BioHackrXiv

Infrastructure for synthetic health data

Núria Queralt-Rosinach[1], Basel Alshaikhdeeb[2], Luca Bolzani[2], Muhammad Shoaib[2], Marcos Casado Barbero[4], Sergi Aguiló-Castillo[3], Davide Cirillo[3], Leyla Jael Castro[5], Ginger Tsueng[6], Matúš Kalaš[7], Magnus Palmblad[8], Danielle Welter[2], Soumyabrata Ghosh[2], Venkata Pardhasaradhi Satagopam[2], and Rahuman S. Malik Sheriff[4]

1 Human Genetics, Leiden University Medical Center, Leiden, Netherlands **2** Luxembourg Center for Systems Biomedicine, University of Luxembourg, Luxembourg **3** Barcelona Supercomputing Center, Barcelona, Spain **4** European Molecular Biology Laboratory - European Bioinformatics Institute (EMBL-EBI), Hinxton CB10 1SD, UK **5** ZB MED Information Centre for Life Sciences, Cologne, Germany **6** Scripps Research Institute, La Jolla, CA 92037, US **7** Department of Informatics, University of Bergen, Norway **8** Center for Proteomics and Metabolomics, Leiden University Medical Center, Leiden, Netherlands

BioHackathon series:
BioHackathon Europe 2022
Paris, France, 2022
Project 15

Submitted: 21 Jul 2023

Published by BioHackrXiv.org

Introduction

Machine learning (ML) methods are becoming ever more prevalent across all domains of life sciences. However, a key component of effective ML is the availability of large datasets that are diverse and representative. In the context of health systems, with significant heterogeneity of clinical phenotypes and diversity of healthcare systems, there exists a necessity to develop and refine unbiased and fair ML models. Synthetic data are increasingly being used to protect the patient's right to privacy and overcome the paucity of annotated open-access medical data. Synthetic data and generative models can address these challenges while advancing the use of ML in healthcare and research.

Following up the efforts currently undertaken in the ELIXIR Health Data and the Machine Learning Focus Groups around the synthetic health-data landscape, this project focuses on the health data providers' need for a ready-to-use synthetic data platform assessed by health data experts, researchers, and ML specialists. Aligned with ELIXIR Health Data Focus Group's objectives, we aim at building an infrastructure for synthetic health data offering a containerised synthetic data generator based on the open-source libraries Synthetic Data Vault (SDV) and ydata-synthetic with state-of-the-art ML methods. This framework will enable users to generate synthetic data that have the same structure and statistical properties as the original dataset from a variety of sources (clinical, variational, or omics). Despite the capacity to generate their own datasets, a set of exemplary datasets will be publicly available in appropriate repositories and will include rich metadata descriptions according to the DOME recommendations and GA4GH standards. OpenEBench will host a community of practice for comparing different approaches for synthetic data generation. Here, we present our proof of concept for the generation of synthetic health data and our proposed FAIR implementation of the generated synthetic datasets. The work was developed during and after the one-week-long BioHackathon Europe, by together 20 participants (10 new to the project), from different countries (NL, ES, LU, UK, GR, FL, DE, . . .).

Infrastructure for synthetic health data

For test and stress development of new ML methods/tools, we need suitable data to properly demonstrate the method/tools application. However, ML developers without health data access are not able to see how the tool performs for its intended application. One way to enable this is to generate *synthetic data*, where new "fake" data is created from real data using a specifically designed generation model. Importantly, the generation model must maintain the

original features and structure to be as realistic as possible to the original data. In particular, synthetic health data goals are to ensure:

1. **Quality**: it must be a sufficient and necessary representation of the real data.
2. **Applicability**: the synthetic data must fit for its intended application.
3. **Privacy**: synthetic data must be anonymous or pseudo-anonymous. In other words, linkage between the synthetic data and a natural person must be protected, thus preventing that a generated synthetic patient is actually real.

