Computer Methods and Programs in Biomedicine 226 (2022) 107111

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Development of a REDCap-based workflow for high-volume relational data analysis on real-time data in a medical department using open source software[☆]

Guido Rovera[a,*], Piero Fariselli[b], Désirée Deandreis[a]

[a] Department of Medical Sciences, Nuclear Medicine, AOU Città della Salute e della Scienza di Torino, University of Turin, Corso Bramante 88, Turin 10126, Italy

[b] Department of Medical Sciences, Computational Biomedicine, AOU Città della Salute e della Scienza di Torino, University of Turin, Turin, Italy

ARTICLE INFO

Article history:
Received 16 April 2022
Revised 27 July 2022
Accepted 4 September 2022

Keywords:
Database management systems
Data collection
Data analysis
Workflow

ABSTRACT

Background/Aim: The current availability of large volumes of clinical data has provided medical departments with the opportunity for large-scale analyses, but it has also brought forth the need for an effective strategy of data-storage and data-analysis that is both technically feasible and economically sustainable in the context of limited resources and manpower. Therefore, the aim of this study was to develop a widely-usable data-collection and data-analysis workflow that could be applied in medical departments to perform high-volume relational data analysis on real-time data.

Methods: A sample project, based on a research database on prostate-specific-membrane-antigen/positron-emission-tomography scans performed in prostate cancer patients at our department, was used to develop a new workflow for data-collection and data-analysis. A checklist of requirements for a successful data-collection/analysis strategy, based on shared clinical research experience, was used as reference standard. Software libraries were selected based on widespread availability, reliability, cost, and technical expertise of the research team (REDCap-v11.0.0 for collaborative data-collection, Python-v3.8.5 for data retrieval and SQLite-v3.31.1 for data storage).

The primary objective of this study was to develop and implement a workflow to: a) easily store large volumes of structured data into a relational database, b) perform scripted analyses on relational data retrieved in real-time from the database. The secondary objective was to enhance the strategy cost-effectiveness by using open-source/cost-free software libraries.

Results: A fully working data strategy was developed and successfully applied to a sample research project. The REDCap platform provided a remote and secure method to collaboratively collect large volumes of standardized relational data, with low technical difficulty and role-based access-control. A Python software was coded to retrieve live data through the REDCap-API and persist them to an SQLite database, preserving data-relationships. The SQL-language enabled complex datasets retrieval, while Python allowed for scripted data computation and analysis. Only cost-free software libraries were used and the sample code was made available through a GitHub repository.

Conclusions: A REDCap-based data-collection and data-analysis workflow, suitable for high-volume relational data-analysis on live data, was developed and successfully implemented using open-source software.

1. Introduction

The availability of large volumes of clinical data has provided clinicians with the opportunity to perform larger scale analyses, which can generate higher levels of evidence and improve scientific research. Moreover, the application of data science in the medical field is also paving the way towards a more personalized medicine in the future, thanks to innovative approaches such as artificial intelligence and machine learning techniques [1].

☆ This strategy could provide medical departments with a sustainable solution to further improve scientific research, streamlining time-consuming tasks and facilitating future multi-center collaborations.

* Corresponding author.
E-mail address: guido.rovera@unito.it (G. Rovera).

https://doi.org/10.1016/j.cmpb.2022.107111

Table 1
Checklist of requirements for an effective data-collection and data-analysis strategy.

1. Data Entry
A- Allow users without specific technical training to collect data in a standardized format
B- Store data efficiently, preserving the information about data relationships
C- Allow for multi-user collaboration, with role-based access to sensitive information
D- Facilitate future multi-center cooperation, in compliance with security protocols
2. Data Export
A- Dinamically retrieve data from the database in real-time
B - Automatically anonymize exported data
C – Allow complex filtering of database records
3. Data Transformation and Processing
A- Limit manual time-consuming tasks through scripted data transformation & processing
B- Allow automatic computation of new variables based on primary data
4. Data Analysis
A- Allow scripted analyses on the retrieved live datasets
5. Data Audit
A- Automatically perform reproducible analyses at multiple time-points, to monitor the data-collection process, analyze data distribution, and evaluate preliminary results

