Article published online: 2021-08-25

Linking a Consortium-Wide Data Quality Assessment Tool with the MIRACUM Metadata Repository

Lorenz A. Kapsner[1,2] Jonathan M. Mang[1] Sebastian Mate[1] Susanne A. Seuchter[1] Abishaa Vengadeswaran[3] Franziska Bathelt[4] Noemi Deppenwiese[1] Dennis Kadioglu[3,5] Detlef Kraska[1] Hans-Ulrich Prokosch[1,6]

[1] Medical Center for Information and Communication Technology, Universitätsklinikum Erlangen, Erlangen, Germany
[2] Department of Radiology, Universitätsklinikum Erlangen, Friedrich-Alexander-University Erlangen-Nürnberg, Erlangen, Germany
[3] Medical Informatics Group (MIG), Goethe University Frankfurt, University Hospital Frankfurt, Frankfurt am Main, Germany
[4] Institute for Medical Informatics and Biometry, Carl Gustav Carus Faculty of Medicine, Technical University Dresden, Dresden, Germany
[5] Data Integration Center, University Hospital Frankfurt, Frankfurt am Main, Germany
[6] Department of Medical Informatics, Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Erlangen, Germany

Address for correspondence Dr. med. Lorenz A. Kapsner, Medical Center for Information and Communication Technology, Universitätsklinikum Erlangen, Krankenhausstr. 12, 91054 Erlangen, Germany (e-mail: lorenz.kapsner@uk-erlangen.de).

Appl Clin Inform 2021;12:826–835.

Abstract

Background Many research initiatives aim at using data from electronic health records (EHRs) in observational studies. Participating sites of the German Medical Informatics Initiative (MII) established data integration centers to integrate EHR data within research data repositories to support local and federated analyses. To address concerns regarding possible data quality (DQ) issues of hospital routine data compared with data specifically collected for scientific purposes, we have previously presented a data quality assessment (DQA) tool providing a standardized approach to assess DQ of the research data repositories at the MIRACUM consortium's partner sites.

Objectives Major limitations of the former approach included manual interpretation of the results and hard coding of analyses, making their expansion to new data elements and databases time-consuming and error prone. We here present an enhanced version of the DQA tool by linking it to common data element definitions stored in a metadata repository (MDR), adopting the harmonized DQA framework from Kahn et al and its application within the MIRACUM consortium.

Methods Data quality checks were consequently aligned to a harmonized DQA terminology. Database-specific information were systematically identified and represented in an MDR. Furthermore, a structured representation of logical relations between data elements was developed to model plausibility-statements in the MDR.

Keywords
- data quality
- electronic health records
- metadata
- secondary use

received
April 19, 2021
accepted after revision
June 27, 2021

DOI https://doi.org/10.1055/s-0041-1733847.
ISSN 1869-0327.

Georg Thieme Verlag KG, Rüdigerstraße 14, 70469 Stuttgart, Germany

Results The MIRACUM DQA tool was linked to data element definitions stored in a consortium-wide MDR. Additional databases used within MIRACUM were linked to the DQ checks by extending the respective data elements in the MDR with the required information. The evaluation of DQ checks was automated. An adaptable software implementation is provided with the R package *DQAstats*.

Conclusion The enhancements of the DQA tool facilitate the future integration of new data elements and make the tool scalable to other databases and data models. It has been provided to all ten MIRACUM partners and was successfully deployed and integrated into their data integration center infrastructure.

Background and Significance

Secondary use of data from electronic health records (EHRs) in observational studies is a common goal of many research initiatives.[1–5] The German Medical Informatics Initiative (MII),[6] comprised of four national consortia, established data integration centers (DIC) at each site, in which EHR data are integrated within research data repositories to allow for site-specific and federated analyses. However, researchers have illustrated that data collected in the context of patient care might be of poorer data quality (DQ) than data collected in scientific studies.[7–10] It would therefore be desirable to accompany the constantly growing content of these research data repositories with data quality assessment (DQA) methods to ensure high DQ and enable the consideration of the underlying DQ when interpreting the resulting research findings. In the setting of a multisite research network such methods can be centrally developed, maintained, and adapted to the research data repositories. To be usable by all network collaborators, they should be straightforward to implement and analyze DQ in accordance with the data protection regulations at each partner site.

Source databases from different network partners are often based on heterogeneous data structures and data definitions. Consequently, for multicenter research studies, one usually needs to agree on a set of harmonized and consented data item definitions.[12] Such definitions can be managed and maintained within a metadata repository (MDR). A common model for MDRs is the ISO/IEC 11179–3 standard.[11] MIRACUM,[12] one of the four MII consortia, uses the *Samply.MDR*[13] as its central metadata repository (M-MDR),[14] where all agreed-upon dataset specifications are published and communicated.

As part of MIRACUM's DIC toolset (the MIRACOLIX ecosystem),[12] we continuously improve the previously developed MIRACUM DQA tool.[15] During MIRACUM's conceptual phase, DQA was performed via random sampling checks for the first proof-of-concept analyses, because a systematic consortium-wide approach was not yet established. The DQA tool has been developed during the first project year in 2018 to provide a standardized approach of assessing DQ at each consortium's partner site by analyzing selected data elements of the MII-consented core dataset modules available within the MIRACUM data repositories at this time.[14–16]

Initially, the tool was solely based on an R Markdown script.[17,18] Although the resulting PDF report was comprehensive, a detailed manual interpretation of the results was required to uncover irregularities and DQ issues. Furthermore, a major limitation was that all analyses were hard coded in the R Markdown script. As a result, the tool was difficult to maintain, the integration of new data elements to be analyzed was time consuming and error prone, and, most importantly, the tool was not generalizable to other databases or data models.