Our main goal was to provide to the health-research community and life scientists with a simple and reusable infrastructure for generating privacy-preserving and effective synthetic health data. This is to be used in artificial intelligence technologies and ML methods/tools to speed up research on health data. During the hackathon we designed and implemented a prototype as a proof of concept of our infrastructure idea. There are already some existing methods to generate synthetic data. First, we investigated some of the most popular of these methods. Second, we investigated the current quality assessment methods (metrics and tools) to check the quality of the generated synthetic datasets. Next, we present the infrastructure prototype composed of: 1. the workflows we developed to generate synthetic health data; 2. the quality metrics we implemented to assess the generated synthetic data; 3. the user interface to interact with the whole process: a web user interface (UI) and a Docker container.

1. Synthetic data generation workflows

To develop an infrastructure for generating synthetic health data that is easy to use, we first explored how to generate synthetic health data from real data. There are many existing methods to generate synthetic data. The first challenge was that these methods are usually tailored for a specific type and format of the input dataset.

We first defined the input data type as *tabular data* for its simplicity. Then, we chose a starting dataset that was already published and open-access. We selected the Breast Cancer Wisconsin (Diagnostic) Data set published on the Kaggle website, rated with a usability of 8.53. This dataset contains 32 columns, including: 2 attributes (identifier *ID* (integer) and *Diagnosis* (M=malignant, B=benign)); 10 real-valued features computed for each cell nucleus from a digitised image of a breast mass; and three statistical parameters for each feature (*Mean*, *standard error* and *worst*). They describe characteristics of the cell nuclei present in the image (Bennett & Mangasarian, 1992). This dataset is similar to those for clinical research making it suitable to produce synthetic data for health.

We experimented with three synthetic data generators: SDV, DataSynthesizer, and SyntheaTM. Then, we provided some generation workflows based on these existing methods. The workflows are available as Jupyter Notebooks on Github. Finally, we did a model comparison to aid the user to decide the model/parameters to generate a proper synthetic dataset.

Synthetic Data Vault (SDV)

(Patki et al., 2016) presented a framework known as Synthetic Data Vault (SDV) that was intended to generate synthetic relational data. Such a framework combines both statistical approaches and deep learning methods to accommodate the synthetization. The authors of this framework were taking into account the generation of synthetic data that is statistically similar to the original one even in terms of missing values, categorical and datetime distribution. Meanwhile, the authors have focused on increasing privacy by enabling the user to adjust the model's parameters in order to add different noise.

Within the statistical part, the SDV framework utilised different versions of multivariate Copula Gaussian distribution (*e.g.* uniform, truncated gaussian, gamma, beta, *etc.*). In this regard, two main aspects have been considered including distribution and covariance where the first refers to the distributional probability of the values within each attribute while the latter refers to the impact of each value within an attribute in accordance with the values within other attributes.

In addition, SDV framework has utilised an approach called Conditional Parameter Aggregation (CPA) to mimic the relationships between tables during the generation of a relational database in which the user could provide metadata about the tables. Furthermore, SDV framework provides two paradigms of synthetic generation: through mode-based which aims at generating synthetic data based on an existing model, and knowledge-based which aims at expanding existing data by generating more examples.

On the other hand, SDV framework also utilises a deep learning architecture known as Generative Adversarial Network (GAN). This architecture is based on Deep Neural Network which consists of two main components; generator and discriminator where the first refers to the model that is trained to generate new data while the latter refers to the model that is trained to classify the data into real or synthetic (Wang et al., 2017).

DataSynthesizer

DataSynthesizer (*SSDBM '17*, 2017) is a privacy-preserving synthetic data generator. This tool takes a private sensitive dataset as input and generates synthetic data that simulates that given dataset while ensuring strong privacy guarantees. It aims to facilitate collaboration between domain-expert data owners and external data scientists. Importantly, it applies *differential privacy* techniques to achieve strong privacy guarantee. Differential privacy is a family of techniques that guarantee that the output of an algorithm is statistically indistinguishable on a pair of neighbouring databases, *i.e.*, a pair of databases that differ by only one tuple. In particular, it uses privacy-preserving learning of the structure and conditional probabilities of an existing Bayesian network. One of the main features of this tool is its usability since the data owner does not have to specify any parameters to start generating and sharing data safely and effectively.

