Visualizing Infection Surveillance Data for Policymaking Using Open Source Dashboarding

Monika Maya Wahi[1,2]   Natasha Dukach[2,3]

[1] General Education, Laboure College, Milton, Massachusetts, United States

[2] DethWench Professional Services, Boston, Massachusetts, United States

[3] Biotechnology Program, Northeastern University, Boston, Massachusetts, United States

Address for correspondence Monika Maya Wahi, MPH, CPH, 30 Newbury Street, 3rd Floor, Boston, MA 02116, United States (e-mail: dethwench@gmail.com).

Abstract

Background   Health care-associated infections, specifically catheter-associated urinary tract infections (CAUTIs), can cause significant mortality and morbidity. However, the process of collecting CAUTI surveillance data, storing it, and visualizing the data to inform health policy has been fraught with challenges.

Objectives   No standard has been developed, so the objective of this article is to present a prototype solution for dashboarding public health surveillance data based on a real-life use-case for the purposes of enhancing clinical and policy-level decision-making.

Methods   The solution was developed in open source software R, which allows for the creation of dashboard applications using the integrated development environment developed for R called RStudio, and a package for R called Rshiny. How the surveillance system was designed, why R was chosen, how the dashboard was developed, and how the dashboard features were programmed and function will be described.

Results   The prototype dashboard includes multiple tabs for visualizing data, and allows the user to interact with the data by setting dynamic filters. Controls were used to facilitate the interaction between the user and application. Rshiny is reactive, in that when the user (e.g., clinician or policymaker) changes the parameters on the data, the application automatically updates the visualization as well as parameters available based on current filters.

Conclusion   The prototype dashboard has the potential to enhance clinical and policy-level decision-making because it facilitates interaction with the data that provides useful visualizations to provide such guidance.

Keywords
► system improvement
► dashboard
► user acceptance and resistance
► organizational change management

Background and Significance

Health care-associated infections (HCAIs), also termed nosocomial infections, are infections in patients that are caused by the health care environment which can cause significant mortality and morbidity.[1–4] HCAIs are caused by the use of invasive devices such as catheters and ventilators.[1] One type of HCAI, catheter-associated urinary tract infection (CAUTI), has been identified as a priority for prevention, as the rate of urinary catheterization has been found to be between 12 and 19% in hospitals in different countries,[5–7] and that in the United States, the risk of acquiring CAUTI increases by 3 to 7% per catheter day.[6]

For these reasons, the World Health Organization (WHO), the Gulf Cooperation Council, and agencies in individual

countries such as the Centers for Disease Control and Prevention (CDC) in the United States, have published guidance for the prevention, treatment, and surveillance of CAUTI in health care settings.[6,8,9] However, the process of collecting CAUTI surveillance data, storing it, and leveraging the data to inform health policy has been fraught with challenges. In the United States, the National Health Safety Network (NHSN) was designed as a tracking system for CAUTI and other HCAI that health care settings can opt into to use.[10]

While the NHSN presents one solution, the system has been shown to have many problems. The NHSN requires use of standardized forms for data collection procedures, and because these forms are used across the very diversified U.S. health care system, they become prohibitive for individual centers to complete.[10,11] Also, NHSN results are of questionable use for evidence-based policy; NHSN rates of CAUTI are extremely sensitive to case definition features,[12] miss many important cases differentially by location,[13] and suffer from unacceptably low interrater reliability,[14] to name a few issues. Further, the NHSN acknowledged in 2013 that there were issues with HCAI case definitions that affected surveillance estimates, so it undertook a process of revision.[15]

While the NHSN policies and procedures might represent the most widely used system for CAUTI surveillance globally, as many countries outside the United States follow NHSN's lead,[16–18] it is an incomplete solution. Unfortunately, these challenges mean that the work of determining optimal reporting and visualization of CAUTI data to guide health care policy has not even started.[19] The authors could find no recommendations for optimal visual presentation of such data, even from NHSN. It was not possible to consider improving upon the NHSN system due to the many technical issues arising from having one centralized system for surveillance of many different types of infections.