Objectives

The objective of this paper is to link the DQA tool with common data element definitions stored in an MDR to overcome the abovementioned limitations. Furthermore, we aimed at adopting Kahn et al's harmonized DQA framework[7] in our tool and to integrate DQ checks in the MDR. We further describe the application of the DQA tool within the MIRACUM consortium.

Methods

Connecting a Metadata Repository with the DQA Tool

To develop a generic and configurable tool, the use of an MDR to store and manage data element specific quality checks was an obvious choice. It enables a centralized specification of data elements in a structured, formalized, and standardized way.[13] In addition to conventional metadata elements covered by the ISO 11179 standard, such as the name, description, data type, and allowed values of data elements, we systematically identified further concepts that were necessary to avoid hard-coding of relevant information in the tool's code. These concepts were represented in an MDR, including, e.g., mapping information of a data element in different databases and data models and logical relations between data elements, the latter being necessary to model, e.g., plausibility checks.

Alignment to the Harmonized Data Quality Assessment Framework

With the growing interest and availability of EHR data in the last decades, many researchers have addressed the subject of systematically capturing and reporting DQ in such

datasets.[19–23] The lack of a standardized terminology related to DQ aspects in current literature motivated Kahn et al to initiate a community-based effort in collaboration with the Electronic Data Methods Forum (EDM)[2] to develop a harmonized three-category framework, which states that each of the categories conformance, completeness, and plausibility can be interpreted in the two contexts "verification" and "validation." The verification context evaluates the conformity of data values with their expected values, which can, for e.g., be obtained from metadata information, and can be determined without external resources. In contrast, in the validation context, the agreement of data values with external sources is covered, e.g., by comparing data to gold standards.[7]

The following description of the three DQ categories is based on Kahn et al's work[7]:

- Conformance features evaluate the concordance between the representation of the data and its definitions, e.g., by checking the agreement of data elements with certain rules, which refer to permitted values or value ranges and can be derived from the data model or data dictionaries (in the verification context).
- Completeness features evaluate the absence of data.
- The evaluation of plausibility characteristics, which refer to the credibility of data values, is done by assessing the value of one data element in the context of a second data element or over time. The subcategory "uniqueness plausibility" checks for multiple occurrences of data values in contexts where unambiguous assignments between two data elements are required. "Atemporal plausibility," for example, relates observed data values or frequency distributions to external knowledge or other independent resources.

This framework is well established and is used in a variety of scientific work.[24–31] To avoid linguistic misunderstandings, we considered it necessary and sensible to use this uniform harmonized DQA terminology in our present work.

Implementation of the Data Quality Concepts

For each of those three DQ categories, we have implemented at least one check in the DQA tool. Already existing implementations of these checks in the previous version of the tool were aligned with the harmonized DQA terminology. We furthermore continued and enhanced the tool's capability to assess DQ of two different databases simultaneously.

Conformance

Value conformance is assessed by checking the determined values of a data element against its allowed values as documented in the MDR. These checks can be performed separately for each database, allowing the definition of different constraints for corresponding data elements in different data sources (see also the "Plausibility" section below). Permitted values of continuous variables are expressed in the MDR by the respective minima and maxima, which can be used for comparison with the determined values. If either the determined minimum is smaller than the allowed minimum or the determined maximum is greater than the allowed maximum, an error message is generated, and the corresponding check is marked as "failed". We have currently implemented two alternative approaches to express and analyze discrete data elements in the DQA tool:

1. If the value set of a given data element comprises a small number of values (e.g., "principal diagnosis" or "secondary diagnosis" as constraint for the type of diagnosis), it is reasonable to specify the whole value set for the given data element in the MDR, along with choosing the validation type "list of permitted values".
2. For larger value sets (e.g., patient identifier), a regular expression can be defined as a constraint for permitted values in the MDR to describe an expected formatting along with choosing the validation type "string".

If the determined values of discrete data elements contain at least one value that is not specified in the allowed value set or represented by the respective regular expression, the corresponding check fails.

Completeness

The completeness criterion is assessed by counting the number of existing values or entries, missing values, and distinct values for each data element. When assessing DQ of two databases simultaneously, these counts are generated for each database separately. To assess the validity of the extract transform load (ETL) jobs, the counts for each data element in each database are compared and corresponding checks are marked as "passed" or "failed", respectively.

In addition, the (absolute and relative) amount of missing values for each data element is summarized in a table, with the results of the different databases displayed side by side for easy comparison.

Plausibility

To assess plausibility, the values of a data element are set in relation to one or more other data elements and checked for their meaningfulness. For this purpose, a logical link between different data elements is required. To implement such plausibility checks in a general way, a representation in JSON-format has been developed to define these checks in a structured manner and to store them per data element in the MDR.