This is an end-to-end system that is implemented in Python 3 and it assumes that the private sensitive dataset is presented in CSV format. The system implements three intuitive modes of operation that allow to generate synthetic data at three different levels of statistical fidelity: *random mode* for cases of extremely sensitive data which simply generates type-consistent random values for each attribute, *independent attribute mode* for cases in where the correlated attribute mode is too computationally expensive or when there is insufficient data to derive a reasonable model which a histogram is derived for each attribute, noise is added to the histogram to achieve differential privacy and then samples are drawn for each attribute, and *correlated attribute mode* for the rest of cases which learns a differentially private Bayesian network capturing the correlation structure between attributes, then draw samples from this model to construct the result dataset. At the level of implementation it consists of three high-level modules – DataDescriber, DataGenerator, and ModelInspector. The first DataDescriber, investigates the data types, correlations and distributions of the attributes in the sensitive dataset, and produces a data summary adding noise to the distributions to preserve privacy. DataGenerator samples from the summary computed by DataDescriber and outputs synthetic data. ModelInspector shows an intuitive description of the data summary that was computed by DataDescriber, allowing the data owner to evaluate the accuracy of the summarization process and adjust any parameters, if desired. It provides several built-in functions to inspect the similarity between the sensitive dataset and the synthetic dataset. This way, the data owner can quickly test whether the tuples in the synthetic data are detectable by inspecting and comparing at-a-glance the statistical properties of both datasets.

We selected this library because it was the synthetic data generator of choice of the Common Infrastructure for National Cohorts in Europe, Canada, and Africa (CINECA) project and we implemented three different workflows as Jupyter Notebooks to showcase its use with different level of fidelity: random mode, independent attribute mode, and correlated attribute mode. DataSynthesizer is open source, and is available for download at https://github.com/DataResponsibly/DataSynthesizer.

SyntheaTM

SyntheaTM generates synthetic data from the medical history of patients. It aims to create high-quality, realistic data related to patients and associated health records without privacy and security constraints. Thus, with this approach, we can generate data for creating OMOP synthetic datasets.

One of the greatest qualities of SyntheaTM is having more than 90 different modules, each one containing models for different diseases or medical observations. However, most of these modules have dependencies between them, and it is not recommended to restrict the search for a subset of them.

For creating a SyntheaTM dataset we have used the guide section "Population of OMOP CDM tables with synthetic patient data" from the BioHackathon 2021 project called OMOP to Phenopackets for COVID-19 analytics. The basic command line to generate data, in SyntheaTM v2.7, is the following:

```
java -jar synthea-with-dependencies.jar -p 1000 \
-c /pathtosynthea/src/main/resources/synthea.properties
```

Where `-p` is the population size of the data generated and `-c` is the configuration file that must be edited to export data in CSV format by uncommenting the following line `exporter.csv.export = true` . Other options are shown with the `-h` option:

```
java -jar synthea-with-dependencies.jar -h
Usage: run_synthea [options] [state [city]]
Options: [-s seed] [-cs clinicianSeed] [-p populationSize]
         [-r referenceDate as YYYYMMDD]
         [-g gender] [-a minAge-maxAge]
         [-o overflowPopulation]
         [-m moduleFileWildcardList]
         [-c localConfigFilePath]
         [-d localModulesDirPath]
         [-i initialPopulationSnapshotPath]
         [-u updatedPopulationSnapshotPath]
         [-t updateTimePeriodInDays]
         [-f fixedRecordPath]
         [--config* value]
          * any setting from src/main/resources/synthea.properties
Examples:
run_synthea Massachusetts
run_synthea Alaska Juneau
run_synthea -s 12345
run_synthea -p 1000)
run_synthea -s 987 Washington Seattle
run_synthea -s 21 -p 100 Utah "Salt Lake City"
run_synthea -g M -a 60-65
run_synthea -p 10 --exporter.fhir.export true
run_synthea -m moduleFilename:anotherModule:module*
run_synthea --exporter.baseDirectory "./output_tx/" Texas
```

To generate different types of data with modules, one must use the `-m` option with the name of your modules. Check the page with an example here.

