Building a Data-Analytics Workflow For Antimicrobial-Resistance Studies: An Experience Report

Pei-Yu Hou
North Carolina State University
Raleigh, North Carolina, USA
hpeiyu@ncsu.edu

Jing Ao
North Carolina State University
Raleigh, North Carolina, USA
jao@ncsu.edu

Rada Chirkova
North Carolina State University
Raleigh, North Carolina, USA
rychirko@ncsu.edu

***Abstract*—In real-life analytics-oriented information-integration projects, the processes of information curation and integration cannot be completely automated. Rather, in each large-scale project the key objectives include maximizing scalability and throughput, while at the same time keeping the processes manageable and productive for the human experts in the loop. In this paper, we describe our experience with addressing these major objectives in the process of building a scalable end-to-end data-extraction, integration, and analytics workflow in the domain of antimicrobial resistance (AMR). The workflow is built using open-source tools, with the aims of enhancing the efficiency and accuracy of data collection and integration, while involving an acceptable level of efforts by collaborative multidisciplinary teams of humans-in-the-loop. We present the components of the proposed workflow, outline the challenges encountered in its development and testing, and discuss the experiences and lessons learned in enabling AMR experts and data analysts to interact with the workflow, with some of the lessons potentially applicable to other application domains.**

***Index Terms*—data analytics, data integration, antimicrobial resistance, experts-in-the-loop, analysts-in-the-loop**

I. Introduction

In a number of domains, including science and global health, large amounts of data are often collected independently by multiple teams or organizations over time. As the entities' needs evolve, in many cases such data need to be put together, typically to address the reporting and analytics needs of the organization. The resulting technical problems of integration of heterogeneous data have been studied for decades, see, e.g., [1]–[3]. At the same time, it is well recognized [4], [5] that in real-life analytics-oriented information-integration projects, the processes of information curation and integration cannot be completely automated; rather, in each large-scale project the key objectives include maximizing scalability and throughput, while at the same time keeping the processes manageable and productive for the human experts in the loop.

In this paper we describe our experience with addressing these major objectives in the process of building a collaborative data-integration and analytics workflow for the domain of antimicrobial resistance (AMR). AMR refers to the ability of a microorganism to cease an antimicrobial from working against it. It is considered to be "one of the most serious global public health threats in this century" [6], studied at both the national and international levels [7]–[10]. Very large volumes of AMR data have been accumulated worldwide.

In this work, our focus has been on integrating and analyzing large-scale AMR data in the context of longitudinal and related studies. (Through the NCSU College of Veterinary Medicine, the authors have access to over a terabyte of AMR data appropriate for these purposes.) These studies, aiming to discover and track historical changes in the degree of resistance to antimicrobials by bacteria serotypes,[1] specifically increases or decreases in their resistance to the same dosage of antibiotic medicine, attract considerable ongoing efforts [11]–[16]. The outcomes of such studies can provide useful information of assistance to governmental agencies, e.g., the FDA [17], in decision making involving the approval of safe and effective antimicrobial drugs. The information can also help pharmaceutical companies in developing adequate antibiotics. The ultimate goals of the research are to protect the public health and prevent a potential global health crisis.

In longitudinal and related studies, AMR experts use data that are relevant to tracking the AMR status both locally and globally. The data are collected both in the unprocessed (raw) form, e.g., data collected from farms, as well as in the form of summaries. Some of the summarized information is made publicly available by governmental agencies, e.g., [9], [18].

Historically, it has been challenging for AMR experts to extract data from published reports, as such extraction is typically performed manually from PDF files and hence does not scale. Further, the traditional process of manually integrating raw and summarized data can be tedious, time consuming, and error prone, with the additional challenge of enforcing data quality. There exist software tools that would allow AMR experts to automate to some degree the individual operations involved in the data extraction and integration. At the same time, to the best of our knowledge, there does not exist a general methodology – let alone a software framework or system – that would enable AMR scientists to address the problem of scalable and verifiable information extraction and integration

[1] A *serotype* is a distinct variation within a species of bacteria.

for longitudinal and related studies in an end-to-end manner.

This paper introduces a scalable collaborative end-to-end data-extraction, integration, and analytics workflow for AMR longitudinal and related studies. The workflow is built using open-source tools, with the aims of enhancing the efficiency and accuracy of the data collection and integration, while keeping the efforts of the humans in the loop at an acceptable level. We present the components of the proposed workflow, outline the challenges encountered in the work, and discuss the experiences and lessons learned in enabling AMR experts and data analysts to interact with the workflow, with some of the lessons potentially applicable to other application domains.

The remainder of the paper is structured as follows. In Section II, we introduce the data and process characteristics for the relevant AMR data analytics, and outline the challenges faced by AMR experts in the pipeline. Our proposed workflow, whose aims and tools are specified in Section III, is introduced in Sections IV–V. Section VI discusses the challenges that we faced in the workflow development and testing, as well as the lessons learned in the process. We conclude in Section VII.

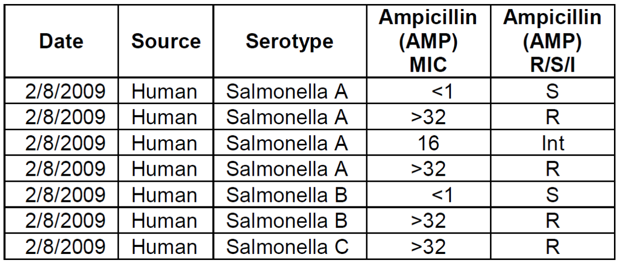
II. Background

The proposed scalable end-to-end data-extraction, integration, and analytics workflow for longitudinal and related studies has been developed iteratively in a collaboration between AMR experts and data analysts. We started the project by studying typical source data and data-processing practices that are traditionally used in AMR studies, and then articulated the main tasks of the envisioned workflow (see Section III). This section describes representative source data and typical processes applied to the data in their preparation for the analytics used in the AMR studies, as well as the major pain points for AMR researchers in these processes.

A. The Source Data

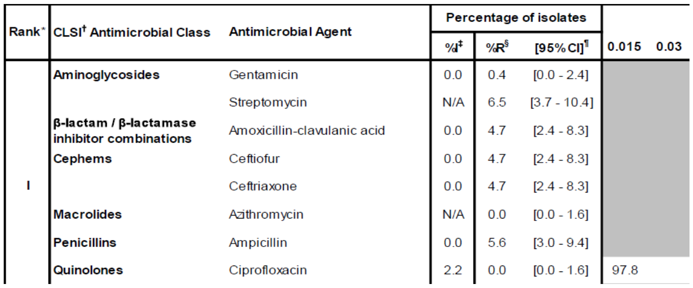
In their research, AMR experts use diverse sources of data. Raw data arise from experiments in the laboratories and are typically stored as spreadsheets. Summarized information comes from reports published annually by governmental agencies, including NARMS in the U.S. [18] and DANMAP in Denmark [9], and is formatted as tables in PDF files.

a) Raw Data Stored in Spreadsheets: The raw data in the format[2] shown in Fig. 1 were manually collected by NCSU College of Veterinary Medicine researchers during their visits to farms in the U.S. The collected data were screened in laboratories for certain types of antimicrobial drugs, and the *minimum inhibitory concentration (MIC)* indicator was recorded for each drug, with multiple levels of drug dosage. The antimicrobials tested for include, e.g., Amikacin (AMI) and Ampicillin (AMP). Based on the MIC levels, experts can interpret (label) each sample according to its degree of resistance to the antimicrobials, with possible values including "susceptible" (S), "intermediate" (Int), and "resistant" (R). Fig. 1 shows an example of possible raw spreadsheet data that describe Salmonella isolates collected from human samples.