This project represents a prototype based on a use-case for a real organization that will be referred to from now on as Organization A. The dashboard solution was custom-developed for a particular health care corporation and involved subject matter experts from this organization. For the purposes of protecting privacy, a fictitious scenario based on the real case was created. Since Organization A's reach covers a wide range of geographic territory and represents an integrated delivery system of health care with a multitude of both general and specialty hospital settings as well as outpatient clinics, to function as a useful prototype for this use-case, intensive care units and other units that use urinary catheters were selected to be included in the dashboard from four locations: two general hospitals, a specialty heart hospital, and a specialty cancer center. The dashboard prototype was developed as part of an upgrade of the CAUTI surveillance system that was conducted at Organization A.

Objectives

The objective of this article is to present a prototype solution for dashboarding public health surveillance data based on a real use-case for the purposes of enhancing clinical and policy-level decision-making.

Methods

This prototype is based on a real use-case, but for the purposes of protecting privacy, a fictitious scenario based on the real case was created, and will be described below.

Setting

In the fictional scenario, a town in the Midwestern United States named Holiday is created to mimic a few of the real settings in Organization A. Organization A's two general hospitals were recreated into the prototype as Holiday General Hospital (HGH) and smaller Anderson-Walker Hospital (AWK). The Holiday Heart Hospital (HH) and Holiday Cancer Center (CC) were included as the specialty settings, respectively. The structure of the data for this prototype was based on the structure of the data developed for Organization A, but the data used in this project were fabricated.
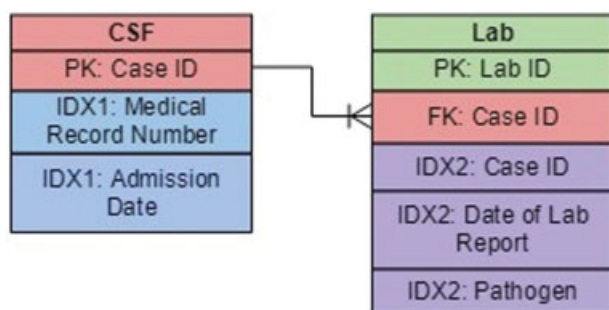
Surveillance System Upgrade

Originally, the system at Organization A for CAUTI was a manual process designed to track data about each CAUTI case. The director of the infectious disease department (DI) led educational interventions so her team was apprised of the NHSN case definition of CAUTI.[6] When a potential case of CAUTI was identified, the DI would be notified, and this was followed by a laboratory-confirmed diagnosis and treatment for the patient. A paper form would be completed and stored documenting case data along with laboratory reports.

However, it was found that the data collected were not standardized in any way. They did not adhere to standards set by regulatory bodies (e.g., the CDC), nor did they follow internally set standards. A review of the literature revealed that risk factors for CAUTI vary by country and risk factor groups (e.g., older age groups, immunocompromised)[7,20,21]; and that there were no global data standards for data entry, storage, and analysis of laboratory results. For noncommunicable diseases, the WHO recommended using a STEPwise approach to developing unique surveillance systems customized for conditions and risk factors in a particular population.[22] The STEPwise model was used in redesigning the surveillance system.

Data Collection

In the new system, a risk factor form was created called a case surveillance form (CSF) and was developed in Microsoft Word. The DI worked with one author (M.M.W.) to develop the questions on the CSF, which gathers the following risk factor information: if patient was a transfer, if they were receiving antibiotics within 90 days of admission, if they had at least one chronic disease upon admission, if they were bedridden, if they were incontinent, if they were immunocompromised, if they had an infection at any site at the time of admission, if they had undergone urinary surgery within 7 days of admission, and if they were a neonate. Risk factors were selected based on priorities set by the DI and local policy.[22]

For this demonstration project, one author (M.M.W.) generated data in the same structure as the data from the CSF (called CSF data), with one case per row. One author (M.M.W.) also

Fig. 1 Entity-relationship diagram. This shows how the two tables described in ►Table 1 are related in the surveillance system. FK, foreign key; IDX, index; PK, primary key.

determined the data structure for the laboratory data (called Laboratory data), which was to list one organism per row, plus all its attributes (drug sensitivities and resistances). Each case is assigned a confidential CAUTI identity (ID), and this identifier is included in each row of the CSF data. For the Laboratory data, the CAUTI ID is present on each row, as well as the date of the report, and the organism on the report, creating a triple index to make the row unique. This does not demonstrate a real-time data processing system, but instead a warehousing approach on which a dashboard could be placed. This is because CAUTI is so rare that real-time processing is not necessary.

