datasets.

Correlation matrix and abstract

The correlation matrix demonstrates the putative relatedness between datasets.^{20,21} To generate the correlation matrix for the given datasets, pairwise similarity between datasets is calculated using the following Jaccard index equation²²:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

where A and B are sets of biological keywords present in two datasets. The biological keywords are identified as explained earlier in the Keywords section.

Paragraphs generated from datasets for the keywords identification are merged and divided into sentences. Each sentence is further divided into words and stopwords were removed. The frequency of each remaining word in a sentence is counted and converted into vectors where keywords represent the direction and frequencies represent the magnitude. The distance of two sentences is calculated using equation $1 - \cos(\theta)$ where cosine similarity is expressed as follows:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

where A and B are words frequency in vectors of two sentences. Based on the pairwise distance of sentences a pagerank is assigned to each sentence using Python networkX module and sentences are ordered based on pagerank in decreasing order.²³ The top 10 highest-ranked sentences are selected to generate a common abstract for the datasets. This abstract is intended to provide a quick idea of any common study design and/or findings.

Data plots

If data files in.mat format are found under the “derivative” folder, the data processing Python script extracts and saves them then provides them to our MATLAB script that is compiled and deployed on o²S²PARC. The script collates the data into a data table. The script next determines which columns in the data table can be used for plotting purposes. Columns containing categorical data are limited to the x-axis. Columns containing numerical data can be plotted on the x-axis or y-axis. Columns containing any other type of data are excluded. Plots are then generated for every variable that can be displayed on a y-axis against every variable that can be plotted on an x-axis. In addition to the plots, the MATLAB script outputs an Excel file that lists each of the plots created and the variables included in each plot. The Excel file also includes

data for each plot. Additionally, the script creates a json file that includes all data for each plot. These plots quickly highlight to the user relations between similar quantities measured across datasets.

Image clustering

An additional visualization item we aimed to provide to the user but could not complete during the Codeathon due to time constraints was a clustering of images across datasets, which may be particularly useful for histological data. All image data from SPARC datasets are stored on Biolucida. We currently have a function in place to retrieve image data from Biolucida given a Pennsieve dataset ID using the Biolucida API.¹⁶ In the future, image clustering and visualization components will be added in the Python script and front end, respectively, to provide an additional element to the user for comparing datasets.

Use case

Setup

KnowMore was developed and tested using three datasets available at sparc.science (Table 2). These datasets were selected because they have a common theme – quantified vagus nerve morphology – and span three species: rat, pig, and human. In principle, KnowMore is not specifically designed around these datasets and is coded to work with any user-selected datasets. However, for demonstration purposes, the data plots are currently limited to only appear when working with all or a subset of the three datasets listed in Table 2. Reasons for this are addressed in the Challenges section below and recommendations are put forth to expand the usability of this feature and increase the interoperability of SPARC datasets.

Initiating a KnowMore analysis requires five steps:

1. Use the search feature or browse for possible datasets of interest at sparc.science.
2. As datasets are identified that the user wants to compare, click on the “Add to KnowMore” button, visible in the header of the datasets or the search results. This will add the datasets to the KnowMore analysis.
3. Go to the KnowMore tab at the top of the webpage and check that all of the desired datasets are listed.
4. Decide which output to display. All possible output is displayed by default.
5. Click on the “Discover” button to initiate the automated analysis.

The number of datasets selected will affect the duration of time required to run the full discovery analysis. The use case with these three datasets takes about 4 min to generate all the visualization items.

Outputs

Knowledge graph

The Knowledge graph provides an interactive tool to visualize metadata across the three datasets (Figure 3). This provides the ability to quickly determine, for example, that all three datasets had four investigators in common (Cariello, Grill, Goldhagen, and Pelot) affiliated with the Department of Biomedical Engineering at Duke and that the human dataset had additional investigators (Ezzell and Clissold) affiliated with the Department of Cell Biology and Physiology at the University of North Carolina.

Table 2. List of datasets used for our use case.