After producing the CSV files, one needs to create the OMOP database on their own, or continue following the guide from OMOP to Phenopackets for COVID-19 analytics. The approach is not embedded in the main infrastructure as the different output options need to be discussed for the FrontEnd part.

2. Quality assessment

Once the synthetic health data is generated, one should evaluate the quality (distribution), applicability (model effectiveness) and privacy (re-identification risk) of the synthetic data. This is a hot topic of research and the community has not reached yet a consensus on how to do this assessment. The second challenge was to evaluate the quality of the generated synthetic dataset to know whether the model and the data are actually useful. It is important to set up a minimum way to evaluate the data generated. Here, we explored and discussed some metrics and tools such as SDMetrics library with the ultimate goal to propose them to the OpenEBench community.

SDV parameter comparison

As mentioned earlier, SDV includes different models and a wide range of parameters during the generation of synthetic data. Therefore, an investigation of all these models and parameters would be useful for the user to identify the most appropriate model and settings. These parameters are either referring to the desired statistical distribution through different Copula Gaussian or even using the deep learning of GAN which is called CTGAN. Using the Breast Cancer dataset, a comparison between these parameters has been conducted, in which the dataset was synthesised multiple times using one of these parameters. To evaluate this comparison, a metric score within SDV has been used which compare the synthetic data with the original one. Figure 1 shows such a comparison.

Figure 1. SDV model comparison

As shown in Figure 1, the most accurate generation for the Breast Cancer dataset was depicted by GaussianCopula through Kernel Density Estimation (KDE) which has the highest evaluation score of 0.96. Note that for this reason, this distribution significantly suits the Breast Cancer dataset.

SDV utility classification assessment

As an additional aspect of evaluation, investigating the classification performance of the synthetic data alongside the original one would provide a significant insight on how the synthetic data could be used for further downstream analysis. To this end, the synthetic data generated by the best SDV model parameter of GaussianCopula-KDE has been used. In order to provide a consistent evaluation on such a classification task, a classifier of Random Forest (RF) has been used with 10 cross-fold validation for both datasets (*i.e.*, real and synthetic). In addition, different parameters of RF have been also considered within the classification including the criterion (*i.e.*, gini or entropy) and max-features (*i.e.*, auto, sqrt, log2). Both datasets were divided into training and testing and classified using each RF parameter through the use of Grid Search, in which the average F1-score accuracy was considered for each experiment. Figure 2 shows the results of this comparison.

Figure 2. Utility classification using RF

As depicted in Figure 2, for both real and synthetic dataset, there were no significant differences between the training and testing which demonstrates the absence of either underfitting or overfitting. On the other hand, the results of accuracy for the synthetic were lower but still similar to the real ones which proves the efficacy of the generated synthetic data.

3. User interface

Finally, we designed and implemented a user interface to interact with the infrastructure.

Web UI

Firstly, we designed a Web UI to: 1. Help the user to decide the generation model that best suits its purposes; 2. Access the selected generation workflow and set the generation settings; 3.

Upload the generated synthetic data in the Web UI to assess its quality by means of predefined metrics and visualisation plots. Even though our goal in the beginning was also to provide computing capacity to run the generator workflow from the Web UI, we decided to run the generator tool on the client side due to the high computational cost of these types of jobs.

Figure 3. Infrastructure prototype design we envisioned during the hackathon: photographs of the whiteboard taken during brainstorming.

Secondly, we implemented a prototype of the Website using the Streamlit Python library. Code is available on Github. In Figure 3 there is a sketch of our prototype design, in Figure 4 a screenshot of the Web UI to select the settings, and in Figure 5 a screenshot of the quality assessment analysis component.