However, this scenario has brought forth the need for an effective strategy of data-storage and data-analysis that is both economically sustainable and technically feasible in the context of everyday clinical and research practice in medical departments, with limited resources, time and manpower. Indeed, many departments still rely on local spreadsheet files, which inevitably fall short when trying to manage and analyze larger volumes of relational data [2]. Moreover, data processing before the intended analyses should be automated as much as possible since manually performing this task on large volumes of data would be too time-consuming for clinicians and also prone to errors (e.g., typos, missing values, format inconsistencies and duplicate entries). Finally, statistical analyses should be scripted in order to guarantee reproducible analyses at different timepoints.

Thus, there is a need for a widely-usable and effective data-workflow that would allow clinicians to:

- collaboratively collect large volumes of structured data in a relational database.
- extract subsets of data and carry out the required data manipulation with limited manual intervention.
- perform scripted analyses on data subsets retrieved in real-time.

Therefore, the aim of this study was to develop a widely-usable data-collection and data-analysis workflow that could achieve all the objectives above and could be applied to the research projects of a medical department to perform high-volume relational data analysis on live data.

1.1. Objectives

The primary objective of this study was to develop and implement a data workflow that would allow clinicians to collaboratively collect and store large volumes of structured data into a relational database, and perform scripted analyses on live data subsets. The secondary objective was to improve the cost-effectiveness of the developed strategy by using only open-source and/or cost-free software libraries.

2. Methods

A list of requirements for an effective data-collection and data-analysis strategy was defined based on shared research experience of the authors: the resulting checklist (Table 1) was used to guide the development of the data workflow and to validate its performance.

The following technologies were selected for the project, based on widespread availability, reliability, cost, degree of technical difficulty and expertise of the research team:

- REDCap (v11.0.0) [3,4] for data-collection.
- SQLite (v3.31.1) [5] for relational data persistence.
- Python (v3.8.5) programming language [6], along with the Peewee (v3.7.0) library [7] for database interaction, and the Scikit-learn v0.22 [8] and Matplotlib v3.1.2 [9] libraries for data analysis and visualization, respectively.

The developed data workflow was applied to a research project at our Nuclear Medicine department to store and analyze data on the performance of prostate-specific-membrane-antigen/positron-emission-tomography (PSMA-PET) molecular imaging for the early detection of disease relapse in prostate cancer patients with biochemical recurrence after radical treatment (i.e., PSA elevation after radical prostatectomy or radiotherapy).

Finally, the key steps required to implement the data strategy were documented and the core code of the model was shared through a GitHub repository in order to effectively disseminate results and facilitate other departments in adopting a similar strategy.

3. Results

The newly developed data workflow met the established requirements (defined based on shared research experience of the authors – see Table 1 checklist) and was successfully implemented in a research project at our Nuclear Medicine department to evaluate the performance of prostate-specific-membrane-antigen/positron-emission-tomography (PSMA-PET) in detecting disease localizations in prostate cancer patients with biochemical recurrence after radical treatment.

Specifically, the data strategy allowed to: collaboratively collect and store large volumes of relational data, retrieve live datasets, filter data through complex queries, perform automatic computation of new parameters, and perform scripted analyses on real-time data. The key steps required to implement this newly-developed data workflow are detailed below and summarized in Fig. 1. Moreover, the code of the sample project has been made available through a dedicated GitHub repository [10] (https://github.com/grovera-md/REDCap-based-Data-Workflow).

Step 1: REDCap database and Repeating Instruments

REDCap (v11.0.0) is a web-based software platform designed to allow multiple non-technical users to remotely collaborate and collect data in a reproducible and standardized format [3,4]. REDCap licenses are provided at no cost to non-profit organizations.