Date	Source	Serotype	Ampicillin (AMP) MIC	Ampicillin (AMP) R/S/I
2/8/2009	Human	Salmonella A	<1	S
2/8/2009	Human	Salmonella A	>32	R
2/8/2009	Human	Salmonella A	16	Int
2/8/2009	Human	Salmonella A	>32	R
2/8/2009	Human	Salmonella B	<1	S
2/8/2009	Human	Salmonella B	>32	R
2/8/2009	Human	Salmonella C	>32	R

Fig. 1. The raw-data example for Salmonella isolates includes information, for each sample, on its collection date, source, serotype, MIC values for specific antimicrobials, and their interpretations (R/S/Int), with R standing for "resistant," Int for "intermediate," and S for "susceptible." For instance, the first row of the table represents a bacteria sample collected from a human on 02/08/2009; the serotype of this sample is *Salmonella A*, its MIC of using Ampicillin (AMP) is less than 1, and its interpretation is "susceptible" (S).



Rank*	CLSI† Antimicrobial Class	Antimicrobial Agent	Percentage of isolates				
			%I‡	%R§	[95% CI]¶	0.015	0.03
I	**Aminoglycosides**	Gentamicin	0.0	0.4	[0.0 - 2.4]		
		Streptomycin	N/A	6.5	[3.7 - 10.4]		
	β-lactam / β-lactamase inhibitor combinations	Amoxicillin-clavulanic acid	0.0	4.7	[2.4 - 8.3]		
	Cephems	Ceftiofur	0.0	4.7	[2.4 - 8.3]		
		Ceftriaxone	0.0	4.7	[2.4 - 8.3]		
	Macrolides	Azithromycin	N/A	0.0	[0.0 - 1.6]		
	Penicillins	Ampicillin	0.0	5.6	[3.0 - 9.4]		
	Quinolones	Ciprofloxacin	2.2	0.0	[0.0 - 1.6]	97.8	

Fig. 2. Squashtogram fragment from [18]; the squashtogram contains data for serotype *Salmonella ser. Newport* in year 2015. The first row provides the statistics for using drug Gentamicin on *Salmonella ser. Newport.* Specifically, the percentage of "intermediate" values (%I) is 0%, and of "resistant" (%R) is 0.4%, with the 95% confidence interval for %R (95% CI for %R) is [0.0% – 2.4%] . The remaining columns show the distribution of MIC. The MIC values are not shown (except for the last row), due to most of the values being between 0.25 and 32, while the table presents only the values between 0.015 and 0.03, as shown in the column header; see [18] for the details.

b) Summarized Information Stored as Tables in Reports: For each serotype being monitored, annual reports of government agencies, published as PDF files, provide summary tables called *squashtograms,* which describe the status of AMR among enteric bacteria isolated from humans in that year. Fig. 2 shows a typical example taken from the NARMS 2015 report [18]. The squashtogram shown in Fig. 2 has attributes Rank, CLSI Anitimicrobial Class, Antimicrobial Agent, Percentage of isolates (%I, %R, and 95% CI for %R), and the distribution of MIC. The contents of the squashtogram reflect the status of antimicrobial resistance for serotype isolates with respect to each antimicrobial agent tested for.

B. Summarizing the Raw Data

There is a rather straightforward relationship between raw data stored in the format of Fig. 1 and squashtograms storing data in the format of Fig. 2. We now outline this relationship and explain how it is used in summarizing raw data in the AMR analytics pipeline. The outcomes of summarizing raw data in the format of Fig. 1 match the format of squashtograms

[2] Due to data-confidentiality issues, the data shown in Fig. 1 are the result of altering original data collected by the researchers.

as in Fig. 2. As a result, the integration step is simple – intuitively, the process is to append the rows of the summarized raw data to the rows of the squashtograms.

Consider, for the squashtograms storing data in the format of Fig. 2, the columns for the percentage of isolates for each antimicrobial agent tested. In these columns, the value of `%I` (percentage of "intermediate," Int) indicates the proportion of samples, for a given serotype, whose MIC value is in a prespecified range between the boundaries of the levels considered to be "resistant," R, and "susceptible," S, for the same serotype. Similarly, `%R` (percentage of "resistant," R) indicates the proportion of samples for a serotype whose MIC value exceeds a specific threshold. (These two values are critical for assessing whether the AMR of the serotype shows an increasing or decreasing trend.) The `95% CI for %R` value helps the users to determine if the result is reliable.

To illustrate how the values of `%I, %R,` and `95% CI for %R` in squashtograms are obtained from raw data, we use the raw data for serotype Salmonella A in the first four rows of Fig. 1. Using these data, the value of `%I` for AMP is calculated as 25% (due to Int being the value of AMP R/S/I in just one of these four rows). Using similar calculations, we obtain that the value of `%R` for AMP is 50%. Finally, the value of `95% CI for %R` is computed as $[CI_L, CI_U]$, where

$$CI_L = p - z_{1-\alpha/2}\sqrt{\frac{p\,(1-p)}{n}} \tag{1}$$

$$CI_U = p + z_{1-\alpha/2}\sqrt{\frac{p\,(1-p)}{n}} \tag{2}$$

From these formulae and with the values of p = `%R` (= 0.5, see above); n = 4 (the sample size for Salmonella A in Fig. 1); and α = 0.05: We arrive at $CI_L = 0.01$ and at $CI_U = 0.99$. As a result, we obtain `%I` = 25%, `%R` = 50%, and `95% CI for %R` = [1 - 99] (%) for the AMP R/S/I column for Salmonella A for the data in Fig. 1. In the remainder of the paper, we use analogous calculations in aggregating the given raw data for each antimicrobial agent tested on each serotype.

C. Challenges Faced by AMR Experts

In analyzing the traditional pipeline of data collection and integration for AMR studies as described above, we articulated the following pain points for AMR experts:

1) Manual extraction of squashtograms from PDF documents is tedious, time consuming, and error prone;
2) Traditional raw-data summarization into squashtograms using software such as Excel is not scalable;
3) Even though software tools are available for steps in the data-extraction and integration process, finding out which tools would be applicable and how to build the overall pipeline for each individual AMR project would be labor intensive, with potentially nontrivial learning-curve and debugging efforts; and
4) Data quality can be an issue for integrating disparate large-scale data sources. That is, if some of the pipeline components produce erroneous outputs, the final conclusions may be incorrect (cf. [19]).

The proposed workflow addresses these pain points, as described in the following sections.

III. The Workflow: Objectives and Tools

We now begin outlining the design of the proposed scalable end-to-end workflow for information integration and analysis for AMR longitudinal and related studies. The workflow incorporates collaborations among humans-in-the-loop in two capacities: (1) The role of *experts-in-the-loop* includes providing feedback on the correctness of the outcomes of individual workflow stages, as well as selecting inputs and postprocessing the outputs for their purposes; this role is taken on by AMR experts. (2) The role of *analysts-in-the-loop* includes guiding the data through the workflow pipeline, tuning the steps of the pipeline in consultation with experts-in-the-loop, and, finally, driving the feedback loops in the workflow; this role is taken on by data analysts. We now describe the workflow objectives and the software tools selected for its implementation.