Figure 4. Settings component of the Web UI to generate synthetic data.

Figure 5. Quality assessment component of the Web UI to score and visually check the quality of the synthetic data.

Docker container

For executability and portability, we deployed the application in a Docker container. The generative Jupyter workflows and the application should be launched by Docker. We provide the steps to run the Docker image on the README file of the project GitHub repository. The actual generation of the synthetic data should be run by the user through the container by executing the selected workflow. The user should ensure before execution, that the container is launched from a system with the sufficient computing capacity.

FAIR synthetic datasets

In parallel with providing infrastructure for the generation of synthetic health data, experts on health data FAIRness and modelling discussed a FAIR implementation of the generated synthetic data. Next, we present our proposal on how to make the synthetic health datasets comply with the FAIR principles, with the main focus being data reusability. Firstly, we defined a minimal machine-readable metadata model for synthetic health data. Secondly, we identified an existing data repository where to deposit the synthetic data once generated. We decided to use the BioStudies repository from EMBL-EBI, to store and implement our proposed model. In the following sections, we describe the metadata model, the used criteria for creation and submission, and the criteria used to select an appropriate repository for deposition.

1. Metadata model

Having in mind the provision of synthetic datasets that were reusable for the community and readable by machines, *i.e.*, FAIR data, we designed a metadata model to use as guidance for end-users of our framework. This metadata model is publicly available in the GitHub repository. It is meant primarily to improve findability of synthetic health datasets in public repositories. It is composed of 24 descriptors that, together, provide context for 4 different semantic groups: the dataset, generation tool, attribution and provenance. Not all of these descriptors are required for a functional metadata model, and thus we ranked them following the MoSCoW criteria: 11 Must-, 8 Should- and 7 Could-haves. For the sake of consistency in the metadata of these fields, we provided the recommended bio-ontology mappings (preferably OBO ontologies) to use when populating the metadata model. Thanks to EDAM maintainers also attending BioHackathon Europe, we were able to align with EDAM (Black et al. (2022), Kalaš et al. (2020)) and map most terms to this ripe ontology.

The 24 descriptors were defined after reviewing several existing repositories for ML datasets such as Kaggle or huggingface, and identified a minimal number and linked to a MoSCoW rank according to our experience from other projects such as the curation effort within the ELIXIR ML/synthetic data group. Equity and fairness are very important concerns when developing

health tools for research and decision making for the clinical setting. To enable this we propose to record the *biological sex* property. The model is available as TSV and JSON distributions on our project GitHub repository.

Bioschemas

Bioschemas (Gray et al., 2017) is a community effort to facilitate the structured markup of web pages in Life Sciences. Bioschemas profiles are community-standardised recommendations on the application of *Schema.org* types defining a subset of properties and constraints relevant to the life sciences. To further improve FAIRness of the synthetic datasets, we mapped properties in our metadata model to the Bioschemas Dataset profile and included examples of how each property could be used to capture the metadata in JSON Schema, following the Bioschemas Dataset profile.

Bioschemas provides a profile, *i.e.*, structured semantic specification together with recommendations of usage and examples, to model datasets that can be used to describe synthetic data with rich metadata, to ease its reuse by third parties. As the Bioschemas *Dataset* profile covers cases beyond ML and synthetic data, some customization is needed for its effective use, without deviating much from the general case to preserve compatibility with other datasets. One of the desired characteristics when describing synthetic data, and more in general data that could be used in ML training processes, is providing information useful for training purposes. For instance, the sort of content is important to assess whether or not the data is appropriate for a task, *e.g.*, protein or phenotype data, cancer patients stage II, *etc.*. The distribution and characteristics of the data points are also an important aspect for classification tasks, *i.e.*, a training algorithm needs to take into account data skews and possible bias. These two were the synthetic data description cases, on which we focused on during the BioHackathon Europe 2022.