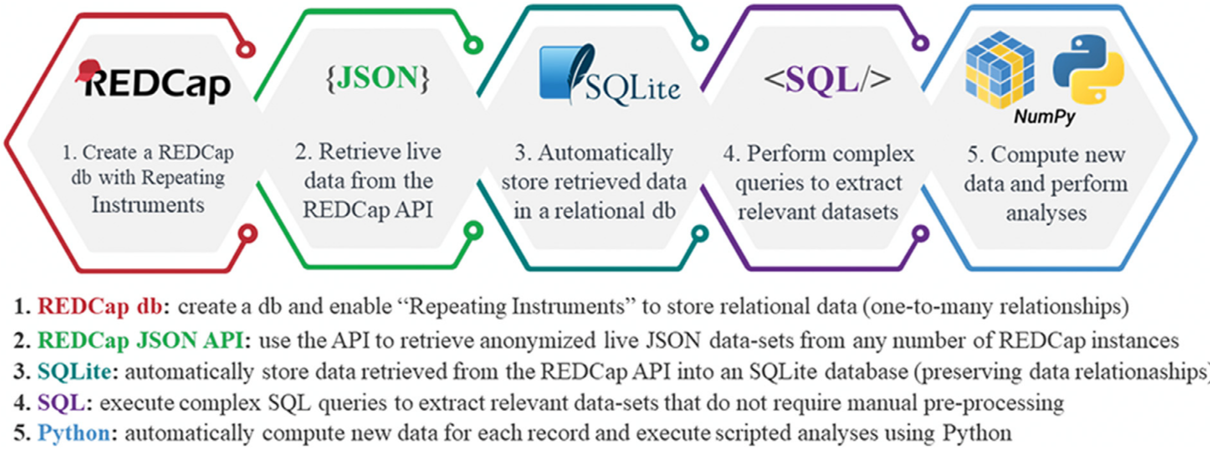

Fig. 1. Key steps of the newly developed data-collection and data-analysis approach, allowing multi-center high-volume relational data analysis on dynamically-retrieved live data.

Since a REDCap instance was already available at our institution, this tool was chosen for data collection. Clinical data such as TNM cancer staging, histopathologic grading, previous treatments, latest PSA values, and PET/CT imaging results were collected. REDCap allowed to store data using coded values and proper validation rules (e.g., minimum / maximum accepted values, date formats), ensuring that only valid information got stored and improving data quality and consistency (checklist item 1A). Specifically, the data fields corresponding to strings, numbers and dates were created in REDCap using the “text boxes” functionality with adequate validation options. Moreover, in order to store relational data, the REDCap “Repeating Instruments” functionality was used: indeed, this feature allowed to store multiple instances of the same event for each record (e.g., all PSA determinations of a prostate cancer patient), without having to define a fixed number of event repetitions in advance (checklist item 1B).

REDCap also provided role-based access to sensitive data in compliance with standard security and privacy protocols, and the REDCap user privileges and de-identification methods allowed to automatically remove fields tagged as identifiers from the exported patients' records (checklist items 1C). Finally, some of the REDCap functionalities could greatly facilitate future multi-center studies: on the one hand, using a common REDCap project xml template can guarantee the reproducibility of the database design among different institutions, while on the other hand the REDCap application programming interface (API) allows to pull data from the REDCap databases of other centers in real-time (checklist items 1D).

Step 2: Real-time data export through the REDCap API and Persistence to a relational database

The REDCap API successfully allowed for real-time retrieval of anonymized datasets from the REDCap database (checklist item 2A-B). The capability to access real-time / live data stored in a centralized repository allowed to automatically retrieve and analyze all newly-inserted and updated records up to the time of the script execution. However, the REDCap API provides only limited query options to filter retrieved records compared to native SQL queries, therefore a Python (v3.8.5) software was developed to automatically persist all live data retrieved from the REDCap API to a relational SQLite (v3.31.1) database, and the Peewee (v3.7.0) object-relational mapper (ORM) was used to bridge data stored in relational tables with Python objects in order to perform more in-depth filtering with SQL-queries at a later stage. In the SQLite database, numbers data fields were mapped to integers/reals while strings and dates were converted to text fields (as per the Peewee library documentation [11]), ensuring a consistent format across all entries; join tables and foreign keys were used to store one-to-many relationships, while unique indexes allowed to prevent duplicate entries.

The SQLite database can be stored locally until needed, so that statistical analyses can be repeated without the need to perform a network request every time. On the other hand, if data need to be refreshed, the local SQLite file can be discarded and a new instance of the live data can be retrieved through the API.