Data Structure

►Table 1 summarizes the data structure behind the CAUTI surveillance system at Organization A, and ►Fig. 1 shows the entity-relationship diagram to depict how the two tables are related, and the composite index on each table that makes each row unique.

As is seen in ►Table 1 and ►Fig. 1, two data tables are used in the surveillance system, CSF and Laboratory, and each has their own primary key. They are related through the primary key for CSF, Case ID, being present in the Laboratory table in each record as a foreign key.

Choices for Data Visualization and Dashboard Development

After the upgrade, one author (M.M.W.) wrote a report for Organization A that included data visualizations. This expe-

rience prompted the choices that were made for data visualization used in the dashboard implementation. In contrast to the static report, the dashboard allows the user to interact in real-time to visualize data in the dashboard.

Choice of Rshiny Dashboard Solution

SAS and SPSS are statistical software that have traditionally been used for health data analytics, but they do not have advanced visualization capabilities and are commercial products.[23] Making a dashboard using the Web visualization approach of Python programming supporting a JAVA display requires knowledge of both Python and JAVA, and therefore is difficult to maintain. Using a visualization package like Tableau is limiting, in that it does not have its own programming language. Tableau, however, does have some programming capabilities, with functions like those in Microsoft Excel, and also interfaces with R to make it more extensible.[24] Also, considerations must be made as to how the final dashboard can interface with the health care organizations' internal data systems when selecting software at the design stage.[25]

R is an open source software that allows for programmers in the community to contribute "packages" as optional add-ons to R.[23] Although there are many ways to program R, R developers created two popular programmer interfaces: R GUI and RStudio.[23] R GUI simply provides a programming "console" window where code is run, and scripts windows where code can be programmed and saved. If code run in R GUI generates a plot, it opens in a separate window. In contrast, RStudio is an integrated development environment that allows for advanced visualization capabilities that can seamlessly interface with the World Wide Web, and is necessary to use for dashboard development.[26] In RStudio, Web applications can be developed, where running code outputs graphics to a Web browser. Because of this structure, the code is easy to maintain, and because it is open source, it is easy to interface R with most data systems.

A package, Rshiny,[27] has been developed specifically for dashboarding visualizations; Rshiny can be leveraged with a constellation of other visualization packages that can be added to R to achieve various visual and data management goals. R script files, which are basic text files saved with the .R extension, allow users to deploy an Rshiny application to a

Table 1 Data structure behind the CAUTI surveillance system analytic data

Table	Domain	Data collection	Row entity (Indexes)

CSF	Administrative	Data transcribed from medical record	Primary key: Case ID assigned by DI from roster
			Composite index: None, but theoretically:
			    patient medical record plus
			    admit date

	Demographics – Clinical		
	Risk factors for CAUTI		

Laboratory	Pathogen	Transcribed from laboratory report and classified to standard abbreviations	Primary key: Laboratory ID (automatically generated)
			Composite index:
			    Case ID
			    date of laboratory report
			    pathogen
			Foreign key to CSF:
			Case ID

	Drugs and resistance		
	Drugs and sensitivity		

Abbreviations: CAUTI, catheter-associated urinary tract infection; CSF, case surveillance form; DI, director of the infectious disease; ID, identity.

local or remote server. The application code can incorporate both back- and front-end components in one long program, or can be modularized into independent code segments.[28] Modularization provides flexibility, because modules of code can be selectively called by the front-end code through available packages and functions.[28]

Ultimately, R's flexibility that allows programmers to build visualizations on data coupled with its data management and analysis capabilities were the main reasons why R, and specifically Rshiny, was chosen for this prototype solution. Another important reason why R was chosen is that it is open source, which allows for the following features: (1) no cost to obtain the software, and (2) the potential for sharing the solution across platforms to promote adoption. The packages shiny and shiny-dashboard were used to achieve all the functionality described in this prototype.[27,29] The plotting package called ggplot2[30] was used for the bar charts, and package UpSetR was used for the other plots.[31] A resource online called Shiny from RStudio provides a gallery of dashboards contributed by community users, along with documentation.[32] RStudio makes a public Web page available called ShinyApps that provides a space for R developers to share their Rshiny dashboard solutions.[33] The dashboard solution that is the subject of this article is published there, and the source code is available on GitHub (https://github.com/NatashaDukach/CAUTI_dashboard/blob/master/app.R).[34]