Results

R Package "DQAstats"

We implemented the abovementioned methods in a new R package called *DQAstats*, using the "R" programming language.[32] This package contains a set of R functions to determine the DQ of one or two databases based on the criteria described by Kahn et al.[7] For convenience, the package provides a function **dqa()**, which wraps all required steps to perform the DQ checks into a single function call. Data from different data sources is imported into an R session and transformed according to the metadata definitions to

Table 1 Exemplary (atemporal) plausibility statements, adapted from Kapsner et al.[15] Each statement is accompanied by its corresponding expected values

Plausibility statement	Expected value
A diagnosis from ICD-10-GM chapter XV (pregnancy, childbirth, and puerperium) is only permitted for female patients.	Sex is female.
Malignant neoplasms of the male genital organs (ICD-10-GM C60-C63) are only permitted as hospital diagnosis for male patients.	Sex is male.

perform all statistical analyses and DQ checks. Finally, a PDF report is created, which contains all DQA results of the tested databases using the R packages *rmarkdown*[17,18] and *knitr*.[33–35] *DQAstats* is available on GitHub under the GPL v3 open-source license (see "Software Availability" below).

DQAstats analyzes DQ using the database-specific and data model-specific information represented in an MDR. A comma separated value (CSV) file can be used as the simplest method to provide *DQAstats* with structured metadata, comparable to an MDR. However, as demonstrated below, by providing suitable interfaces, the DQA tool can also be connected to established MDR implementations, such as *Samply.MDR*, if they provide machine-readable interfaces and allow for storing the DQA tool-relevant information. Unless otherwise stated we use the term "MDR" to refer to the CSV file-based MDR in the following.

MDR Representation of Plausibility Checks

A generic JSON object-based representation has been developed to store plausibility statements in the MDR and evaluate them automatically by the DQA tool (see **Appendix, C1** and **C2**).

Uniqueness Plausibility

For instance, the uniqueness plausibility rule "Every encounter ID may only be associated with one principal diagnosis" can be represented in the MDR in the context of the data element "encounter ID". The JSON object contains the information to associate the related data elements, here the "encounter ID" with the data element "principal diagnosis". Subsequently, the DQA tool evaluates, if any "encounter ID" is associated with multiple instances of "principal diagnosis" in the data.

Atemporal Plausibility

Similar to the uniqueness plausibility checks, atemporal plausibility rules were integrated into the MDR. ►Table 1 exemplarily shows two atemporal plausibility statements, as used in the MIRACUM consortium.

To automatically evaluate the atemporal plausibility check results, we further combined those checks with conformance checks to evaluate the results regarding the expected values.

If defined in the MDR, any of the abovementioned DQ checks are evaluated automatically during the execution of the DQA tool. If irregularities are detected, the tool logs the values that violate the DQ rules along with their identifiers to a CSV file, which can then be used as a starting point for further debugging efforts.

Coverage of Data Quality Checks

►Table 2 shows the currently within *DQAstats* implemented DQ checks compared to those proposed by Kahn et al's harmonized DQA framework.[7] For the verification context, each DQ category is covered by at least one implemented check.

Deployment within the MIRACUM Consortium

The R package *DQAstats* is the central component of the MIRACUM DQA framework. To connect it to the *Samply.MDR*-based M-MDR, 21 data elements represented in the M-MDR were complemented in a prototypical approach with the information necessary for the application of the DQA tool. Furthermore, we developed an M-MDR connector that transforms the CSV file-based MDR representation of each data element into a JSON object, which can be stored in a data element's so-called slot in *Samply.MDR*. These slots are

Table 2 *DQAstats* data quality assessment (DQA) checks coverage. The table shows the current state of the implementation of the DQA categories proposed by Kahn et al[7] into the *DQAstats* R package. For the verification context, each data quality category is covered by at least one check

Category	Subcategory	Verification context	Validation context
Conformance	Value conformance	Implemented	Missing
Conformance	Computational conformance	Missing	Missing
Conformance	Relational conformance	Missing	Missing
Completeness		Implemented	(Implemented[a])
Plausibility	Uniqueness plausibility	Implemented	Missing
Plausibility	Atemporal plausibility	Implemented	Missing
Plausibility	Temporal plausibility	Missing	Missing

[a]Strictly speaking, the source databases are not to be seen as validated gold standards and thus the term "verification context" might not be applicable; however, we use the term in this context to underline that the target databases should include the same information that is also present in the source databases.

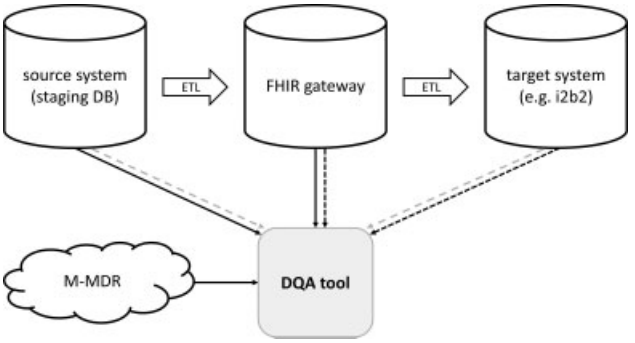

Fig. 1 DQA tool integration in the MIRACUM data integration center (DIC) infrastructure (schema). Within the DIC, pseudonymized data are transferred by ETL processes from the source systems via a FHIR gateway into the target research data repositories. Each combination of these ETL steps can be analyzed separately by the DQA tool. The solid lines depict the comparison between the source system and the FHIR gateway. The dark dashed lines show the comparison between the FHIR gateway and the target system. The gray dashed lines present the comparison of the source system and the target system. ETL, extract-transform-load. DQA, data quality assessment.