Step 3: Querying the relational database, Computation of new values and Data-analysis

The Pewee (v3.7.0) ORM allowed to perform complex queries to extract relevant datasets from the database (checklist item 2C). As an example, in the sample project code hosted on GitHub [10], the Pewee ORM was used to extract a subset of prostate cancer patients having at least three PSA determinations during the 6 months preceding the PSMA-PET date (which are required to estimate the PSA kinetics).

Then, the Python programming language was used to automatically perform data manipulation (e.g. string sanitization) and computation of new variables for each record, thus greatly reducing the amount of time-consuming manual tasks (checklist item 3A-B). Specifically, in the sample project, the PSA doubling time (which is a measure of PSA kinetics) was automatically computed as natural log of 2 divided by the slope of the relationship between the log of PSA and time [12,13].

Finally, scripted analyses were then performed on the resulting live dataset using the Scikit-learn (v0.22) Python library (checklist item 4A).

On a final note, the streamlined workflow described above could also be leveraged to automatically perform reproducible analyses at different time-points in order to audit the data-collection process and obtain interim results (checklist item 5A).

3.1. Performance evaluation

The performance of the proposed workflow has been evaluated at different data volumes.

Specifically, the execution times of the three key steps of the script (REDCap API data export, SQLite data import, SQLite database query) have been measured considering 1, 5, 10, 20, 35 and 50 $\times$ 10^3 records in the primary table. The execution times of the first two steps have been averaged across 10 iterations, while 100 iterations where considered for the third step. The corresponding results have been reported in detail in Table 2 and Fig. 2, and

Table 2
Workflow performance evaluation: script execution time at different data volumes.

Sample size (main records)	REDCap API data export (seconds)	SQLite data import (seconds)	SQLite db query (seconds)	Total (seconds)
1.000	1,50	0,14	0,01	1,65
5.000	1,77	0,73	0,06	2,56
10.000	4,78	1,45	0,12	6,34
20.000	6,99	2,49	0,23	9,72
35.000	11,37	4,38	0,41	16,15
50.000	14,99	6,31	0,56	21,86

Notes: sample size defined as the number of records in the primary table; REDCap API data export and SQLite data import execution times averaged across 10 iterations; SQLite db query execution time averaged across 100 iterations.

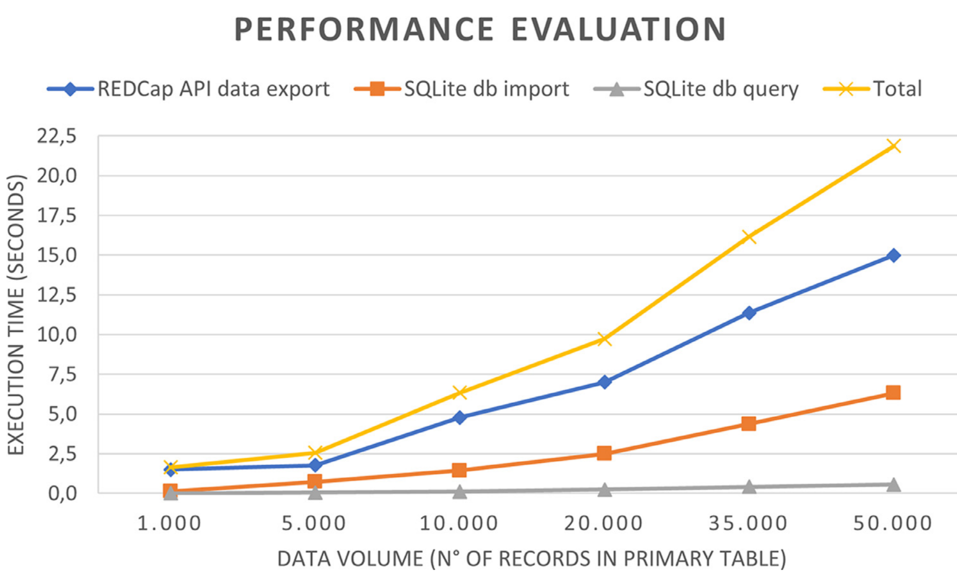

Fig. 2. Workflow performance evaluation: script execution time at different data volumes.

the programming code used for performance evaluation has been included in the GitHub repository in order to ensure reproducibility.