Pennsieve ID	Title
60	Quantified Morphology of the Rat Vagus Nerve ²⁵
64	Quantified Morphology of the Pig Vagus Nerve ²⁶
65	Quantified Morphology of the Human Vagus Nerve with Anti-Claudin-1 ²⁷

Figure 3. Knowledge Graph output for the three datasets in our use case.

Table 3. KnowMore summary table output for the three datasets in our use case.

Dataset ID	60	64	65
Title	Quantified Morphology of the Rat Vagus Nerve.	Quantified Morphology of the Pig Vagus Nerve.	Quantified Morphology of the Human Vagus Nerve with Anti-Claudin-1.
Subtitle	Binary traces from segmentation of cross sections of cervical and subdiaphragmatic rat vagus nerves stained with Masson's trichrome. Quantified effective nerve diameter, effective fascicle diameter, number of fascicles, and perineurium thickness.	Binary traces from segmentation of cross sections of cervical and subdiaphragmatic pig vagus nerves stained with Masson's trichrome. Quantified effective nerve diameter, effective fascicle diameter, and number of fascicles.	Immunohistochemistry micrographs of human vagus nerves labeled with anti-claudin-1. Binary traces from segmentation to quantify effective nerve diameter, effective fascicle diameter, number of fascicles, and perineurium thickness.
Publication date	2020-09-30	2020-10-01	2020-10-01
Number of subjects	10	11	15
Species	Rattus norvegicus	Sus scrofa domesticus	Homo sapiens
Age	75 days - 268 days	10.5 weeks - 15 weeks	54 years - 90+ years
Sex	Female, Male	Female, Male	Female, Male
Number of samples	18	18	20
Specimen type	Vagus nerve	Vagus nerve	Vagus nerve
Anatomical location(s)	Left cervical vagus nerve; 11 mm from carotid bifurcation, subdiaphragmatic vagus nerve; 8.5 mm from esophageal hiatus and 8.5 mm from gastroesophageal junction; hepatic branch 10 mm from esophageal hiatus.	Left cervical vagus nerve; 15 cm from bottom of jaw to top of sternum; sample middle ~2 cm; 6 cm from middle of sample to carotid bifurcation, left cervical vagus nerve; 13 cm from bottom of jaw to top of sternum; sample middle ~2 cm; 5 cm from middle of sample to carotid bifurcation.	Left cervical vagus nerve; 35 mm from carotid bifurcation, left cervical vagus nerve; 20 mm from carotid bifurcation.

Summary table

The Summary Table provides the user with key pieces of information from each study in tabular format (Table 3). From this table, the user can easily determine that datasets have several common metrics. However, perineurial thickness is not quantified in dataset 64.

Common keywords

The common keywords figure provides a graphical depiction of words that show up multiple times across the selected datasets (Figure 4). This size of the word in the image provides a visual representation of the weight (or frequency) of that word across the datasets. Not surprisingly, “nerve” is a large word as it shows up many times. Many other keywords highlight the quantified morphology across the datasets (diameter, cross-sectional area, fascicle, etc.).

Correlation matrix and abstract

KnowMore generates a heatmap illustrating the correlation between the studies based on the words used in the text of these studies (Figure 5). This figure can guide the user in selecting highly correlated studies or eliminating studies that do

Figure 4. Common keywords output for the three datasets in our use case.

Figure 5. Correlation of the words used to describe the three datasets in our use case.

not correlate well. Additionally, KnowMore generates a combined abstract that provides an overview of all datasets included in the study.