imposed by the ISO 11179 standard to allow for extensible data structures.[11,13] The M-MDR entries equipped with the information required by the DQA tool are publicly available at https://MDR-test.miracum.org/view.xhtml?namespace=dqa. All MIRACUM customizations, including the M-MDR connector, SQL statements, and further specific configurations are implemented in the R package *miRacumDQA*, which is also publicly available (see "Software Availability" below).

To deploy the DQA tool within the MIRACUM consortium, we have built a Docker image[36] with R, *DQAstats*, *miRacumDQA*, and all required dependencies preinstalled, which has been provided to the ten MIRACUM partners and which was successfully deployed and integrated into their respective DIC infrastructure in accordance with the MIRACUM project plan. The link to the local data repositories can be parametrized using environment variables that are provided to the Docker container. The partner sites use the DQA tool to regularly create DQ reports and to supervise and monitor their local data integration process.

►**Fig. 1** schematically presents the integration of the DQA tool in the MIRACUM DIC infrastructure. Pseudonymized data are transferred via ETL processes from the source systems via a FHIR gateway server[37] into the target research data repositories. The DQA tool is able to analyze each database on its own or to compare the analysis results of two independent databases, e.g., to validate the ETL jobs.

Extensibility to New Data Sources

When new harmonized data structures are defined, their data element specifications are also added to the M-MDR. The MIRACUM consortium currently uses two data repositories and their respective data models to store the research data: i2b2[38] and the OMOP[4,39] common data model (CDM). The previously reported version of the DQA tool only focused on the DQ evaluation of the i2b2 research data repository.[15] The present work includes the extension of the DQ checks to the MIRACUM OMOP repository,[40] the FHIR gateway server,[37] and a MIRACUM staging database by complementing the respective data elements in the MDR with the information required by the DQA tool as well as providing corresponding SQL statements.

Discussion

The demonstrated DQA tool provides several methods to automatically assess the DQ of data repositories, which exist as CSV files or SQL databases. It enables the application of a standardized set of DQA checks on the MIRACUM consortium's data repositories at each partner site and helps to detect irregularities. Such irregularities can be introduced by custom ETL jobs, for example, or may already exist in the source databases, both of which are possible points of intervention to continuously improve DQ. By extending its functionality to further MIRACUM research data repositories, we demonstrated its extensibility and applicability to new databases and data models, which is particularly important in the context of federated research networks. Furthermore, a structured and machine-processable representation of plausibility checks has been demonstrated, which can be stored in an MDR and allows them to be automatically evaluated by our tool, irrespective of the underlying database or data model.

The Advantages of Using a Metadata Repository

One of MIRACUM's requirements for testing DQ is the ability to easily adapt the DQA tool to new databases and data models. The use of an MDR enables us to define and maintain the DQ checks centrally and quickly adapt them to the variety of research data repositories in our research network. The agreed-upon standardized datasets and common data element definitions are required for our network-wide data analysis, whereas the source databases remain heterogeneous. The information that is uniform across all partner sites is stored centrally in the M-MDR. In addition, each site can also represent its local systems in a local MDR instance to extend the DQ checks to its site-specific systems. We believe that this offers a high degree of flexibility. To adapt the DQA checks of *DQAstats* to a new dataset, the respective metadata can be described in a text-based CSV file by default. However, as we have demonstrated, it is also possible to connect the DQA tool to MDR implementations, such as the *Samply.MDR*-based M-MDR.[13,14] In order to avoid the disadvantages in terms of comparability of the results that comes with assessing the DQ of different systems with different tools, our approach allows the DQA of all the systems in one tool.

Adoption of Kahn et al's Harmonized DQ Terminology

We have aligned the DQ checks implemented in our tool to the three DQ categories suggested in Kahn et al's harmonized DQA terminology.[7] System dependent information are loaded from the MDR each time before carrying out these checks on a specific database. If a user aims to assess DQ in two different databases simultaneously, the results of each check are further automatically compared between both databases. As described by Kahn et al, depending on the specific use case, verification and validation are both suitable for

assessing DQ. Validation, however, requires information that exists independent of the data source, such as external gold standards, which often make the verification of local EHR data the only applicable method.[7] For this reason, this proof-of-concept implementation focuses on analyzing DQ within the verification context. Furthermore, not all subcategories of Kahn et al's harmonized DQA terminology are currently covered (►Table 2). While the future addition of the subcategories "computational conformance" and "temporal plausibility" would be possible by implementing the respective logic into *DQAstats*, the integration of "relational conformance" checks would involve greater challenges. The latter checks evaluate the concordance of data elements that are given by the specific database or data model.[7] To implement these checks, one would need to harmonize the information on such structural constraints across different databases and to load these information in a standardized manner into R to evaluate the "relational conformance" for each database and compare the results across different databases with *DQAstats*. As indicated by Kahn et al in their manuscript, the completeness category is not further stratified into subcategories related to different reasons for "missingness."[7] In *DQAstats*, completeness is implemented in a rather basic manner by counting the number of existing values or entries, missing values, and distinct values for each data element. However, it is currently possible to test for an "expected missingness" of a data element in relation with the presence or absence of another data element with *DQAstats* by defining "atemporal plausibility checks".