Bioschemas does not provide yet a specific way to express what the content (or topic) of a dataset is about. There are two Dataset properties that could be used to this end: keywords and about. In both cases, it is possible to link to controlled vocabularies via a DefinedTerm, making it easier for machines to "understand" the topic's context. Our recommendation is using topics defined in the EDAM ontology, *i.e.*, those under the section Topic; for instance *Drug discovery* or *Data management*. Whether favouring *keywords* or *about* to describe a dataset topic is a discussion that will further develop in the Bioschemas community.

Regarding the description of data points and other characteristics of the dataset - *e.g.*, age, sample size, sample size, or the number of attributes - could be achieved via variableMeasured with a value expressed using a PropertyValue. Table 1 shows some examples with the corresponding markup in JSON-LD. For other elements necessary to the description of synthetic datasets, we refer the reader to the comprehensive spreadsheet mapping.

Characteristic	JSON-LD markup
Age	`variableMeasured : [{"@type": "PropertyValue", "name": "Age", "value": 32, "unitText": "year"}, "variableMeasured" : {"@type": "PropertyValue", "name": "Age", "value": "adult"}]"`
Sample size	`variableMeasured : [{"@type": "PropertyValue", "name": "Sample size", "value": 10}]"`
Number of attributes	`variableMeasured : {"@type": "PropertyValue", "name": "Number of Attributes", "value": 32}"`

In the future, we aim at generating a Bioschemas profile based on our model using the Data Discovery Engine's Schema Playground (Cano et al., 2023) editor. The profile for synthetic datasets can be a subclass of *Dataset*.

EDAM ontology

As mentioned above, during the BioHackathon we decided to use EDAM concepts to describe the property values of the synthetic datasets metadata to increase findability in data repositories. EDAM is a domain ontology of data analysis and data management in bio- and other sciences, and science-based applications. It is one of the ontologies of reference to annotate provenance metadata of processed data, especially in life sciences (Black et al., 2022) and imaging (Kalaš et al., 2020). The mapping of the metadata model to EDAM was finished during a follow-up hybrid mini-hackathon held a week after the BioHackathon, and a virtual sprint a few months later. Importantly, we added generative ML concepts, and did general refinements in EDAM Bioimaging, which provides a concise but rigorous subsection with ML concepts (to be merged into the mainline EDAM in the near future). We also detected potential new concepts for the synthetic data subdomain to be added to EDAM. Finally, we agreed to map the model descriptors *per se* to EDAM concepts in the future.

2. Repository for deposition

Our goal was to propose to the community a repository for the deposition of the synthetic (health) datasets. Our approach was to review existing data repositories with a machine-learning focus. We wanted to check first their suitability according to our criteria to avoid the development of a new repository. Our criteria was: 1. to use a widely used and mature data repository; 2. for ML datasets or ML modelling; 3. enables to deposit data by authors/users. BioStudies is a database that holds descriptions of biological studies, links to data from these studies in other databases, it can accept a wide range of types of studies described via a simple format, and it also enables manuscript authors to submit supplementary information and link to it from the publication. Furthermore, it is a Recommended ELIXIR Deposition Database, containing almost 10,000,000 files (but not much content on synthetic data from the queries "synthetic" or "synthetic data"), has a flexible submission metadata model (description here) that already aligns with Bioschemas. Datasets submitted to BioStudies are indexed and searchable via omicsDI and EBI Search. We agreed in the suitability of this data repository to extend their scope to synthetic data. After the BioHackathon and once we had the final first version of the metadata model agreed, we first made a version for BioStudies following their submission metadata template, which can be found on GitHub. Then, we contacted them and sended the synthetic metadata template proposal to them to request to use the repository for synthetic data deposition. BioStudies managers, after evaluating our request, agreed to set up a "synthetic data collection" with useful search facets, as in *e.g.* EU-ToxRisk. This is ongoing, but BioStudies will be ready soon to be used as the repository of reference for the deposition of synthetic data.