Data plots

For this use case, KnowMore also generates 20 scatter plots. Due to time constraints, these plots are currently generated in the backend as png files and then displayed in the front-end. Data points are color-coded to each dataset. Each axis is

Figure 6. Three selected KnowMore Data Plots created from the three datasets in our use case.

labeled with the variable being plotted. The variable name is obtained directly from the datasets. Three of the plots are presented here (Figure 6). Plot 3.4 reveals that pigs contain more fascicles in their vagus nerves than humans do, and humans contain more fascicles than rats. Plot 3.4 also reveals that pigs and rats have similar variability (spread) in their fascicle diameters whereas humans have a greater spread in their fascicle diameters. Finally, Plot 3.4 illustrates that humans can have larger fascicles than pigs. Plot 3.5 reveals that humans and pigs have similar-sized nerves, though pigs may, on average, have larger nerves. Plot 3.5 also reveals that the number of fascicles in the nerve may tend to be greater for nerves of larger diameter within each species. That is, there appears to be a positive correlation between the number of fascicles in the nerve and the diameter of the nerve. However, Plot 4.5 suggests that there may not be a trend between the fascicle diameter and the nerve diameter. Although these findings have been previously reported in some form,²⁴ the Data Plots can become a very useful tool in helping researchers quickly understand the underlying data across multiple datasets.

Conclusions and next steps

Potential for this tool

In a few clicks to select datasets, KnowMore can provide both a high-level metanalysis and a granular comparison across two or more datasets on the SPARC portal. KnowMore outputs result at several levels depending on the needs of the researcher. One can quickly determine personnel, institutional, and funding relationships between datasets, and generate an overview of subjects included in the datasets and the techniques used to obtain data. Finally, if data are suitable for plotting, plots can reveal relationships within and across the studies that may reveal larger trends or help the researcher choose or eliminate particular datasets for more detailed analysis.

Challenges

SPARC has done an excellent job of standardizing the metadata associated with a study, and, as such, most of the KnowMore output is available across any selected studies. However, SPARC has not enforced standardization for tabular data. As such, the Data Plot output of KnowMore is currently limited to datasets that contain identical variable names and formats. This is an uncommon occurrence across datasets. Data can currently be stored in any number of formats. KnowMore's Data Plot currently requires data to be stored in a MATLAB.mat file due to our use case, but this could be expanded to several other file formats. It would be preferable from a programming perspective if all data formats and variable naming are consistent, however, within MATLAB alone, data can be stored in multiple formats. Data may be stored in vectors/matrices; cells; cell arrays of vectors, matrices or more cells; structures; or tables, among other formats. Even small differences in variable names such as NerveDiam versus NerveDiameter versus DiameterOfNerve are not immediately reconcilable, though NLP may alleviate such inconsistencies. Without unified variable naming, comparisons across datasets become very challenging. Inconsistent variable names are not the only challenge, however. Even if variable names are identical, the values stored for that variable may be different from study to study. Without unified data types, comparisons across datasets become very challenging. To make the KnowMore Data Plot tool universal we propose standardization of commonly used variable names, data formats, data types, and data units. We also recommend the inclusion of key pieces of information that describe the data in the metadata. We have submitted these recommendations to SPARC and a copy of the document is available in our GitHub repository. This may require a significant amount of effort to convert previously uploaded datasets but should not put an exceptional burden on new studies. Data standardization across the SPARC platform would make the data ready for much broader analysis using more sophisticated big data tools that could provide insights that are otherwise obscured or not readily accessible.

Future directions

Currently, the discovery process takes several minutes to run and display the visualization items (about 5 min for the use case). To improve performance, we suggest using multi-threading in the Python script; moving the.mat file processing directly into the Python script; collecting all required raw data (e.g., text) when a dataset is uploaded (e.g. save it in the metadata.json file) and even pre-process it (clean the text) so it is readily available during our discovery process. Image clustering components can be included in the future as well as any other visualization items that are deemed useful to the user. If the above-mentioned challenges with tabular data are addressed, the Data Plots feature of KnowMore can be generalized to work with any datasets.

The SPARC data ecosystem that is built to deliver FAIR datasets, provides a unique opportunity to automate knowledge discovery across datasets. During this project, we leveraged that ecosystem to demonstrate what can be achieved to increase the speed and convenience of discoveries across SPARC datasets. The tool we have developed is a statement of the power of FAIR practices and the effort of SPARC in that regard. We believe that we have only scratched the surface during the Codeathon and the opportunities are yet immense.