Related Work

To check the DQ in EHR datasets, a variety of tools already exist, each with slightly different approach and advantage, some of which we would like to briefly introduce below and highlight the differences to our approach. The OHDSI community provides a DQ framework tailored to the OMOP CDM. Their *Data Quality Dashboard* is able to identify records that violate given specifications by executing over 3,300 checks, which are also following Kahn et al's framework.[28,41] Their R package *ACHILLES* characterizes the conformity of a database to the OMOP CDM.[39] In addition to that, an R script-based "ETL Unit test" framework can be accessed when using the OHDSI ETL tool *Rabbit-in-a-Hat*.[28] Schmidt et al[42] recently published a DQ framework for observational health research data collections with its software implementation in the R package "dataquieR."[43] Their tool is able to compute advanced statistical measures supporting the assessment of DQ, such as ANOVA, mixed effect models, and more.[42] *Samply.MDR* is also used as central MDR in other research networks.[5,13,44] Juárez et al[5] were able to demonstrate that common definitions stored in their *Samply.MDR* can be used by 10 participating sites of the German Cancer Consortium (DKTK)[45] to evaluate DQ with their "Quality Report Generator" tool. This evaluation is performed at each partner site within a so-called bridgehead after the ETL process. This step is implemented locally at each site, ensuring the conformance with local data protection guidelines and the ability to implement DQA only once for the harmonized data.[5] Furthermore and not specific to the medical domain, common ETL tools, such as Pentaho Data Integration (Hitachi Vantara LLC, Santa Clara, California, United States) or Talend Data Integration (Talend Inc., Redwood City, California, United States), provide specific features to combine DQ checks directly with the development of ETL jobs.

By evaluating DQ aspects only for the already transformed and harmonized data, issues introduced by, for example, extracting the data and transforming it into a new representation, are not considered. Therefore, it often remains unclear, if detected irregularities are already present in the original data source. The limitation of not analyzing DQ directly in the source systems has, e.g., also been noted by Juárez et al and is addressed by the idea of implementing DQ checks in parallel to developing ETL jobs, an approach that is being pursued by some ETL tools, such as *Rabbit-in-a-Hat*[28] or Talend Data Integration,[46] for example. Opposed to the approaches described above, our DQA tool is able to apply the same set of DQ checks and rules to two databases simultaneously. These checks are performed using the same underlying methods making the results inherently comparable, independent of the underlying databases or data models. In the context of EHR data, local source databases can be described in the MDR to apply these checks also to the site-specific systems. We believe this approach is beneficial because it enables to address DQ irregularities directly in the source databases with the effect that all subsequent tasks, whether research work or hospital routine, can potentially benefit from it.

Limitations

The goal of *DQAstats* is to provide a rather general overview of DQ in different databases, whose datasets are linked to each other, e.g., via ETL jobs or by belonging to a research network. Our DQA tool does not consider DQ in connection with specific (research) requirements to a given dataset, also called *fitness for use*.[21,47] While other tools exist to either assess DQ of data represented in CDMs (e.g., the OHDSI tools) or to provide additional detailed statistical insights for data of specific domains (e.g., dataquieR), we understand our DQA tool as complementary to those and find it helpful to get a quick and general overview of the DQ of specific datasets, providing a starting point for a focused exploration of possible irregularities. Furthermore, our tool focuses on analyzing DQ within the verification context and does not cover all subcategories of Kahn et al's harmonized DQA terminology. Additionally, only a minimal set of plausibility checks is available with this proof-of-concept implementation. Nevertheless, further rules can be added in the future, similar to those suggested by Wang et al.[30] However, by choosing the openly accessible programming language R for its implementation, designing the DQA tool as an R package itself and using an MDR for configuring both, new databases and new data elements as well as plausibility checks, we consider it as flexible and extensible so that future requirements can be supplemented accordingly.

Future Directions

To facilitate the operation of the DQA tool we are currently developing a graphical user interface as a frontend for *DQAstats* to provide users with an informative and interactive exploration of the DQA results. Furthermore, a usability evaluation of the tool is currently ongoing to identify improvable features. Regarding the connection to the M-MDR, functionalities such as defining relations between data elements in the MDR are currently redesigned and extended, considering, amongst other things, the requirements of *DQAstats*, which stores such information in one *Samply.MDR* slot at the moment.

Conclusion

We have developed a generic DQA tool and linked it to the central MIRACUM MDR. It has been provided to all 10 MIRACUM partners and was successfully deployed and integrated into their respective DIC infrastructure. The partner sites use the tool to regularly create DQ reports and their feedback is regularly collected to further improve the DQA tool's functionality. We have shown the extension of DQ checks to new databases and data models by extending the respective data elements in the MDR with the required information and by providing corresponding SQL statements. While a deeper understanding of the database to be connected is essential, no programming knowledge is required to represent this information in the MDR. The present work enables the MIRACUM consortium to centrally define a harmonized set of DQA rules in its M-MDR,[14] which can regularly be updated and extended.