Dashboard Structure

The dashboard was divided into three topics: visualizing risk factors specific for a physical location and date; visualizing rates of detected microorganisms and patterns of resistance and sensitivity of the organisms to drugs at the specific location and time; and visual comparison of selected plots. The user sets parameters using controls on the dashboard, and the dashboard reacts by updating the display. ►Fig. 2 represents the default dashboard display (after dismissing splash screen).

This dashboard has several structural components: tabs, controls, and charts. Each tab is a page which consists of three basic elements: header, sidebar, and body. The sidebar is the vertical panel on the left side of the dashboard that can be used to navigate to the tabs, and the body portion of the tab includes all controls and graphical visualization. Per ►Fig. 2, Rshiny default provides three areas of the screen: the top band, the side bar, and the middle section. The programmer can choose to place tabs on the side bar to allow the tab navigation. Controls capture user input; ►Fig. 2 shows controls both above and to the right side of the plot.

In addition to being the default first tab of the dashboard, ►Fig. 2 displays a bar plot of risk factor prevalence from the CSF data. Across the top of ►Fig. 2, there are two controls in the body that allow the user to set parameters: a location dropdown, and a date range selector. In the upper right on the header, there are three buttons; the "about" button describes the project and
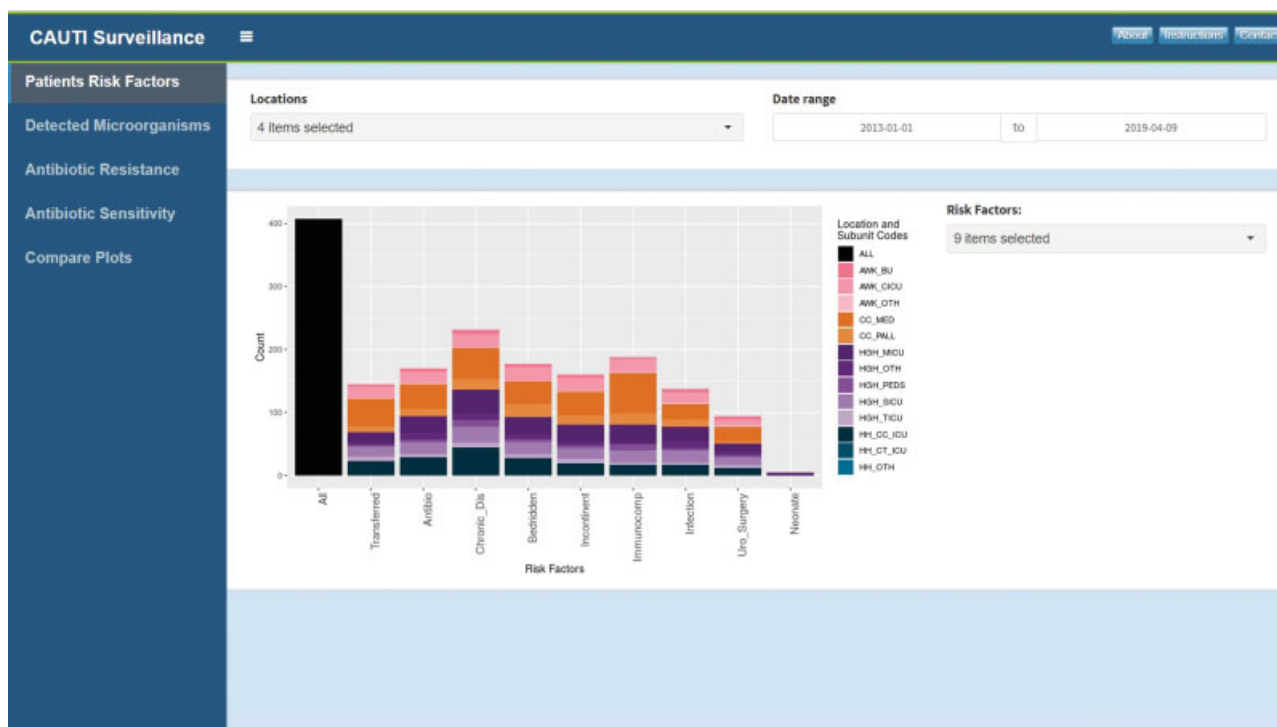
Fig. 2 Dashboard structure. This is the default display of the dashboard after dismissing the splash screen. In terms of the dashboard structure, buttons in the upper right can be clicked to display "about," "instructions," and "contact" information. Across the top are controls that set the parameters for location and time range for data display. The button with three parallel lines on the upper left can be clicked to autohide the menu down the left side. In the center of the default display is the plot visualizing risk factors from the case surveillance form (CSF). On the right, location and subunit codes are used to indicate specific subunits within locations. Codes use the logic of location abbreviation, followed by an underscore, followed by a subunit abbreviation (e.g., HMC_PEDS). Across the bottom, left to right, are the risk factors that were selected for inclusion in the surveillance system: All cases, having received antibiotics within 90 days of admission, having a chronic disease, bedridden at admission, incontinent at admission, immunocompromised at admission, already colonized with infection at any site upon admission, having had urological surgery within 7 days of admission, and being a neonate.