4. Discussion

While the large volume of generated healthcare data holds great potential for clinical research, the knowledge and skills of physicians in the data era need to evolve accordingly in order to effectively leverage such opportunities. Specifically, there is a need to streamline the data collection and data analysis process by developing a workflow that is both tecnhically feasible and economically sustainable. The optimization of the data workflow can greatly facilitate research by limiting the amount of time-consuming tasks, which represent a critical obstacle in the context of limited time and manpower in medical departments, especially in the absence of a data-management unit. Moreover, the diffusion and adoption of best practices will also foster a cultural change in physicians towards a data-science approach, leading to a closer and more effective collaboration with data scientists in the future. However, while electronic data capture systems and effective data management and data analysis strategies are widely adopted at national/international level and in research facilities, they are often lacking or underutilezed in many healthcare institutions/departments or in smaller research studies [14], which sometimes resort to custom solutions for managing multidimensional clinical data in interdisciplinary studies [15]. This issue has also been recently documented by Krahe et al. [16] in an Australian cross-sectional survey, which sought to examine the current research data management (RDM) practices among health and medical researchers: the survey results highlighted the need for further training in RDM best practices and the researchers' desire for support with both theoretical and technical aspects.

While limited literature is currently available on the subject, this study aims to document the development of an effective REDCap-based data workflow, providing other clinicians with a practical solution that could be implemented in other healthcare institutions to facilitate research and boost scientific production. The technical aspects of the developed data strategy have been detailed in the results section of this study and the sample project code has been made available through a GitHub repository [10], in order to effectively disseminate results and allow other healthcare institutions to adopt a similar strategy.

Besides technical features, some key aspects regarding data collection, data extraction/analysis and economic sustainability are discussed below, in order to provide a better undestanding of the advantages offered by the developed data workflow.

4.1. Data collection with relational databases (RDBs)

Nowadays, data collection in many healthcare departments is still carried out through standard spreadsheets [16]. This approach is impractical, memory intensive, high-maintenance and error-prone; moreover, it only accomodes for one-to-one data relationships, thus being insufficient for proper management of clinical data. Conversely, relational databases (RDBs) gracefully manage structured data, since they are based on the relational model of data first introduced by E. F. Codd [17]. RDBs are specifically optimized to handle large volumes of data and, in recent years, they have been successfully adopted in many research projects, ranging from DNA sequencing to cardiovascular diseases [18,19]. Among other advantages, the use of “typed” data fields in RDBs ensures a consistent format across all entries, while unique “keys” prevent duplicate entries. Finally, many high-performance RDBs are open source, thus providing a compelling case for the wide adoption of such potentially cost-effective technologies.

Dispite all the advantages of RDBs for storing structured data, the direct usage of SQL-based RDBs during the data collection phase could pose some challenges due to the degree of technical complexity; thankfully, the introduction of the REDCap “Repeating Instruments” feature has solved this issue by allowing users with low technical expertise to easily collect and store one-to-many relational data through a simple web-based interface. While in the presented workflow data were manually entered in REDCap using the automatically-generated electronic case report forms (eCRF), future further improvements may include directly importing data from electronic health records (EHR). However, this process would require dealing with complex technical issues (e.g., accurate data extraction/text mining from EHR, data validation, etc), as well as normative regulations (e.g., approval for extensive high-level access to patients’ EHR data) and logistic requirements (e.g., funding for third-party support for project development and integration with the local EHR system).

4.2. REDCap data extraction and filtering

In this study, live REDCap data were first exported to an RDB in order to use the SQL language to perform advanced queries (through a Python ORM) and extract relevant datasets in real time.

By default, the REDCap web interface doesn’t allow to export data to an RDB but only to spreadsheets-like files (compatible with main statistical softwares); thus, a custom Python script was developed to connect to the REDCap API, extract data and persist them to an SQLite database.