Software Availability

The R package *DQAstats* is available on GitHub: https://github.com/miracum/dqa-dqastats. It is released under the GPL v3 open source license and provides an exemplary dataset and CSV-MDR, serving as a starting point for new users to adapt the DQA tool to their data.

The R package *miRacumDQA* contains MIRACUM-specific customizations and configurations and is available on GitLab: https://gitlab.miracum.org/miracum/dqa/miracumdqa.

Clinical Relevance Statement

We demonstrate a flexible and extensible data quality assessment (DQA) tool that can be implemented to assess the quality of data stored in various research data repositories for secondary use. By linking data quality checks to common data element definitions stored in a metadata repository and organizing them according to harmonized terminology, these checks can further be applied in a multisite research network in a structured and standardized manner.

Multiple Choice Questions

1. What are the minimal requirements to apply data quality checks with *DQAstats* to new SQL-based databases?
   a. Represent the databases' data element definitions in a YAML configuration file and store SQL statements in a metadata repository.
   b. Represent the databases' data element definitions in a CSV file-based metadata repository and provide corresponding SQL statements.
   c. Represent the databases' data element definitions in *Samply.MDR* and provide corresponding SQL statements.
   d. Represent the databases' data element definitions and corresponding SQL statements in *Samply.MDR*.

   Correct Answer: The correct answer is option b. Although *DQAstats* can be configured to new databases using *Samply.MDR*, the minimal requirement (and default method) is to represent the data element definitions of a new database in a CSV file-based MDR. Furthermore, corresponding SQL statements need to be provided apart from the MDR.

2. According to Kahn et al's harmonized data quality assessment (DQA) terminology, which checks are currently implemented in *DQAstats*?
   a. Value conformance checks, relational conformance checks, uniqueness plausibility checks, and atemporal plausibility checks.
   b. Value conformance checks, relational conformance checks, completeness checks, and atemporal plausibility checks.
   c. Value conformance checks, completeness checks, uniqueness plausibility checks, and atemporal plausibility checks.
   d. Relational conformance checks, computational conformance checks, completeness checks, and temporal plausibility checks.

   Correct Answer: The correct answer is option c. The following data quality checks according to Kahn et al's harmonized DQA terminology are currently implemented in *DQAstats*: Value conformance checks, completeness checks, uniqueness plausibility checks, and atemporal plausibility checks.

Protection of Human and Animal Subjects

Pseudonymized EHR data were used for developing and testing this software. No formal intervention was performed and no additional (patient-) data were collected. The authors declare that this research was performed in compliance with the World Medical Association Declaration of Helsinki on Ethical Principles for Medical Research Involving Human Subjects.

Funding

This work was funded in part by the German Federal Ministry of Education and Research (BMBF) within the Medical Informatics Initiative (MIRACUM Consortium) under the Funding Numbers FKZ: 01ZZ1801A (Erlangen), 01ZZ1801C (Frankfurt), and 01ZZ1801L (Dresden).

Conflict of Interest

None declared.

References

1 Helmer KG, Ambite JL, Ames J, et al; Biomedical Informatics Research Network. Enabling collaborative research using the Biomedical Informatics Research Network (BIRN). J Am Med Inform Assoc 2011;18(04):416–422

2 Holve E, Segal C, Lopez MH, Rein A, Johnson BH. The Electronic Data Methods (EDM) forum for comparative effectiveness research (CER). Med Care 2012;50(suppl):S7–S10

3 McMurry AJ, Murphy SN, MacFadden D, et al. SHRINE: enabling nationally scalable multi-site disease studies. PLoS ONE 2013;8 (Suppl 3):e55811

4 Hripcsak G, Duke JD, Shah NH, et al. Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers. Stud Health Technol Inform 2015;216(216): 574–578

5 Juárez D, Schmidt EE, Stahl-Toyota S, Ückert F, Lablans M. A generic method and implementation to evaluate and improve data quality in distributed research networks. Methods Inf Med 2019;58(2-03):86–93

6 Semler S, Wissing F, Heyder R. German medical informatics initiative: a national approach to integrating health data from patient care and medical research. Methods Inf Med 2018;57(01): e50–e56

7 Kahn MG, Callahan TJ, Barnard J, et al. A harmonized data quality assessment terminology and framework for the secondary use of electronic health record data. EGEMS (Wash DC) 2016;4(01): 1244

8 Brennan PF, Stead WW. Assessing data quality: from concordance, through correctness and completeness, to valid manipulatable representations. J Am Med Inform Assoc 2000;7(01):106–107

9 Kahn MG, Mis BBE, Bathurst J. Quantifying clinical data quality using relative gold standards. Paper presented at: AMIA Annual Symposium proceedings AMIA Symposium 2010:356–360

10 Hersh WR, Weiner MG, Embi PJ, et al. Caveats for the use of operational electronic health record data in comparative effectiveness research. Med Care 2013;51(08, Suppl 3):S30–S37

11 International Organization of Standardization (ISO) ISO/IEC 11179, Information Technology—Metadata Registries (MDR). Part 3: Registry Metamodel and Basic Attributes. 3 rd ed. published 2013–02–12). SC32 WG2 Metadata Standards Home Page; 2013