repeats information that displays on the initial splash screen, the "instructions" button opens a splash page that provides directions about how to use the dashboard, and the "contact" button opens a splash page that displays the information about how to contact the authors. In the upper left, the name of the dashboard is present, and to the right of this are three horizontal lines. This is a button that, when clicked, will expand the middle white part of the screen while autohiding the blue, vertical menu on the left. Clicking the same button again will toggle back to the default display in ►Fig. 2.

Dashboard Features

In Rshiny dashboards, the application programming code can be roughly classified into two categories: back-end programming and front-end programming. Raw data are typically imported into R using back-end scripts for the purposes of achieving data manipulation and staging data for display. Hence, back-end files are usually snippets of R code written for a specific data management task. On the contrary, the front-end programming includes code that manipulates the display. Back-end and front-end files are stored in the same application directory and must be present together for the Rshiny application to function.

Four of the five tabs feature a reactive plot with user-manipulated controls, and the last tab features two plots side by side, with a set of controls determining which plots are displayed. The second tab, "Detected Microorganisms," displays an upset plot showing the pattern of organisms in cases in the system. The third tab, "Antibiotic Resistance," and the fourth tab, "Antibiotic Sensitivity," display upset plots showing patterns of drug resistance and sensitivity in cases, respectively. Finally, the "Compare Plots" tab allows the user to select two of the aforementioned plots so they can be displayed and compared side by side.

Use of Shiny Widgets for Controls

The package shinyWidgets can be installed on top of shiny and shinydashboard to provide a collection of responsive input controls that users can manipulate to select one or several items listed in a dropdown menu.[35] In this dashboard, the two types of controls used from the shinyWidgets package were the dropdown picker and the date range picker. For the dropdown picker, the pickerInput function controls the available choices in a dropdown based on input from other controls.

The second type of control used from the shinyWidgets package was the dateRangeInput function, which allows the user specify a time range to filter the results reported between two dates (see ►Fig. 2). This date filter restricts the data being plotted by only including cases with a date of admission in the date range. The values in the location and date range controls are applied to the visualization on all tabs, and if these values change, visualizations on all tabs are updated.

Use of Shiny Server for Reactive Output

To enable the shinyWidgets to function with real-time reactivity, the shiny application must be set up with code in a specific format in order for the application to run.[36] Through the user making selections using the controls, input values change, and

the new information which is passed to the server-side of the code. To achieve this, the dashboard application scans every few milliseconds to evaluate the current value of every object that is being used during the session, and if it detects any change in object value, R reevaluates and displays a new output. To facilitate redisplay after parameters are changed by the user through controls, server code must include the reactive function expression input$x, where x is a blank placeholder which will be populated by the user-selected value after the user chooses the value through the controls. As R continues to search for updated values in controls, it will repopulate x if it finds that a change has been made, and this will cascade to update the display.[37]