An application called “REDCap-ETL”, available either as an external module or a standalone software, could represent an alternative method to load data into a relational database. However, the “REDCap-ETL” external module is a fee-based service and doesn’t support SQLite files [20], while the standalone software is free of charge and can export data to SQLite files but requires a server and has some limitations in the use of primary/foreign keys with SQLite databases [21]. Moreover, directly retrieving data through the Python REDCap API allows for more granular control and easier data manipulation before data persistence.

4.3. Strategy cost-effectiveness

The data workflow presented in this study has the potential for increased cost-effectiveness compared to traditional spreadsheet-based strategies, by optimizing the data collection/analysis process, limiting time-consuming tasks, and requiring only open-source or cost-free softwares for its implementation. Altough IT support is needed for REDCap setup and maintenance, REDCap licenses are provided at no cost to any non-profit organization. Such a potentially cost-effective strategy could also particularly benefit economically developing countries, as shown in previous similar efforts [22], empowering their healthcare institutions to perform high-volume relational data analysis.

In conclusion, the REDCap-based data workflow presented in this study represents a practical solution for healthcare institutions to perform high-volume relational data analysis on real-time data at no additional cost (besides REDCap setup and maintenance), by using only open source or cost-free softwares. Finally, thanks to the capability of handling large amounts of clinical data and the possibility to streamline data analysis on such large datasets, the data workflow presented in this study could also provide a suitable infrastructure for future machine learning applications, which require processing of large amounts of training data.

4.4. Limitations

This study is not exempt from limitations. First, no formal comparative analysis was carried out to identify the technologies better suited for the development of the data strategy. Instead, technologies were chosen based on availability at our institution and/or according to the research team expertise. However, the REDCap platform is a widely-used and validated tool for electronic data capturing and the other chosen software libraries are well-known and reliable open source projects.

Second, the workflow presented in this study used an SQLite database for data persistence. While other database technologies, such as MySQL and PostgreSQL, would be better suited to handle large volumes of data in a production environment, SQLite was chosen for this project as a trade-off between performance and ease of implementation: indeed, other database technologies often require to setup a server and this increases the technical difficulty faced by clinical researchers; on the other hand, SQLite can benefit from its single-file storage capability and its documentation reports a theoretical maximum database size of 281 terabytes (with up to 2^{64} table rows) [23]. Furthermore, the Peewee ORM used in this project already natively supports other database technologies, including MySQL and PostgreSQL, therefore switching to other RDB platforms would only require minor code changes.

Finally, while basic knowledge of Python and SQL is needed, our experience shows that clinical researchers can successfully learn to use such tools to optimize the research workflow. On the other hand, while specialized IT support is mandatory for REDCap setup and maintenance, REDCap is already available in many institutions and its licenses are provided at no cost to any non-profit organization.

5. Conclusions

A REDCap-based data-collection and data-analysis workflow, suitable for high-volume relational data-analysis on real-time data, was developed and successfully implemented using open-source software. This strategy could provide medical departments with a sustainable solution to further improve scientific research, streamlining time-consuming tasks and facilitating future multi-center collaborations.

Funding

None

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Guido Rovera: Conceptualization, Methodology, Software, Data curation, Investigation, Validation, Writing – original draft, Writing – review & editing. **Piero Fariselli:** Project administration, Supervision, Methodology, Writing – original draft, Writing – review & editing. **Désirée Deandreis:** Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing.