Software availability

Source code available from: <https://github.com/SPARC-FAIR-Codeathon/KnowMore>.¹²

Archived source code at the time of publication: <https://doi.org/10.5281/zenodo.5137255>.²⁸

License: MIT

The repository and archive both contain detailed information for using the source code. They also contain a copy of our recommendation to SPARC for standardizing tabular data.

Acknowledgments

We would like to thank the NIH SPARC Program and the SPARC Data Resource Center (DRC) teams for organizing the 2021 SPARC FAIR Codeathon. We would also like to thank the DRC teams for their guidance and help during this Codeathon.

References

- National Institutes of Health: **Stimulating Peripheral Activity to Relieve Conditions (SPARC)**. (Accessed: 19th September 2020). [Reference Source](#)
- National Institutes of Health: **SPARC Data Portal**. (Accessed: 19th September 2020). [Reference Source](#)
- Wilkinson MD, et al.: **The FAIR Guiding Principles for scientific data management and stewardship**. *Sci. Data*. 2016; **3**: 160018. [PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Bandrowski A, et al.: **SPARC Data Structure: Rationale and Design of a FAIR Standard for Biomedical Research Data**. *bioRxiv*. 2021. 2021.02.10.430563. [Publisher Full Text](#)
- Patel B, Srivastava H, Aghasafari P, et al.: **SPARC: SODA, an interactive software for curating SPARC datasets**. *FASEB J*. 2020; **34**: 1–1.
- Osanlouy M, et al.: **The SPARC DRC: Building a Resource for the Autonomic Nervous System Community**. *Front. Physiol*. 2021; **0**: 929.
- 2021 SPARC FAIR Codeathon. (Accessed: 1st August 2021). [Reference Source](#)
- NIH SPARC.: **Web Application for the SPARC Portal**. (Accessed: 1st August 2021). [Reference Source](#)
- IT'IS Foundation: **Open Online Simulations for Stimulating Peripheral Activity to Relieve Conditions**. (Accessed: 1st August 2021). [Reference Source](#)
- IT'IS Foundation: **osparc API client**. (Accessed: 1st August 2021). [Reference Source](#)
- NIH SPARC: **SPARC Portal API**. (Accessed: 1st August 2021). [Reference Source](#)
- Patel B, Quey R, Schiefer M, et al.: **KnowMore: Automated Knowledge Discovery Tool for SPARC Datasets**. (Accessed: 1st August 2021). [Reference Source](#)
- Pennsieve: **Pennsieve API**. (Accessed: 1st August 2021). [Reference Source](#)
- FDI Lab: **SciCrunch ElasticSearch API**. (Accessed: 1st August 2021). [Reference Source](#)
- protocols.io: **protocols.io for Developers**. (Accessed: 1st August 2021). [Reference Source](#)
- MBF Bioscience: **Bioluminescence API v2021**. (Accessed: 1st August 2021). [Reference Source](#)
- Bird S, Klein E, Loper E: *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc; 2009.
- Honnibal, Matthew Montani I, Van Landeghem S, et al.: **spaCy: Industrial-strength Natural Language Processing in Python**. 2020. [Publisher Full Text](#)
- Neumann M, King D, Beltagy I, et al.: **ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing**. 2019; 319–327. [Publisher Full Text](#)
- Thakur N, Mehrotra D, Bansal A: **Information Retrieval System Assigning Context to Documents by Relevance Feedback**. *Int. J. Comput. Appl*. 2012; **58**: 37–47. [Publisher Full Text](#)
- Kotu V, Deshpande B: **Classification**. *Data Science - Concepts and Practice*. Morgan Kaufmann; 2019; 65–163. [Publisher Full Text](#)
- Jaccard P: **The distribution of the flora in the alpine zone**. *New Phytol*. 1912; **11**: 37–50. [Publisher Full Text](#)
- Hagberg AA, Schult DA, Swart PJ: **Exploring network structure, dynamics, and function using NetworkX**. *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Varoquaux G, Vaught T, Millman J, editors. 2008; 11–15.
- Pelot NA, et al.: **Quantified Morphology of the Cervical and Subdiaphragmatic Vagus Nerves of Human, Pig, and Rat**. *Front. Neurosci*. 2020; **0**: 1148.
- Pelot NA, Goldhagen GB, Cariello JE, et al.: **Quantified Morphology of the Rat Vagus Nerve (Version 4)**. 2020. [Publisher Full Text](#)
- Pelot NA, Goldhagen GB, Cariello JE, et al.: **Quantified Morphology of the Pig Vagus Nerve (Version 4)**. 2020. [Publisher Full Text](#)
- Pelot NA, et al.: **Quantified Morphology of the Human Vagus Nerve with Anti-Claudin-1 (Version 6)**. 2020. [Publisher Full Text](#)
- Quey R, Kiran A, Schiefer M, et al.: **KnowMore: v1.0.0 - Automated Knowledge Discovery Tool for SPARC Datasets (v1.0.0)**. *Zenodo*. 2021. [Publisher Full Text](#)