Discussion

Our overarching goal of developing an infrastructure prototype for the generation of synthetic health data was achieved. Furthermore, we proposed a FAIR implementation recipe by defining a minimal metadata model and suggesting an existing repository for the description, annotation and deposition of the generated synthetic datasets.

Our infrastructure prototype is composed of a set of generation workflows using different tools and methods, and a user interface (web UI + Docker container) to run a generation workflow and assess the quality of the generated synthetic dataset. The web application contains 3 components: 1. *Input data* where the user can upload the real data to synthesise; 2. *Generation model* where the user can select the generative algorithm model and set up

the parameters; 3. *Output data* where the user can upload the generated synthetic data for quality assessment through an intuitive set of metrics and visualisations. This web application prototype is a proof of concept of the simple synthetic health data generation infrastructure envisioned and implemented in collaboration between the health data community and the ML community. The fact that the workflows are separated from the web UI, makes our design modular and easily extendable. With regards to quality assessment, we just explored some metrics and tools, but during the hackathon we already detected some recent existing benchmarking evaluations and visualisations preprints (Kotelnikov et al., 2022). The quality, performance and privacy assessment will be followed up jointly with the OpenEBench effort for community consensus. In the future, we plan to implement this infrastructure for the Health community. An important point would be adding a guidance section for the end-user to aid the decision on what generative model and parameters to use for its downstream application. Another point is to add a data submission form or link to BioStudies to facilitate the FAIRness description and registration of the new synthetic data.

We proposed a minimal FAIR synthetic health data implementation composed of a metadata model and a repository for their deposition. We published version 1 of the metadata model, which is publicly available for open discussion and review on GitHub for the community: metadata model v1 for synthetic health datasets. To enhance synthetic health data findability, we mapped the model to both Bioschemas and the EDAM ontology, which is the ontology used to tag resources *e.g.* in Bio.tools (Ison et al., 2021). In the future, we plan to collaborate with RO-Crate to provide data+tool as FAIR research objects, describe best practices using RDMKit, and set up FAIR recipes for synthetic data generation in FAIRCookbook.

The BioHackathon Europe venue set a perfect environment for engagement: our project had 20 participants, where 6 were new to BioHackathon (plus 4 remote), and from 7 countries (NL, ES, LU, UK, GR, FL, and DE). It gave us the opportunity to collaborate with 5 other projects (nr. 4, 5, 17, 18, and 25 (Lamothe et al., 2023)). Nevertheless, we suffered some challenges as well, such as keeping a steady progress pace between in-person and remote attendees. Overall, it was another great experience with a satisfactory project outcome, and with further follow-up development.

Future work includes:

1. Addition of documentation;
2. Inclusion of more data types and workflows such as using ML with ontologies;
3. Align to the DOME recommendations;
4. Request a *synthetic data* metadata profile to the BioStudies repository and upload some datasets;
5. Description of the synthetic datasets and its workflow generation as RO-Crate/CWL digital objects;
6. Implement as a Python package.

As long-term outcomes, we are planning to submit a manuscript on the synthetic health data infrastructure developed following ELIXIR requirements. The development of the infrastructure *per se* is a long-term outcome, where we envision adding other components such as implementing evaluation metrics to assess the quality of the generated synthetic data and a direct deposition of the synthetic datasets to recommended repositories.

Contributions

All authors actively participated during and after the hackathon to produce the outcomes we detail in this paper; they provided content and all reviewed or revised the paper. Specifically, NQR contributed to providing workflows, the metadata model, wrote about these in dedicated sections and drafted the final paper. BA, DC and SAC contributed to providing workflows and wrote about this in the paper. LB contributed on coding the Web UI and packaged it in a

Docker container, and managed the project GitHub repository and its documentation. MS contributed to the evaluation metrics of the synthetic data and wrote about this in the paper. MCB, SAC, DW, LJC, GT and RS contributed to the metadata model and wrote about this in the paper. MK, NQR, MP, and DC contributed to the metadata model mapping to EDAM, and extended EDAM Bioimaging with generative ML concepts.