A. Objectives of the Workflow

With the focus on the `%I, %R,` and `95% CI for %R` values for antimicrobials with respect to bacteria serotypes, the following steps need to be performed to enable analytics over the integrated squashtograms and summarized raw data:

1) Aggregate raw data into the format of squashtograms (source I: raw-data spreadsheets);
2) Extract squashtograms from reports into readable structured tables (source II: PDF files); and
3) Integrate the squashtograms from both sources.

Data cleaning is incorporated into the feedback loop in each stage, to enhance the effectiveness of the data processing.

B. Software Tools Used in the Workflow

Toward achieving the objectives of the workflow, we looked for software with appropriate functions. Many tools are available these days for data cleaning, integration, and analytics. Our selection criteria included ease of use, flexibility, and the software being open source; we ended up using R [20], SQLite [21], and Tabula [22]. For table extraction from PDF files, Tabula [22] provides a web-based user interface appreciated by experts-in-the-loop. As in the workflow we need to save the inputs and outputs of each procedure into a database, for the database functionalities the workflow uses the powerful and easy to use SQLite [21]. Finally, R [20], a powerful open-source statistical programming language, provides functionalities needed for implementing statistical methods [23], processing multiple types of data [24] including time [25] and strings [26]–[28], performing powerful data visualization [29]–[31], and connecting with SQLite databases [32].

IV. The Workflow: Forward-Flow Phases

Our proposed collaborative data-integration and data-analytics workflow for AMR performs data processing in

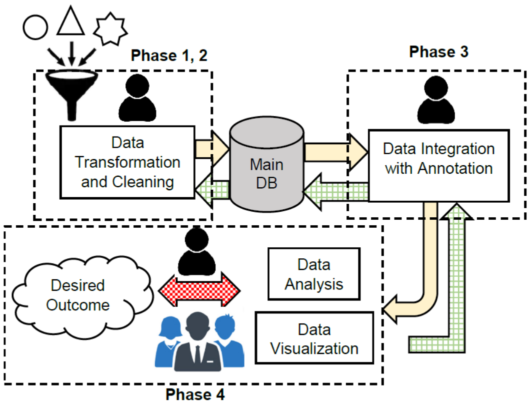


Fig. 3. An overview of the proposed collaborative data-integration and data-analytics workflow for the AMR domain. Individual stages of the workflow, represented as boxes, are mapped to specific forward-flow phases (1 through 4, described in Sections IV-B – IV-E). In this Fig., as well as in Figs. 4 and 6, databases are represented by cylinders, and involvement by analysts-in-the-loop in individual stages is indicated via individual-person symbols. Further, the solid arrows indicate the forward data-and-control flow (Section IV) in the workflow, and the directed checkered arrows show the feedback flow (Section V) focused on working with inappropriate or otherwise surprising results arising from individual phases. The two-way arrow between the workflow outputs and the desired outcomes represents validation of individual workflow iterations by experts-in-the-loop, indicated using the group symbol.

multiple forward-flow phases, including spreadsheet aggregation, table extraction from summary reports, integration of the resulting data, and analytics over the outputs. In this section we provide details on the forward flow of the workflow; Section V discusses the feedback loops built into the workflow.

A. Workflow Overview

Fig. 3 outlines the proposed collaborative data-integration and data-analytics workflow for the AMR domain. The workflow inputs are selected by experts-in-the-loop. The data-processing steps carried out in the first two phases help improve the performance of the downstream data integration (phase 3) and analytics (phase 4). Toward these goals, phases 1 and 2 of the workflow smooth errors in the input data, unify the representation of all the data into a structured format,[3] and store the data into a database. (All the intermediate and final outputs of the workflow are stored in the same database.) Specifically, phase 1 (Section IV-B) cleans and aggregates raw spreadsheet data, and phase 2 (Section IV-C) extracts squashtograms from summary reports. (Note that it is not necessary for phase 2 to follow phase 1; instead, the two phases can be carried out in parallel, as they process different types of data.) Then, after the data integration done in phase 3 (Section IV-D), data analytics and visualization are applied to the data in phase 4 (Section IV-E). Analysts-in-the-loop are involved in all the four phases; they interact with experts-in-the-loop at the postprocessing stage of achieving the final outcomes of the analytics, and also (as described in Section V) in all the feedback-loop discussions and decisions.

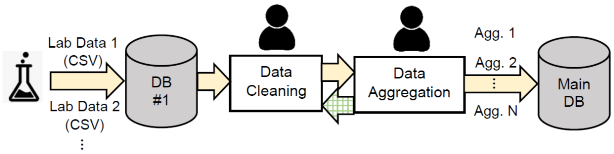


Fig. 4. Workflow phase 1: Spreadsheet-data cleaning and aggregation. Individual processes are represented as boxes. `Agg.1 ... Agg.N` refer to squashtograms obtained by aggregating raw data from laboratory analyses.

B. Phase 1: Spreadsheet-Data Cleaning and Aggregation

Traditionally, AMR domain experts use software such as Excel to summarize raw data into squashtograms. As this approach is not scalable, some level of automation is called for if the process is to be applied to large amounts of data.

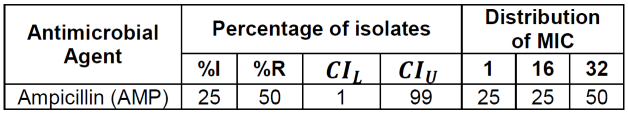
Phase 1 of the proposed workflow semi-automates the data-aggregation process; Fig. 4 provides an outline. In this phase, the input spreadsheet data supplied by experts-in-the-loop are loaded into the database. Then analysts-in-the-loop perform cleaning of the stored data, including removal of the leading and trailing whitespaces, as well as correcting typos and other data errors. For these purposes, phase 1 of the workflow uses standard automated data-analytics functions, such as the `trimws` function, as well as simple statistical functions, such as `counts()` or `table()` in R.

Following the data-cleaning part of phase 1, analysts-in-the-loop aggregate the data using an algorithm developed by the authors of this paper, see Algorithm 1 for the pseudocode. The algorithm takes as input the portion of raw spreadsheet data for a fixed year and fixed bacteria serotype, and implements the approaches described in Section II to calculate the percentage of intermediate (`%I`) and resistant (`%R`) isolates for the year and serotype, as well as the associated necessary statistics, including `95% CI of %R` and the distribution of `MIC`. To facilitate the downstream integration with the squashtograms extracted from summary reports in phase 2 (Section IV-C), Algorithm 1 returns the aggregation outputs in the squashtogram format, see Fig. 5 for an illustration. The aggregated outputs also contain additional columns, with information about the serotypes and years used for aggregating the spreadsheet data. Finally, the outputs of Algorithm 1 get adorned with an additional `dataSource` column, whose values indicate that the data were derived from raw laboratory data, as well as potentially the sources of the samples (e.g., humans). The column addition is done to make compatible, in phase 3 (Section IV-D) of the workflow, the schemas of the aggregated tables and of the squashtograms extracted in phase 2 (Section IV-C) from published summary reports.

All the squashtograms returned by Algorithm 1 are loaded into the main (SQLite) database. The aggregated information is now ready to be examined by analysts-in-the-loop, potentially in consultation with experts-in-the-loop.