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 21 June 2022

<https://doi.org/10.5256/f1000research.77147.r139348>

© 2022 Ammar A. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ammar Ammar

Department of Bioinformatics—BiGCaT, NUTRIM, Maastricht University, Maastricht, The Netherlands

Introduction

The authors developed a software tool composed of three components (frontend, backend, and data processing scripts) to perform an analysis on a selected group of SPARC datasets by the user in order to visualize potential relations, similarities, differences, and correlations between the datasets. That is achieved by leveraging the SPARC data structure and metadata which allows us to perform text mining and different types of plots. The software extends an already available user interface to add knowledge discovery functionalities. Moreover, it is mainly a dashboard for data visualization which utilizes an API and the capabilities of a computational platform (o²S²PARC) besides several other resources to perform data analysis in the form of an asynchronous job-submission paradigm. The tool is useful and potentially will help increase the usability of SPARC datasets and the SPARC data portal by allowing users to discover dataset inter-relations and generate new insights from the combined analysis of multiple datasets.

The code is provided through GitHub, and versioned through Zenodo with a DOI. The requirements.txt provides specific package versions and the code functions are documented. Moreover, the analysis functionality can be used through Jupyter Notebooks without the GUI and the authors provided a full example (in the GitHub repository) on how to use it. The description of the software components and how it works is clear and sound.

Overall, the manuscript describes an important and useful system that address the current limitations resembled in the difficult comparison between SPARC datasets and extracting useful insights from them or decide on their (re)usability. I find the project, even as a prototype, valuable and serve the intended purpose and the manuscript fulfills describing the software and demonstrating its work. However, I have some comments and recommendation addressed below.

Major updates

I could not reproduce the example use case described in the manuscript. At first, step (1) under the “Setup” subtitle (page 7, of the pdf) says “Use the search feature or browse for possible datasets of interest at sparc.science” while KnowMore is not integrated into the main platform and alternatively provided as a Herokuapp demo which is not mentioned anywhere in the manuscript. I had to find it in the GitHub README file. The same misleading information is mentioned in Figure 1, the first thumbnail from the left. Next, after accessing the software using the Herokuapp link and selecting the three datasets, the results sections did not show up and the spinners kept spinning. Further investigation through the console and network tabs in the Google Chrome browser (latest version 102) revealed an HTTP error 500 for the call <https://sparc-know-more-api.herokuapp.com/api/start-osparc-job/> and the following error message:

```
osparc.exceptions.ApiException: (401) Reason: Unauthorized HTTP response headers:
HTTPHeaderDict({'Content-Length': '38', 'Content-Type': 'application/json', 'Date': 'Mon, 20 Jun 2022
23:56:35 GMT', 'Server': 'uvicorn', 'Vary': 'Accept-Encoding, Accept-Encoding'}) HTTP response body:
{"errors":["Invalid API credentials"]}
```

Therefore, I could not see the results mentioned in the manuscript.