Acknowledgements

We thank the organisers of the BioHackathon Europe 2022 for travel support for some of the authors. This work was funded/supported by ELIXIR, the research infrastructure for life-science data. Special thanks to Fotis Psomopoulos, Salvador Capella Gutierrez for having the original project idea and contributing during the hackathon. Also, Nick Juty and Francis Chemorion for your very valuable contributions. Finally, to all participants of the BioHackathon Europe 2022 that made possible this project engaging with fruitful discussions or giving us support.

References

Bennett, K. P., & Mangasarian, O. L. (1992). Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, *1*, 23–34.

Black, M., Lamothe, L., Eldakroury, H., Kierkegaard, M., Priya, A., Machinda, A., Khanduja, U. S., Patoliya, D., Rathi, R., Nico, T. P. C., Umutesi, G., Blankenburg, C., Op, A., Chieke, P., Babatunde, O., Laurie, S., Neumann, S., Schwämmle, V., Kuzmin, I., … Kalaš, M. (2022). EDAM: the bioscientific data analysis ontology (update 2021) [version 1; not peer reviewed]. *F1000Research*, *11*(ISCB Comm J), 1. https://doi.org/10.7490/f1000research.1116432.1

Cano, M. A., Tsueng, G., Zhou, X., Xin, J., Hughes, L. D., Mullen, J. L., Su, A. I., & Wu, C. (2023). Schema playground: A tool for authoring, extending, and using metadata schemas to improve FAIRness of biomedical data. *BMC Bioinformatics*, *24*, 159. https://doi.org/https://doi.org/10.1186/s12859-023-05258-4

Gray, A. J. G., Goble, C. A., & Jimenez, R. (2017). *Bioschemas: From potato salad to protein annotation*.

Ison, J., Ienasescu, H., Rydza, E., Chmura, P., Rapacki, K., Gaignard, A., Schwämmle, V., Helden, J. van, Kalaš, M., & Ménager, H. (2021). biotoolsSchema: a formalized schema for bioinformatics software description. *GigaScience*, *10*(1). https://doi.org/10.1093/gigascience/giaa157

Kalaš, M., Plantard, L., Lindblad, J., Jones, M., Sladoje, N., Kirschmann, M. A., Chessel, A., Scholz, L., Rössler, F., Sáenz, L. N., Mariscal, E. G. de, Bogovic, J., Dufour, A., Heiligenstein, X., Waithe, D., Domart, M.-C., Karreman, M., Plas, R. V. de, Haase, R., … We are welcoming new contributors! (2020). EDAM-bioimaging: the ontology of bioimage informatics operations, topics, data, and formats (update 2020) [version 1; not peer reviewed]. *F1000Research*, *9*(ELIXIR), 162. https://doi.org/10.7490/f1000research.1117826.1

Kotelnikov, A., Baranchuk, D., Rubachev, I., & Babenko, A. (2022). TabDDPM: Modelling Tabular Data with Diffusion Models. *arXiv*. https://doi.org/10.48550/arXiv.2209.15421

Lamothe, L., Jensen, J. R. B., Ienasescu, H., Gustafsson, O. J. R., Gaignard, A., Repchevsky, D., Svobodová, R., Raček, T., Antol, M., Palmblad, M., Kalaš, M., & Ménager, H. (2023). An evaluation of EDAM coverage in the Tools Ecosystem and prototype integration of Galaxy and WorkflowHub systems. *BioHackrXiv*. https://doi.org/10.37044/osf.io/79kje

Patki, N., Wedge, R., & Veeramachaneni, K. (2016). The synthetic data vault. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 399–410. https://doi.org/10.1109/DSAA.2016.49

SSDBM '17: Proceedings of the 29th international conference on scientific and statistical database management. (2017). Association for Computing Machinery. ISBN: 9781450352826

Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F.-Y. (2017). Generative adversarial networks: Introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, *4*(4), 588–598. https://doi.org/10.1109/JAS.2017.7510583