Phase 1 of our proposed workflow makes the process of aggregating spreadsheet data into the squashtogram format more efficient than the traditional baseline used by AMR scientists. Phase 1 accomplishes this objective by supplying

[3] In this project we have focused on data represented via relations.



Antimicrobial Agent	Percentage of isolates				Distribution of MIC		
	%I	%R	CI_L	CI_U	1	16	32
Ampicillin (AMP)	25	50	1	99	25	25	50

Fig. 5. Fragment of data aggregated by Algorithm 1 into the template of Fig. 2. Using the data of Fig. 1 for `Salmonella A`, the value of `%I` is 25, `%R` is 50, and `95% CI of %R` is [1 - 99], see Sec. II for the details. (The columns for `year`, `serotype`, and `dataSource` are not shown.)

analysts-in-the-loop with Algorithm 1, as well as by taking advantage of the existing data-analytics functions and packages in R. Note that the traditional domain-specific time-consuming and labor-intensive iterative data cleaning, data aggregation, and parameter setup in software such as Excel have all been replaced by Algorithm 1. This potential for efficiency improvements is significant, as it can significantly decrease the time and efforts required for data analysts in their processing potentially large volumes of input spreadsheet data.

Algorithm 1: Spreadsheet-Data Aggregation

```
Data: Subset of spreadsheet data selected based on fixed
      values year and serotype.
Result: Squashtogram with fields $I$, $R$, $CI_L$, $CI_U$,
        $MIC$, $year$, and $serotype$.
begin
    $I = \varnothing, R = \varnothing, CI_L = \varnothing, CI_U = \varnothing, MIC = \varnothing$;
    for each antimicrobial agent $i$ do
        $I_i \leftarrow$ occurrence percentage of string ‘Int’;
        $R_i \leftarrow$ occurrence percentage of string ‘R’;
        $CI_{L_i} \leftarrow R_i - z_{1-\alpha/2}\sqrt{\frac{R_i(1-R_i)}{n}}$, see Eq. (1);
        $CI_{U_i} \leftarrow R_i + z_{1-\alpha/2}\sqrt{\frac{R_i(1-R_i)}{n}}$, see Eq. (2);
        $I \leftarrow I \cup I_i$; /* ∪ denotes row binding
            of values into a column */
        $R \leftarrow R \cup R_i$;
        $CI_L \leftarrow CI_L \cup CI_{L_i}$;
        $CI_U \leftarrow CI_U \cup CI_{U_i}$;
    end
    for each antimicrobial agent $i$ do
        for each MIC dosage $j$ do
            $MIC_{ij} \leftarrow$ occurrence percentage of $j$;
            $MIC_i \leftarrow MIC_i \cup MIC_{ij}$;
        end
        $MIC \leftarrow MIC \cup MIC_i$;
        $year \leftarrow$ year;
        $serotype \leftarrow$ serotype;
    end
    $S \leftarrow$ combine the columns $I$, $R$, $CI_L$, $CI_U$, $MIC$,
        $year$, and $serotype$ into squashtogram;
    return S.
end
```

C. Phase 2: Table Extraction from Summary Reports

In the traditional AMR expert-driven data-analytics pipeline, extracting squashtograms from summary reports is not easy to perform. The challenges include high time and labor costs of manual efforts, as well as potential unfamiliarity of AMR experts with other available software tools. This barrier typically prevents AMR experts from getting more comprehensive downstream data-analytics results, as, clearly, it is not easy to perform joint analysis over the aggregated spreadsheet data and tables from summary reports without having access in the database to the tables extracted from the reports. In the proposed workflow, support for summary-table extraction from reports plays an important role in empowering both analysts-in-the-loop and experts-in-the-loop with new important types of downstream data analytics on AMR data.

The proposed workflow makes the process of table extraction from summary reports easier for analysts-in-the-loop, by taking advantage of existing open-source tools such as Tabula [22]. As shown in Fig. 6, analysts-in-the-loop can start this phase by downloading publicly available source summary reports selected by experts-in-the-loop; such reports are typically available in the form of PDF files [9], [18]. Then analysts-in-the-loop, guided by the choice of individual tables in the reports by experts-in-the-loop, can automatically capture and extract the selected tables from the PDF files using Tabula. Note that detecting the tables to be extracted requires a collaboration between machines and humans-in-the-loop: In our experience, analysts-in-the-loop have manually checked whether the table areas automatically detected in the PDF files by Tabula are correct and the tables are indeed the tables preselected by experts-in-the-loop. If the captured tables have been detected correctly, they are next automatically extracted and loaded into the main database as the output of phase 2. (Before being loaded into the database, the captured squashtograms get augmented with an additional `dataSource` column, with information about the type of report, e.g., NARMS, from which the data were derived, as well as with columns with information about the serotypes and years for which each squashtogram had been developed, see caption to Fig. 2 for an illustration. The column addition is done to make compatible, in phase 3 of the workflow, see Section IV-D, the schemas of the final squashtograms and of the aggregated tables obtained in phase 1, see Section IV-B, from raw data.) Analysts-in-the-loop can then discuss the final squashtograms with experts-in-the-loop. If further cleaning of the final squashtograms is needed, our workflow also provides a method to assist analysts-in-the-loop in the cleaning; the method builds on Algorithm 2 developed by the authors of this paper, and is discussed in detail in Section V.

D. Phase 3: Table Integration

As outlined in Fig. 3, phase 3 of the proposed workflow integrates the aggregated spreadsheet data obtained in phase 1, see Section IV-B, with the summary information extracted in phase 2 from published reports, see Section IV-C.

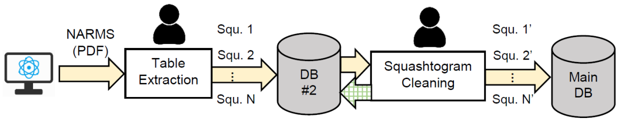
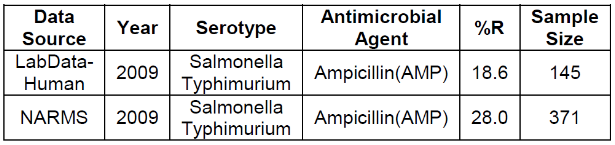


Fig. 6. Workflow phase 2: Table extraction from summary reports (e.g., NARMS [18]). Individual processes are represented by boxes. `Squ.1 ... Squ.N` refer to squashtograms extracted from reports, while `Squ.1' ... Squ.N'` refer to the respective cleaned tables output into the main database.



Data Source	Year	Serotype	Antimicrobial Agent	%R	Sample Size
LabData-Human	2009	Salmonella Typhimurium	Ampicillin(AMP)	18.6	145
NARMS	2009	Salmonella Typhimurium	Ampicillin(AMP)	28.0	371

Fig. 7. Example of integrated data for 2009 Ampicillin (AMP) testing for *Salmonella Typhimurium*: The first row shows the value of `%R` obtained by aggregating the values for 145 samples of raw data sourced from laboratory analyses; the value of `%R` in the second row comes from a summary report. (For readability, some of the required columns are not shown.)