The demo application should be at least working. Moreover, it would be helpful to provide an API credentials for testing purposes so users can try running the tool locally using Docker and Docker Compose. However, I could start the Flask backend server and got the response "status: up" on port 5000.

Minor updates

1. I could not find where to test the platform according to the use case section since the Herokuapp link is not mentioned anywhere in the manuscript. I recommend providing the test URL of the demo application under the “Setup” subtitle (Page 7), in step 1 and removing the currently mentioned URL “sparc.science” where the KnowMore tool is not integrated yet.
2. The plots in Figure 6 have no meaningful titles, it is not clear what is plotted in each plot. The plots need descriptive titles. It is true that the plots are described in the text but this will not be the case for real users when they will use the software.
3. In the GitHub repository, I see the code for the analysis provided under “assets/INPUT_FOLDER”. The naming makes no sense and gives no clue on why to put a main function script under such unrelated path. A better script files structuring can be used here to make the project hierarchy more understandable.
4. I see in the GitHub repository a PDF file describing the measurements taken to provide FAIR software according to (Lamprecht et al. 2020)¹. This is valuable information that should be described in the manuscript mentioning the measures taken to make the KnowMore tool FAIR.

Recommendations

The KnowMore tool authors took serious efforts to make the software FAIR, but the software analysis output itself is still not FAIR. For instance, the tool does not provide its output in a

machine-readable way and adopt a known controlled vocabulary. For example, the analysis results can be expressed (besides the visualization and the HTML output) as JSON-LD representation format that is injected in the HTML upon job completion. Furthermore, schema.org vocabulary can be used to describe the output of the comparison (to the extent that is supported by schema.org) which allows software agents to parse, understands and take actions based on the JSON-LD metadata in an automated way. The SPARC data portal main application (sparc.science) already does a remarkable FAIR job by using JSON-LD to describe the SPARC datasets metadata and uses globally unique identifiers for the datasets (DOI) and the creators (ORCID) besides provenance information like license and free-access status. So, it would be a great addition to KnowMore to adopt the same approach and provide such a machine-readable metadata to describe the dataset comparison and analysis results to make it more FAIR. Storing the analysis results and giving them a unique identifier for later retrieval is also a step toward more FAIR dataset comparison metadata.

References

1. Lamprecht A, Garcia L, Kuzak M, Martinez C, et al.: Towards FAIR principles for research software. *Data Science*. 2020; **3** (1): 37-59 [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, systems biology, data science, data analysis, FAIR, semantic web

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 11 March 2022

<https://doi.org/10.5256/f1000research.77147.r124523>

© 2022 Donoghue T. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Thomas Donoghue

Department of Biomedical Engineering, Columbia University, New York, USA

Overview

In this report, Quey et al develop a prototype software tool, KnowMore, to be used within the SPARC ecosystem of tools and datasets related to stimulating peripheral nerves and organs with the goal of improving organ function. The goal of KnowMore is to address the current limitation whereby it is difficult to directly compare different datasets available through the SPARC portal. Specifically, KnowMore is a data comparison dashboard, which allows for selecting different datasets, and then initializing a process which returns a series of visualizations and reports that can be used to compare between datasets. Notably, KnowMore was developed during a SPARC codeathon, such that what is described in this paper is a working prototype, but has not (yet?) been integrated into the main SPARC portal, and does not currently work across all available datasets.

Overall, I think this manuscript reports a useful and important development of a system to compare between datasets, which addresses a meaningful need for better resources for comparing and combining datasets. The tool is specific to the SPARC context, as is well motivated by this paper, including the use of multiple related tools within this ecosystem. Technically, I find no obvious issues with this tool – the paper describes the main strategies employed, the code is available, the I was able to run the example usage noted in the paper. In that context, I find this to be a useful and valuable project, and broadly a successful paper reflecting that. My main overall comments, detailed below, reflect that though I appreciate the prototype tool and discussion in this paper, I am left slightly unclear with what the goal of this particular version of the paper is – whether the authors intend this paper to reflect a tool people should use, in which case more of roadmap regarding generalizing the tool should perhaps be offered, or whether it's goal is more of a narrative description of “lessons learned” from attempting this kind of prototype, in which case further discussion on how to better support this kind of work in the future may be warranted.