12 Prokosch H-U, Acker T, Bernarding J, et al. MIRACUM: Medical Informatics in Research and Care in University Medicine: a large data sharing network to enhance translational research and medical care. Methods Inf Med 2018;57(01):82–91

13 Kadioglu D, Breil B, Knell C, et al. Samply.MDR—a metadata repository and its application in various research networks. Stud Health Technol Inform 2018;253:50–54

14 MIRACUM Consortium. Miracum MDR. 2021. Accessed March 23, 2021 at: https://mdr.miracum.org/

15 Kapsner LA, Kampf MO, Seuchter SA, et al. Moving towards an EHR data quality framework: the MIRACUM approach. Stud Health Technol Inform 2019;267:247–253

16 Haverkamp C, Ganslandt T, Horki P, et al. Regional differences in thrombectomy rates : secondary use of billing codes in the MIRACUM (Medical Informatics for Research and Care in University Medicine) Consortium. Clin Neuroradiol 2018;28(02): 225–234

17 Allaire JJ, Xie Y, McPherson J, et al. Rmarkdown: dynamic documents for R. R package version 2.7, 2021. Accessed July 20, 2021 at: https://rmarkdown.rstudio.com

18 Xie Y, Allaire JJ, Grolemund G. R Markdown: The Definitive Guide. Boca Raton, FL: Chapman and Hall/CRC; 2018

19 Nasseh D, Nonnemacher M, Stausberg J. Datenqualität in der medizinischen Forschung: Leitlinie zum adaptiven Management von Datenqualität in Kohortenstudien und Registern. MWV Med Wissenschaftliche Verlagsgesellschaft; 2014

20 Johnson SG, Speedie S, Simon G, Kumar V, Westra BL. A data quality ontology for the secondary use of EHR data. Paper presented at: AMIA Annual Symposium proceedings AMIA Symposium; 2015:1937–1946

21 Strong DM, Lee YW, Wang RY. Data quality in context. Commun ACM 1997;40(05):103–110

22 Weiskopf NG, Weng C. Methods and dimensions of electronic health record data quality assessment: enabling reuse for clinical research. J Am Med Inform Assoc 2013;20(01):144–151

23 Weiskopf NG, Hripcsak G, Swaminathan S, Weng C. Defining and measuring completeness of electronic health records for secondary use. J Biomed Inform 2013;46(05):830–836

24 Khare R, Utidjian L, Ruth BJ, et al. A longitudinal analysis of data quality in a large pediatric data research network. J Am Med Inform Assoc 2017;24(06):1072–1079

25 Lee K, Weiskopf N, Pathak J. A Framework for Data Quality Assessment in Clinical Research Datasets. Paper presented at: AMIA Annual Symposium Proceedings; 2017:1080–1089

26 Callahan TJ, Bauck AE, Bertoch D, et al. A comparison of data quality assessment checks in six data sharing networks. EGEMS (Wash DC) 2017;5(01):8

27 Qualls LG, Phillips TA, Hammill BG, et al. Evaluating foundational data quality in the national Patient-Centered Clinical Research Network (PCORnet®). EGEMS (Wash DC) 2018;6(01):3

28 Observational Health Data Sciences and Informatics. The Book of OHDSI. Observational Health Data Sciences and Informatics. 2019. Accessed July 20, 2021 at: https://ohdsi.github.io/TheBookOfOhdsi/

29 Lynch KE, Deppen SA, DuVall SL, et al. Incrementally transforming electronic medical records into the observational medical outcomes partnership common data model: a multidimensional quality assurance approach. Appl Clin Inform 2019;10(05): 794–803

30 Wang Z, Talburt JR, Wu N, Dagtas S, Zozus MN. A rule-based data quality assessment system for electronic health record data. Appl Clin Inform 2020;11(04):622–634

31 Liaw S-T, Guo JGN, Ansari S, et al. Quality assessment of real-world data repositories across the data life cycle: A literature review. J Am Med Inform Assoc 2021:ocaa340

32 R Core Team. R: A Language and Environment for Statistical Computing. Vienna: R Foundation for Statistical Computing; 2020

33 Xie Y. Knitr: A Comprehensive Tool for Reproducible Research in r. In: Stodden V, Leisch F, Peng RD, eds. Implementing Reproducible Computational Research. Boca Raton, FL: Chapman and Hall/CRC; 2014

34 Xie Y. Dynamic Documents with R and Knitr. 2nd ed. Boca Raton, Florida: Chapman and Hall/CRC; 2015

35 Knitr XY. A General-Purpose Package for Dynamic Report Generation in R. R package version 1.31, 2021. Accessed July 20, 2021 at: https://yihui.org/knitr/

36 Merkel D. Docker: lightweight Linux containers for consistent development and deployment. Linux J 2014;239:2

37 Gruendner J, Gulden C, Kampf M, Mate S, Prokosch H-U, Zierk J. A framework for criteria-based selection and processing of fast healthcare interoperability resources (FHIR) data for statistical analysis: design and implementation study. JMIR Med Inform 2021;9(04):e25645

38 Murphy SN, Mendis M, Hackett K, et al. Architecture of the Open-source Clinical Research Chart from Informatics for Integrating Biology and the Bedside. Paper presented at: AMIA Annu Symp Procamia Annu Symp Proc.; 2007:5