References

[1] A. Rajkomar, J. Dean, I. Kohane, Machine learning in medicine, N. Engl. J. Med. 380 (2019) 1347–1358, doi:10.1056/NEJMra1814259.
[2] T.J. Dilling, Artificial Intelligence research: the utility and design of a relational database system, Adv. Radiat. Oncol. 5 (2020) 1280–1285, doi:10.1016/j.adro.2020.06.027.
[3] P.A. Harris, R. Taylor, R. Thielke, J. Payne, N. Gonzalez, J.G. Conde, Research electronic data capture (REDCap)–a metadata-driven methodology and workflow process for providing translational research informatics support, J. Biomed. Inform. 42 (2009) 377–381, doi:10.1016/j.jbi.2008.08.010.
[4] P.A. Harris, R. Taylor, B.L. Minor, V. Elliott, M. Fernandez, L. O'Neal, L. McLeod, G. Delacqua, F. Delacqua, J. Kirby, S.N. Duda, REDCap consortium, The REDCap consortium: building an international community of software platform partners, J. Biomed. Inform. 95 (2019) 103208, doi:10.1016/j.jbi.2019.103208.
[5] R.D. Hipp, SQLite, (2020). https://www.sqlite.org/index.html.
[6] G. van Rossum, Python, (2020). https://www.python.org/.
[7] C. Leifer, Peewee. https://github.com/coleifer/peewee.
[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[9] J.D. Hunter, Matplotlib: A 2D graphics environment, Comput. Sci. Eng. 9 (2007) 90–95, doi:10.1109/MCSE.2007.55.
[10] G. Rovera, REDCap-based Data Workflow, Github (2022). https://github.com/grovera-md/REDCap-based-Data-Workflow.
[11] C. Leifer, Models and Fields, Peewee 3.15.1 Documentation. (2022). http://docs.peewee-orm.com/en/latest/peewee/models.html?#field-types-table (accessed July 25, 2022).
[12] C.R. Pound, A.W. Partin, M.A. Eisenberger, D.W. Chan, J.D. Pearson, P.C. Walsh, Natural history of progression after PSA elevation following radical prostatectomy, JAMA 281 (1999) 1591–1597, doi:10.1001/jama.281.17.1591.
[13] L. Benecchi, A.M. Pieri, C. Destro Pastizzaro, M. Potenzoni, Optimal measure of PSA kinetics to identify prostate cancer, Urology 71 (2008) 390–394, doi:10.1016/j.urology.2007.10.021.
[14] L.A. Modly, D.J. Smith, The need for data management standards in public health nursing: a narrative review and case study, Public Health Nurs. (2022), doi:10.1111/phn.13066.
[15] M. Biermann, A simple versatile solution for collecting multidimensional clinical data based on the CakePHP web application framework, Comput. Methods Programs Biomed. 114 (2014) 70–79, doi:10.1016/j.cmpb.2014.01.007.
[16] M.A. Krahe, J. Toohey, M. Wolski, P.A. Scuffham, S. Reilly, Research data management in practice: Results from a cross-sectional survey of health and medical researchers from an academic institution in Australia, Health Inf. Manag. 49 (2020) 108–116, doi:10.1177/1833358319831318.
[17] E.F. Codd, A relational model of data for large shared data banks. 1970, MD Comput. 15 (1998) 162–166.
[18] R. Wilton, S.J. Wheelan, A.S. Szalay, S.L. Salzberg, The terabase search engine: a large-scale relational database of short-read sequences, Bioinformatics 35 (2019) 665–670, doi:10.1093/bioinformatics/bty657.
[19] R. Megna, M. Petretta, B. Alfano, V. Cantoni, R. Green, S. Daniele, W. Acampa, C. Nappi, V. Gaudieri, R. Assante, E. Zampella, E. Mazziotti, T. Mannarino, P. Buongiorno, A. Cuocolo, A new relational database including clinical data and myocardial perfusion imaging findings in coronary artery disease, Curr. Med. Imaging Rev. 15 (2019) 661–671, doi:10.2174/1573405614666180807110829.
[20] Indiana University, Use REDCap-ETL to transform and move data from IU REDCap to another service, University Information Technology Services. (2022). https://kb.iu.edu/d/bhch (accessed April 6, 2022).
[21] IU REDCap, REDCap-ETL, (2022). https://github.com/IUREDCap/redcap-etl/blob/9f67a317742d146391f6cb32e626c9e824b81785/docs/ConfigurationGuide.md.
[22] T. Tuti, M. Bitok, C. Paton, B. Makone, L. Malla, N. Muinga, D. Gathara, M. English, Innovating to enhance clinical data management using non-commercial and open source solutions across a multi-center network supporting inpatient pediatric care and research in Kenya, J. Am. Med. Inform. Assoc. 23 (2016) 184–192, doi:10.1093/jamia/ocv028.
[23] Implementation Limits For SQLite, SQLite. (2022). https://www.sqlite.org/limits.html (accessed May 16, 2021).