Main comments:

1 – STATUS OF THE TOOL

This manuscript feels like it exists within two contexts, as it both introduces a new tool, and describes a project pursued during a hackathon. While this is clearly a sensible combination for the realities of this project, at times it is unclear if the reader should be reading this as a report on a new tool that they can go and use, and/or if this report is a narrative description of an example application in the general context of comparing between datasets, that future work should learn from. To the extent it's more tool oriented, then it's a limitation that this paper describes a prototype that, to my understanding, only works on a small number of datasets. As such, although the prototype is compelling, it is unclear if this is an actionable tool for people to generally use. In

this context, it feels like a limitation that the paper does not clearly describe the roadmap for if / to what extent the tool will continue to be developed in order to become more generally useable. For example, it seems unclear if there will be development on this tool post-codeathon, and whether there is a plan for this tool to be merged with SPARC, or if and how people should try to use this tool in the interim. For example, if the goal is for people to use this, a possible short-term update would be to extend the tool to make the other visualizations that are not dependent on the data fields, such as the knowledge graph, work across all possible datasets. As far as I can tell, this does not currently work, limiting the actionable usage of this tool. If the tool is expected to be supported, and developed further, this could be made more clear.

2 – STANDARDIZED DATA FORMATS

To the extent that this paper represents a narrative description of the “lessons learned” from working on this project in the context of a codeathon, then I think more could be said about the challenges that arose in order to present some useful commentary for future work on this topic. In particular, this paper notes that a major hurdle that was encountered in generalizing this tool is the lack of data standardization. It feels like the magnitude of this issue is perhaps understated. Data standardization is a large and difficult issue, and if one of the goals of this paper is to address issues such as this that arise in this kind of project, then perhaps this can be discussed further, both in terms of recognizing the scale of the issue at hand, and noting more details on what would need to happen for this to be properly addressed. As I work in a different field, I can’t speak to related discussion in the context of SPARC datasets, but in other areas, these are topics of extended debate, including the need to develop clear ontologies of terms that are accepted and used by the community (example: Yarkoni et al, 2019), and the need for standardized data files and formats that then embody these names, and associated technical tools for file I/O, validation, comparison, etc (example: Gorgolewski et al, 2016). I presume there is likely similar / related work more topical to the data under study here, that could be cited and further discussed. If a goal of this paper is to provide insights for future work on this topic, then more details in terms of what would be needed for the next generation of this tool would be valuable, as well as notes on relevant work and whether there are plans for these topics to be addressed within the SPARC ecosystem.

3 – GOAL OF THE PAPER

To conclude, and integrate across the prior notes, I think an overall, potential update to the paper would be for the authors to explicitly consider and update the paper in terms of what they wish the reader to take away from this report. This might entail clarifying the status and future of the tool itself, and/or digging into this project as an example for future work. To be clear, I don’t think this requires updating the report with respect to everything I mentioned – it may be more sensible to choose an approach and focus on that. I also want to clarify these are potential ideas that may be useful, but aren’t meant to imply that I find anything explicitly of issue with the current draft, but that some minor amendments would serve to better describe and inform the reader of the status of the associated tool. Finally, I’d like to note that I think this kind of work is very important, but also difficult to do and often under-valued, so I would like to commend the authors on pursuing this kind of project.

Signed,
Thomas Donoghue, PhD

References

1. Yarkoni T, Eckles D, Heathers J, Levenstein M, et al.: Enhancing and Accelerating Social Science Via Automation: Challenges and Opportunities. *Harvard Data Science Review*. 2021. [Publisher Full Text](#)
2. Gorgolewski KJ, Auer T, Calhoun VD, Craddock RC, et al.: The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci Data*. 2016; **3**: 160044 [PubMed Abstract](#) | [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Neuroscience; data science; Python programming; data analysis

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research