The data integration is started by analysts-in-the-loop in phase 3 of the workflow by importing into R from the database the stored aggregated raw tables and squashtograms, and by then using functions such as `rbind()` in R to perform row binding [33], which puts all the input rows into the same table. (Recall that phase 1 and phase 2 have been designed in such a way that their outputs have the same schema – Fig. 3 refers to the use in phase 3 of this schema unification as "annotation.")

Via phases 1–3, our workflow assists analysts-in-the-loop in connecting the raw data available from spreadsheets to the information provided by published summary reports. As discussed earlier, integrating data from the two kinds of sources enables AMR experts-in-the-loop to perform on the resulting data certain types of analytics that would otherwise be challenging or even impossible to apply.

E. Phase 4: Data Analytics over the Integrated Tables

Recall that in the traditional expert-driven AMR pipeline, data analytics over integrated data is not immediately feasible, due to challenges arising in table extraction from summary reports and hence in data integration. In phase 4 of the proposed workflow, see Fig. 3, analysts-in-the-loop take advantage of the integrated information, by performing on it data analytics and visualization. In the remainder of this section we provide two examples of possible analytics, and then summarize the data-analytics experience of the team with the proposed workflow.

1) Hypothesis Testing: Recall that in preparation for phase 3 (data integration), the data in the workflow are enhanced with additional information, to harmonize the schemas of the integration inputs. After integrating the harmonized data, analysts-in-the-loop can enable hypothesis testing to check if there is a statistically significant difference, for specific serotype and year values, between the aggregated raw data from laboratory sources and the summary reports.

For example, suppose experts-in-the-loop would like to see if the percentages of resistance (`%R`) for *Salmonella Typhimurium* are different between the raw data available for 2009 and the 2009 NARMS data. Analysts-in-the-loop can locate in the integrated data the values of `%R` and the respective sample sizes for the serotype for 2009, see Fig. 7. A two-sample proportion test [34] can then be performed on the highlighted data, to determine whether, for the given `%R` values and sample sizes, there is a statistically significant difference between the data from the two data sources. Suppose p_1 is the value of `%R` from the raw data, and p_2 is `%R` from the NARMS data. The hypothesis would be $H_0 : p_1 = p_2$ vs. $H_1 : p_1 \neq p_2$. Using the data in Fig. 7, we have $\hat{p_1} = 0.186$, $n_1 = 145$; $\hat{p_2} = 0.28$, and $n_2 = 371$. Then the test statistic is $z = -220.6$, indicating that the p-value is approximately zero under the assumption that the proportion difference follows a standard normal distribution. As a result, analysts-in-the-loop can conclude that the percentages of resistance (`%R`) between the two data sources are statistically different for these values of serotype and year. This result should be further discussed with experts-in-the-loop. After the communication, feedback loops might be activated, as outlined in Section V.

2) Visualization: `%R` and the corresponding value of `95% CI` over time are major measurements tracked in summary reports such as NARMS [18] and DANMAP [9]. To enable comparisons between summarized raw-data results and the information published in summary reports, analysts-in-the-loop can use time plots, see, e.g., Fig. 8. On our team, experts-in-the-loop have reported that working with visualizations such as the one shown in Fig. 8, combined with the hypotheses testing described earlier, facilitates value comparisons over time for the AMR experts' research purposes, and enables them to formulate further research questions, some of which can also be addressed within the proposed workflow.

3) Enabling Better Data-Analytics Capabilities: In our experience, phases 1–4 of the proposed workflow have assisted analysts-in-the-loop on our team in enabling experts-in-the-loop to use data analytics that would be challenging to do or even not feasible using the traditional expert-driven pipeline. As an example, comparing AMR trends between different countries is easier to do with the proposed workflow: Analysts-in-the-loop and experts-in-the-loop can take this opportunity as the next step in examining the global AMR trends.

V. Feedback Loops in the Workflow

In this section we discuss the feedback loops built into the proposed workflow. The purpose of the feedback loops is to assist analysts-in-the-loop, in collaboration with experts-in-the-loop, in identifying and correcting potential errors made in the forward-flow phases of the workflow.

A. General Overview

Due to the presence of multiple forward-flow data-processing stages in the proposed workflow, see Fig. 3, and to the involvement in the workflow of humans-in-the-loop with different backgrounds, it could be challenging for analysts-in-the-loop to pinpoint, without help from experts-in-the-loop,

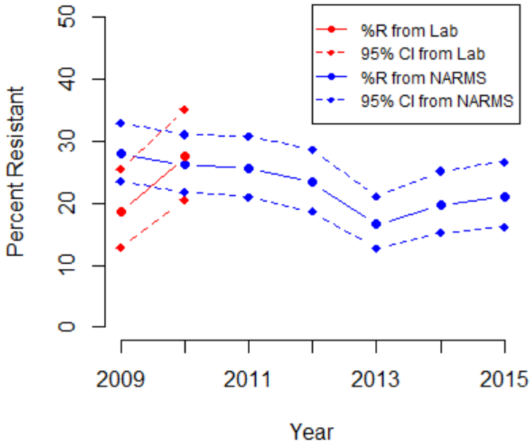


Fig. 8. Example visualization over integrated data, showing the percentage of resistance (`%R`, solid lines) and its 95% confidence interval (`95% CI`, dashed lines) of *Salmonella Typhimurium* using Ampicillin. The red lines shown for the years 2009–10 represent aggregations over raw data (cf. Fig. 1). The blue lines represent NARMS data for 2009 through 2015 (cf. Fig. 2).

the causes of any anomalous data-analytics outputs. To address this issue, we introduced a feedback loop into each data-processing phase, to help analysts-in-the-loop revise, together with experts-in-the-loop, the work done in that phase, thus ensuring acceptable quality of the final data-analytics results.

Generally, for each forward-flow workflow phase, analysts-in-the-loop collect the phase outputs as feedback that is used jointly with experts-in-the-loop to evaluate the appropriateness of the work done in the phase. The workflow also admits domain knowledge as another form of feedback if needed.

In the rest of this section we describe the feedback loop for each forward-flow phase, and discuss how collaborations between analysts-in-the-loop and experts-in-the-loop can leverage the feedback to tune the forward flows of the workflow.

B. Feedback Loop for Phase 1

For the phase of spreadsheet-data aggregation, the feedback loop (checkered arrow in Fig. 4) takes as input the resulting aggregated values. By examining these values and discussing them with experts-in-the-loop as needed, analysts-in-the-loop can discover potentially missing data-cleaning steps or erroneous aggregation steps in the work performed in the phase.

a) Data Cleaning: As an illustration, if some aggregated values output by phase 1 are associated with empty serotype values, analysts-in-the-loop might notice that the original spreadsheet data used for the aggregation contain rows with missing values in the *Serotype* column. This realization would prompt the analysts to decide, together with experts-in-the-loop, whether to impute the missing values or to delete entire rows in the input data. Further, if some output values are associated with very similar names for serotypes, e.g., "*Salmonella Typhimurium C*" vs. "*Salmonella Typhimurium.c*"', analysts-in-the-loop could posit that the *Serotype* column in the input data could contain typos and, in consultation with experts-in-the-loop, brainstorm ways for correcting them.

b) Data Aggregation: This feedback loop could also be helpful in identification of erroneous aggregation steps. For example, suppose that the aggregated percentage of resistance (`%R`) for Ampicillin (AMP) for *Salmonella A* is 25%, vs. 50% as shown in Fig. 5. When consulted about this discrepancy, experts-in-the-loop could inform analysts-in-the-loop that the aggregated percentage is erroneous. In this case, analysts-in-the-loop could recheck the aggregation algorithms of phase 1 for errors, while at the same time making sure that the anomaly is not caused by incorrect data-cleaning steps.