39 Ryan P, Schuemie M, Huser V, Knoll C, Londhe A, and Taha Abdul-Basser. Achilles: Creates Descriptive Statistics Summary for an

Entire OMOP CDM Instance; R package version 1.6.7, 2019. Accessed July 20, 2021 at: https://github.com/OHDSI/Achilles
40 Maier C, Lang L, Storf H, et al. Towards Implementation of OMOP in a German University Hospital Consortium. Appl Clin Inform 2018; 9(01):54–61
41 DataQualityDashboard. Published 2021. Accessed April 6, 2021 at: https://ohdsi.github.io/DataQualityDashboard/index.html
42 Schmidt CO, Struckmann S, Enzenbach C, et al. Facilitating harmonized data quality assessments. A data quality framework for observational health research data collections with software implementations in R. BMC Med Res Methodol 2021; 21(01):63
43 Richter A, Schmidt CO, Krüger M, Struckmann S. DataquieR: assessment of data quality in epidemiological research. Journal of Open Source Software, 6(61), 3093, https://doi.org/10.21105/joss.03093
44 Lablans M, Kadioglu D, Mate S, Leb I, Prokosch H-U, Ückert F. Strategien zur Vernetzung von Biobanken. Klassifizierung verschiedener Ansätze zur Probensuche und Ausblick auf die Zukunft in der BBMRI-ERIC. Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz 2016;59(03):373–378
45 German Cancer Consortium, Available at: https://dktk.dkfz.de/en
46 Souibgui M, Atigui F, Zammali S, Cherfi S, Yahia SB. Data quality in ETL process: a preliminary study. Procedia Comput Sci 2019; 159:676–687
47 Juran JM, ed. Juran's Quality Handbook. 5th ed. New York, NY: McGraw-Hill; 1999

Appendices C1 and C2

Code C1: Uniqueness plausibility JSON representation (example). For each uniqueness plausibility check, the JSON object contains the respective DQA tool internal variable names as keys and another JSON object as value, which includes the rule's name, a description, and further information on how to filter the related variable to receive the desired results. The logical link between data elements is imposed by the location of this plausibility check in the metadata repository (MDR): Here, the encounter ID is to be analyzed in the context of the data element "encounter_diagnosis_rank," hence this statement has to be defined in the MDR as a part of the data element "encounter ID." The "filter" key is optional. If present, its value is a JSON object, containing key-value pairs for each database with a corresponding filter criterion. This filter criterion is applied to the contextually corresponding data element (here "encounter_diagnosis_rank") prior to the evaluation of the plausibility check. Note: the different filter values for the different databases are related to the respective mappings imposed by their data models/data dictionaries.

{ "uniqueness": **{** "encounter_diagnosis_rank": **{** "name": ["Pl.uniqueness.Item01"], "description": ["Every encounter ID may only be associated with one principal diagnosis."], "all_observations": ["1"], "filter": **{** "i2b2": ["DIAGNOSEART:HD"], "p21csv": ["HD"], "p21staging": ["HD"], "omop": ["44786627"] **} } } }**

Code C2: Atemporal plausibility JSON representation (example). For each atemporal plausibility check, the JSON object contains the respective DQA tool internal variable name as keys and another JSON object as value, which includes the rule's name, a description, and further information on how to filter the related variable to receive the desired results. The logical link between data elements is imposed by the location of this plausibility check in the metadata repository (MDR): Here, the data element "sex" is to be analyzed in the context of the data element "condition_code_coding_code," hence this statement has to be defined in the MDR as a part of the data element "sex." The data element "condition_code_coding_code" is the DQA internal variable name of the ICD-10 diagnosis code. The "join_crit" key stores a join criterion, which is the name of a third data element that might be required to automatically evaluate the plausibility statement (here "encounter_identifier_value," the DQA tool internal variable name for encounter ID), which is necessary for evaluating the data element "diagnosis code," which is in our case accompanied with the database key "encounter ID" in context with "sex," which is accompanied by the database key "patient ID." Therefore, the join of a third table holding the mapping between "encounter ID" and "patient ID" is required to perform this check. The JSON "join_crit" is automatically detected by *DQAstats* to connect—*join*—the data to generate a representation, which can then be used to evaluate such a plausibility check in a meaningful manner. The "constraints" key can be used to define the expected values of the atemporal plausibility check to subsequently perform a value conformance check with the results of the plausibility check.

{ "atemporal": **{** "condition_code_coding_code": **{** "name": ["Pl.atemporal.Item01"], "description": ["A diagnosis from ICD-10-GM chapter XV (pregnancy, childbirth and the puerperium) is only permitted for female patients."], "filter": **{** "omop": ["^O[0-9]"], "i2b2": ["^(ICD10\\:)O[0-9]"], "p21csv": ["^O[0-9]"], "p21staging": ["^O[0-9]"], "fhirgw": ["^O[0-9]"] **},** "join_crit": ["encounter_identifier_value"], "constraints": **{** "value_set": **{** "omop": ["w"], "i2b2": ["DEM|GESCHLECHT:w"], "p21csv": ["w"], "p21staging": ["w"], "fhirgw": ["female"] **} } } } }**