Reactive functions used in this dashboard were: observeEvent, updatePickerInput, and renderPlot. It is the observeEvent function that essentially "listens" for changes in parameters through the user manipulating controls. The observeEvent function should be set on every control that can be dynamically updated and should therefore reactively update the display. For the dynamic dropdowns in this dashboard, if the observeEvent function detects the event of a change of location or date range in the header, it launches the update-PickerInput function which then updates the dropdowns dependent upon these parameters. For example, if the location selected were changed from AWK to HGH, the observeEvent function would detect this, and then the updatePickerInput function would change the risk factor dropdown choices (as shown in ►Fig. 2) to offer only those available for location HGH.

►Fig. 3 provides an example of where reactive output was facilitated by the observeEvent function. ►Fig. 3 shows an example of the upset plot for detected microorganisms. Note that the top controls indicate the locations and time frame selected. For the data that meet these criteria, only eight microorganisms were available for plotting. Were the user to change the selection of the plotted microorganisms, or change the selection of the header controls, the observeEvent function would notice the changes, and re-render the plot with renderPlot. If location changed, this would change the microorganisms in the picklist, so the updatePickerinput function would arrange for this.

## Results

►Fig. 4 shows an example of one use of the dashboard for decision-making.

As shown in ►Fig. 4, the compare plots tab is used to compare prevalence of risk factors for CAUTI with the pattern of antibiotic resistance shown at the location CC. What is immediately evident is that CC has a high prevalence of CAUTI patients that are immunocompromised and have chronic disease (bar plot), and also has a serious problem with organisms with multiple resistances (upset plot).

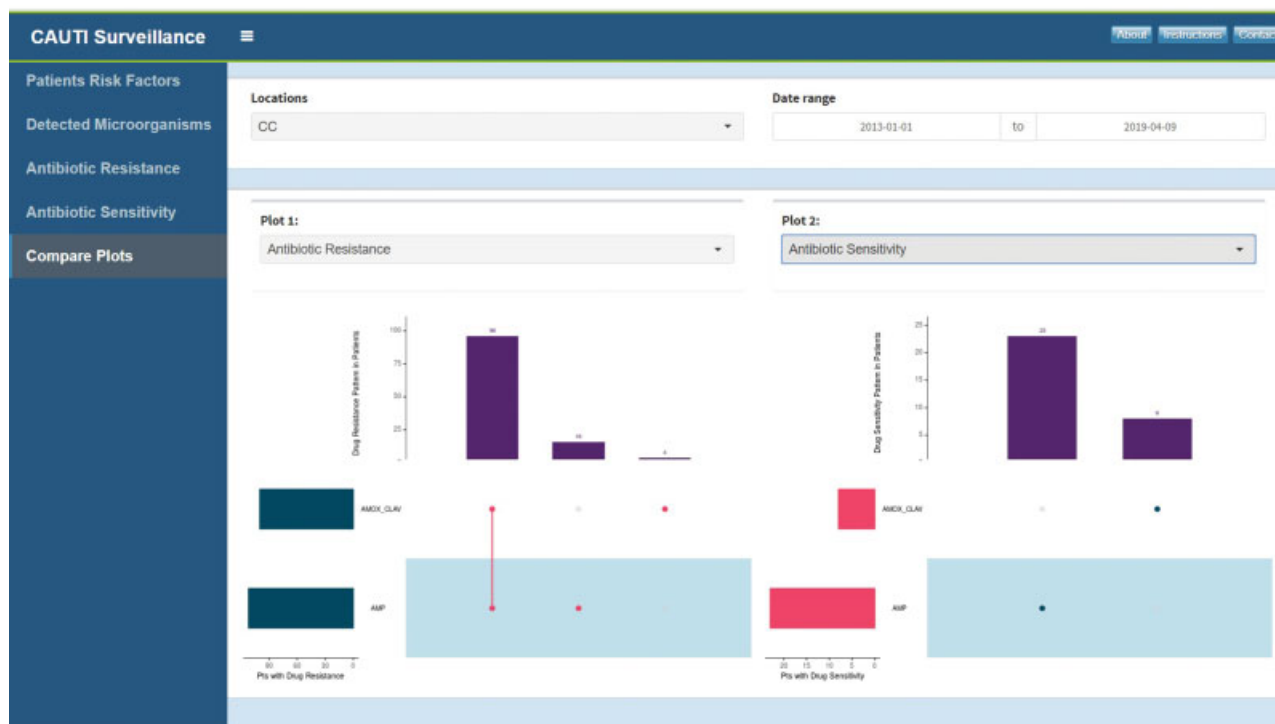►Fig. 5 shows another way that data can be visualized in the dashboard to facilitate decision-making.