Algorithm 2: Finding Columns For Value Splitting

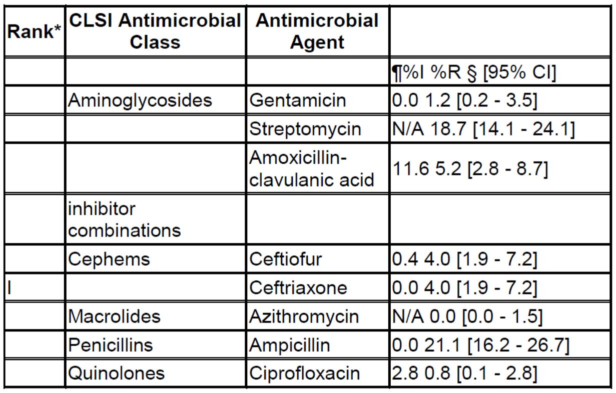
```
Data: Multilevel squashtogram headers $I$ with $n$ layers
      and $m$ columns, target (correct) squashtogram
      headers $T$ with $n$ layers and $m$ columns.
Result: Index of column designated for value splitting.
begin
    $j \leftarrow 1$; /* Checking starts from 1          */
    $M_j \leftarrow 1$;
    while $M_j > 0.7$ do
        /* user-defined threshold 0.7                  */
        for $i \leftarrow 1..n$ do
            $c_{ij} \leftarrow$ stringsim($I_{ij}$, $T_{ij}$);
            /* string-similarlity function;
               others can be used                      */
        end
        $M_j \leftarrow \frac{\sum c_{ij}}{n}$;
        $j \leftarrow j + 1$;
    end
    return $j$.
end
```

C. Feedback Loop for Phase 2

For the phase of extracting tables from summary reports, the feedback loop (checkered arrow in Fig. 6) takes as input the extracted tables. For instance, when using Tabula to extract tables, analysts-in-the-loop could find that multiple columns in the resulting tables are squeezed together into a single column, as shown in Fig. 9. Since this result is not appropriate for downstream data processing, analysts-in-the-loop can retrace the extraction process in search of possible solutions.

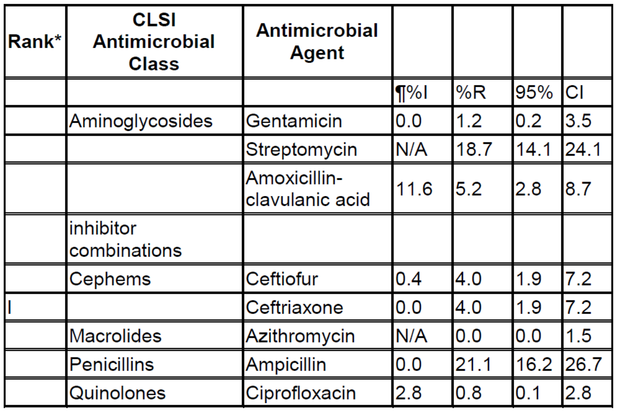
The problem illustrated in Fig. 9 is caused by the input squashtograms tables having multiple layers in their headers, see the first two rows in Fig. 10 (cf. Fig. 2). A straightforward way to address this issue would be to manually split up the erroneous columns. However, this solution does not scale in the number of tables to be extracted. Moreover, manual correction may introduce typos and other errors.

To aid analysts-in-the-loop in addressing this issue, we have developed an algorithm to split up the erroneous columns, see Algorithm 2 for the pseudocode. Given the erroneous table headers (e.g., the first two rows in Fig. 9) as the source, and the correct table headers provided by analysts-in-the-loop (e.g., the first two rows in Fig. 10) as the target, the algorithm identifies



Rank*	CLSI Antimicrobial Class	Antimicrobial Agent	
			¶%I %R § [95% CI]
	Aminoglycosides	Gentamicin	0.0 1.2 [0.2 - 3.5]
		Streptomycin	N/A 18.7 [14.1 - 24.1]
		Amoxicillin-clavulanic acid	11.6 5.2 [2.8 - 8.7]
	inhibitor combinations		
	Cephems	Ceftiofur	0.4 4.0 [1.9 - 7.2]
I		Ceftriaxone	0.0 4.0 [1.9 - 7.2]
	Macrolides	Azithromycin	N/A 0.0 [0.0 - 1.5]
	Penicillins	Ampicillin	0.0 21.1 [16.2 - 26.7]
	Quinolones	Ciprofloxacin	2.8 0.8 [0.1 - 2.8]

Fig. 9. Example of table extracted using Tabula; we can see that data from columns `%I` through `95% CI` are squeezed into a single column. As a result, the information in this table cannot be used for downstream data analysis.



Rank*	CLSI Antimicrobial Class	Antimicrobial Agent				
			¶%I	%R	95%	CI
	Aminoglycosides	Gentamicin	0.0	1.2	0.2	3.5
		Streptomycin	N/A	18.7	14.1	24.1
		Amoxicillin-clavulanic acid	11.6	5.2	2.8	8.7
	inhibitor combinations					
	Cephems	Ceftiofur	0.4	4.0	1.9	7.2
I		Ceftriaxone	0.0	4.0	1.9	7.2
	Macrolides	Azithromycin	N/A	0.0	0.0	1.5
	Penicillins	Ampicillin	0.0	21.1	16.2	26.7
	Quinolones	Ciprofloxacin	2.8	0.8	0.1	2.8

Fig. 10. Information obtained from the table of Fig. 9 by splitting values in its last column. The resulting table can be used for downstream data analysis.

the next column in the source-table headers to be split up (e.g., the last column in Fig. 9).

Specifically, for each column I_j in the source-table header I with n header rows, Algorithm 2 first computes the string similarity $c_{ij} = \texttt{stringsim}(I_{ij}, T_{ij})$ between each source-table cell value I_{ij} and the corresponding target-table cell value T_{ij}. It then computes the *match rate* M_j between I_j and the column T_j in the target-table header, using the averaged string similarities between the cell values:

$$M_j = \frac{\sum_i c_{ij}}{n}. \tag{3}$$

If the match rate of any column is below a threshold prespecified by analysts-in-the-loop, the algorithm returns the index j of the column, as an indication that the column should be split up. Analysts-in-the-loop can then use functions such as `cSplit()` in R [28] to split up the identified column in the entire source table (rather than just in the header), based on the delimiters used in the source table. Analysts-in-the-loop can apply this process iteratively on the source table, until all its incorrectly extracted columns are identified and split up.

As an example, Algorithm 2 has been used to split up the last column of the table shown in Fig. 9, resulting in the table of Fig. 10. To compute the string similarities in the example, we used the R function `stringsim()` [26] with the similarity-measure method Damerau-Levenshtein [35].

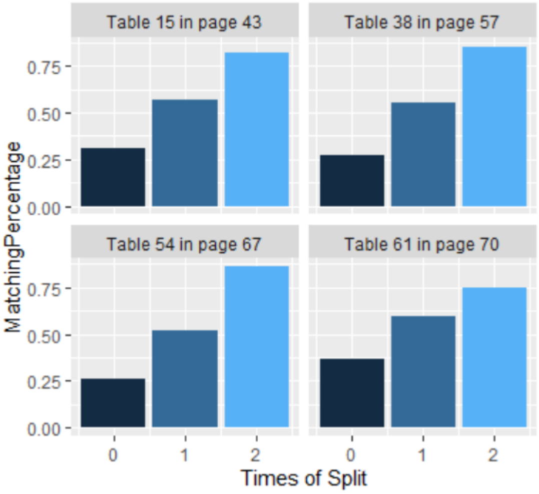


Fig. 11. The matching-percentage improvement resulting from applying Algorithm 2 to tables obtained from summary reports via automatic extraction. The matching percentage shown is for individual tables from the NARMS 2015 report [18], and is measured between (a) the tables published in [18] and (b) the originally extracted tables (time split = 0), as well as the tables obtained via the first iteration/split (1) and the second iteration/split (2). In our experience, the splitting process done according to Algorithm 2 has resulted in tables with 80% matching percentage on average.

Fig. 11 shows the number of iterations (split times) and the improvement of the degree to which the source table values match the values of the ground-truth squashtogram table (from the original summary reports), achieved by splitting up columns with Algorithm 2. The match degree is measured by counting, after each split, the number of matching cell values between the two tables. We use the formal notion of *matching percentage,* defined as follows.

Definition V.1. Let a cell value of the source table I (after column splitting) be I_{ij}, and its corresponding cell value in the ground-truth table C be C_{ij}. Then the *matching percentage* MP_{IC} for I and C is

$$MP_{IC} = \frac{\sum_i \sum_j y_{ij}}{nm} \tag{4}$$

where

$$y_{ij} = \begin{cases} 1, & \text{if } I_{ij} = C_{ij} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

(n, resp. m, is the number of rows, resp. columns, in C.) Note that it goes to 0 when we cannot find corresponding I_{ij} for C_{ij}, due to the different numbers of columns in I and C.

D. Feedback Loop for Phase 3

For the phase of data integration, the feedback loop takes as input the statistics collected on the integrated tables. Analysts-in-the-loop can use the statistics to detect errors made in the data-integration process. For example, suppose we know how many antimicrobial agents should be in the integrated table. Then, in case discrepancies are found between the correct total number of agents and the total number of agents present in the integrated table, the analysts-in-the-loop can activate the feedback loop. As one option, the row-binding function `rbind()` used for the integration could be examined.

E. Feedback Loop for Phase 4

For the phase of visualizing and analyzing the integrated data, analysts-in-the-loop can detect errors made in the data-visualization or analytics steps by looking for any surprising (potentially incorrect) results and by discussing the findings with experts-in-the-loop.

For example, in Fig. 8, the AMR trends (`%R`) calculated for the years 2009–10 based on the raw data for those years that come from laboratory sources, are at odds with the trends coming from the NARMS report for the same years. Moreover, using two-sample proportion testing as described in Section IV-E1, a statistically significant difference (p-value ≈ 0) is found between the percentages of resistance (`%R`) for the two data sources. This issue could be due to the different area coverage in collecting the testing samples. Indeed, the raw data in this example originate from sources in North Carolina, while the NARMS data come from all U.S. states. Still, this issue could also be partly due to errors coming from the data sources. Hence, analysts-in-the-loop should consult experts-in-the-loop for appropriate interpretations.

F. Data Provenance in Feedback Loops

In our experience with the proposed workflow, we have identified opportunities for utilizing data-provenance techniques in further enriching the feedback loops built into the workflow. Indeed, data provenance can help analysts-in-the-loop identify more precisely the origins of the potentially erroneous outcomes of phases of the forward flow of the workflow. As a result, the workflow outcomes can be made more accurate, and the workflow process more efficient. Existing data-provenance approaches, see, e.g., [36], [37], often leverage annotations in the input data to help in retracing the origins of the problems in the outputs, including the case of data aggregation. We are currently working on extending the feedback loops in the proposed workflow to include data-provenance approaches.

VI. Challenges and Lessons Learned

In this section we discuss the challenges that we faced in our workflow building and testing experience, as well as the lessons learned in the process.

A. Leveraging Humans-In-the-Loop in Analytics Workflows

Our proposed workflow involves collaborative work by humans-in-the-loop on preparing and processing data. The human-effort costs thus incurred in data-analytic solutions are nontrivial and nonnegligible. It might appear that such human involvement should be ultimately eliminated. Perhaps surprisingly, in our workflow building and testing experience we found that such collaborations improve the quality of the data-integration and analytics outcomes, and thus help bring about better data-analytics solutions than in the fully automated cases. The reason is, work done by humans and by machines is complementary instead of duplicative. In addition, collaborations between humans-in-the-loop further solidify the results obtained in the integration and analytics pipeline.

B. Collaborations on Multidisciplinary Teams

Collaborations on multidisciplinary teams introduce the challenge of knowledge and experience translation between experts with different backgrounds. At the same time, the results obtained with the help of our workflow show that ultimately collaborating on multidisciplinary teams is a rewarding experience, which gives all the participants better insights and is likely to markedly improve final workflow outcomes.

C. Communicating between Experts via Visualization

From our experience in this project, using data visualization for communications between analysts-in-the-loop and experts-in-the-loop is superior to using structured (e.g., relational) aggregation or integration results for the same purpose. This way, collaborations between the humans-in-the-loop on such interdisciplinary projects can be made more productive.

D. Introducing NLP into Data-Analytics Workflows

In the proposed workflow, analysts-in-the-loop semi-automatically extract tables from summary reports. At the same time, any natural-language captions or notes that accompany the tables in the source files have to be extracted manually. There is thus an opportunity for introducing natural language processing (NLP) techniques such as those of [38] into this and similar workflows, with the aim of automatically extracting both tables and their associated natural text.

E. Developing Extensible Workflows

We observe that data-analytics workflows in different application domains, as discussed in the literature, can overlap to large degrees both with each other and with the workflow introduced in this paper. If a workflow is not extensible, (partially) reusing it for analyzing data in a new domain can be challenging. We posit that in building workflows for a domain, it is worth focusing on making them more general and easier to be migrated to other domains in the future. This can be accomplished, for example, by setting up general extensible workflow pipelines for interacting with humans-in-the-loop, by developing general methodologies for modeling human knowledge for specific domains, and by coordinating standalone data-analytics tools, see, e.g., [19], [39].

VII. Conclusion

In this paper we reported on our experience of developing a scalable collaborative end-to-end data-integration and data-analytics workflow for antimicrobial-resistance (AMR) research. We also reported on the experiences, challenges, and lessons learned in using this workflow to integrate and analyze data from the AMR domain. Our workflow contains multiple phases that can process data in a bidirectional manner, i.e., via the forward flow and feedback loops, and allows for interventions by analysts-in-the-loop and experts-in-the-loop, with the potential of improving the quality and usability of the workflow outcomes. Collaborations between analysts-in-the-loop and experts-in-the-loop in data integration and analytics in the workflow are supported and, more importantly, encouraged. Specifically, we found that incorporating domain knowledge into the data-analytics process can aid in more efficient discovery and mitigation of potential errors, and thus in improved reliability of the final analytics outcomes. With the help of the incorporated open-source tools and of the new algorithms developed by the authors, the proposed workflow can help humans-in-the-loop alleviate the time and labor costs of performing data analytics. We posit that the workflow can be a promising solution for analyzing data for the AMR domain, as well as potentially for other domains that have similar data characteristics and data-processing requirements.

References

[1] A. Doan, A. Y. Halevy, and Z. G. Ives, *Principles of Data Integration*. Morgan Kaufmann, 2012.

[2] U. Dayal, M. Castellanos, A. Simitsis, and K. Wilkinson, "Data integration flows for business intelligence," in *EDBT*, 2009.

[3] M. Lenzerini, "Data integration: A theoretical perspective," in *PODS*, 2002.

[4] W. Inmon, *Building the data warehouse, 2nd ed.* New York: John Wiley & Sons, 1996.

[5] R. Kimbal, L. Reeves, M. Ross, and W. Thornthwaite, *The data warehouse lifecycle toolkit: Expert methods for designing, developing, and deploying data warehouses*. New York: John Wiley & Sons, 1998.

[6] F. Prestinaci, P. Pezzotti, and A. Pantosti, "Antimicrobial resistance: A global multifaceted phenomenon," *Pathogens and global health*, 2015.

[7] B. E. Karp, H. Tate, J. R. Plumblee, U. Dessai, J. M. Whichard, E. L. Thacker *et al.*, "National Antimicrobial Resistance Monitoring System: Two decades of advancing public health through integrated surveillance of antimicrobial resistance," *Foodborne pathogens and disease*, 2017.

[8] E. P. Lesho, P. E. Waterman, U. Chukwuma, K. McAuliffe, C. Neumann, M. D. Julius *et al.*, "The antimicrobial resistance monitoring and research (ARMoR) program: the US Department of Defense response to escalating antimicrobial resistance," *Clinical Infectious Diseases*, 2014.

[9] B. Borck Hg, F. Bager, H. B. Korsgaard, J. Ellis-Iversen, K. Pedersen, L. B. Jensen *et al.*, "DANMAP 2017 - use of antimicrobial agents and occurrence of antimicrobial resistance in bacteria from food animals, food and humans in Denmark," Copenhagen: Statens Serum Institut, National Veterinary Institute, Technical University of Denmark National Food Institute, Technical University of Denmark, Tech. Rep., 2018, https://danmap.org.

[10] National Academies of Sciences - Engineering - Medicine *et al.*, *Combating Antimicrobial Resistance: A One Health Approach to a Global Threat: Proceedings of a Workshop*. National Academies Press, 2017.

[11] R. Barrell, "Isolations of Salmonellas from human, food and environmental sources in the Manchester area: 1976–1980," *Epidemiology & Infection*, 1982.

[12] A. Douris, P. J. Fedorka-Cray, and C. R. Jackson, "Characterization of Salmonella enterica serovar agona slaughter isolates from the animal arm of the National Antimicrobial Resistance Monitoring SystemEnteric Bacteria (NARMS): 1997 through 2003," *Microbial Drug Resistance*, 2008.

[13] S. Hong, A. Rovira, P. Davies, C. Ahlstrom, P. Muellner, A. Rendahl *et al.*, "Serotypes and antimicrobial resistance in Salmonella enterica recovered from clinical samples from cattle and swine in Minnesota, 2006 to 2015," *PloS one*, 2016.

[14] S. Keelara, H. M. Scott, W. M. Morrow, W. A. Gebreyes, M. Correa, R. Nayak *et al.*, "Longitudinal study of distributions of similar antimicrobial-resistant Salmonella serovars in pigs and their environment in two distinct swine production systems," *Appl. Environ. Microbiol.*, 2013.

[15] J. Lai, C. Wu, C. Wu, J. Qi, Y. Wang, H. Wang *et al.*, "Serotype distribution and antibiotic resistance of Salmonella in food-producing animals in Shandong province of China, 2009 and 2012," *International journal of food microbiology*, 2014.

[16] Y. Wang, C. Cao, W. Q. Alali, S. Cui, F. Li, J. Zhu *et al.*, "Distribution and antimicrobial susceptibility of foodborne Salmonella serovars in eight provinces in China from 2007 to 2012 (except 2009)," *Foodborne pathogens and disease*, 2017.

[17] "U.S. Food and Drug Administration," https://www.fda.gov.

[18] CDC, "National Antimicrobial Resistance Monitoring System for Enteric Bacteria (NARMS): Human Isolates Surveillance Report for 2015," Atlanta, Georgia: CDC, Tech. Rep., 2018.

[19] R. Lourenço, J. Freire, and D. Shasha, "Debugging machine learning pipelines," in *DEEM*, 2019.

[20] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2019, https://www.R-project.org/.

[21] "SQLite," 2019, https://www.sqlite.org/index.html.

[22] Manuel Aristarn, Mike Tigas and Jeremy B. Merrill, "Tabula," 2018, https://tabula.technology/.

[23] S. Dorai-Raj, *binom: Binomial Confidence Intervals For Several Parameterizations*, 2014, R package version 1.1-1, https://CRAN.R-project.org/package=binom.

[24] H. Wickham and L. Henry, "Tidyr: Easily tidy data withspread ()andgather ()functions," *R package version 0.6*, 2017.

[25] G. Grolemund and H. Wickham, "Dates and times made easy with lubridate," *Journal of Statistical Software*, 2011.

[26] M. van der Loo, "The stringdist package for approximate string matching," *The R Journal*, 2014.

[27] H. Wickham, "stringr: Simple, consistent wrappers for common string operations," *R package version*, 2017.

[28] A. Mahto, *splitstackshape: Stack and Reshape Datasets After Splitting Concatenated Values*, 2019, R package version 1.4.8, https://CRAN.R-project.org/package=splitstackshape.

[29] S. Garnier, *viridis: Default Color Maps from 'matplotlib'*, 2018, R package version 0.5.1, https://CRAN.R-project.org/package=viridis.

[30] C. Ginestet, "ggplot2: Elegant graphics for data analysis," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2011.

[31] B. Rudis, *hrbrthemes: Additional Themes, Theme Components and Utilities for 'ggplot2'*, 2019, R package version 0.6.0, https://CRAN.R-project.org/package=hrbrthemes.

[32] K. Muller, H. Wickham, D. A. James, and S. Falcon, *RSQLite: 'SQLite' Interface for R*, 2018, R package version 2.1.1, https://CRAN.R-project.org/package=RSQLite.

[33] R. Becker, J. Chambers, and A. Wilks, *The New S Language.* Chapman and Hall/CRC, 1988.

[34] O. Miettinen and M. Nurminen, "Comparative analysis of two rates," *Statistics in Medicine*, 1985.

[35] G. V. Bard, "Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric," in *ACSW*, 2007.

[36] Y. Amsterdamer, D. Deutch, and V. Tannen, "Provenance for aggregate queries," in *PODS*, 2011.

[37] T. J. Green, G. Karvounarakis, Z. G. Ives, and V. Tannen, "Update exchange with mappings and provenance," in *VLDB*, 2007.

[38] S. Wu, L. Hsiao, X. Cheng, B. Hancock, T. Rekatsinas, P. Levis, and C. Ré, "Fonduer: Knowledge base construction from richly formatted data," in *SIGMOD*, 2018.

[39] J. Freire, D. Koop, F. Chirigati, and C. T. Silva, "Reproducibility using vistrails," *Implementing Reproducible Research*, 2014.