►Fig. 5 shows the compare plots tab being used to compare antibiotic resistance and sensitivity patterns filtered for selected drugs. This allows for policymakers to troubleshoot antibiotic policy and make decisions about potentially quarantining or reserving drugs for future use
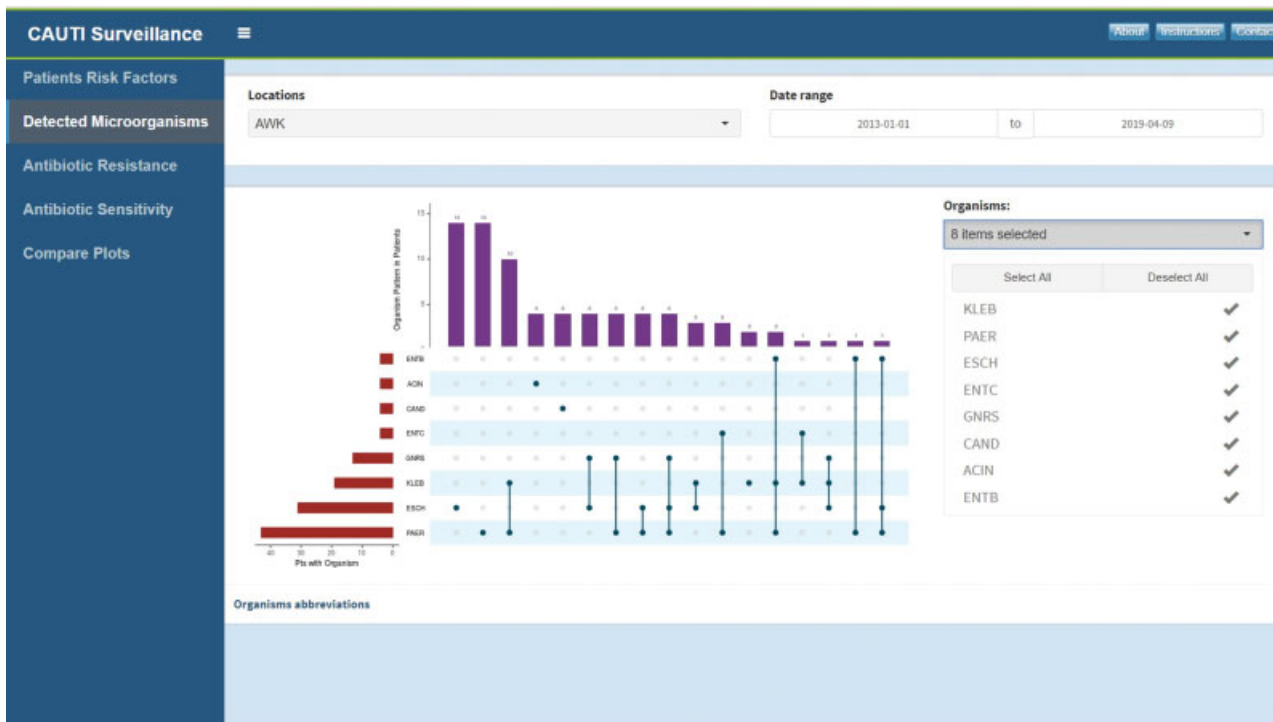
Fig. 3 Example of upset plot for detected microorganisms. The header controls indicate the locations and time frame selected. On the right, abbreviations for the microorganisms available for plotting for the locations and time frame selected appear in a dropdown. For the data that meet these criteria, only eight microorganisms were available for plotting. These same abbreviations are used on the y-axis of the upset plot.
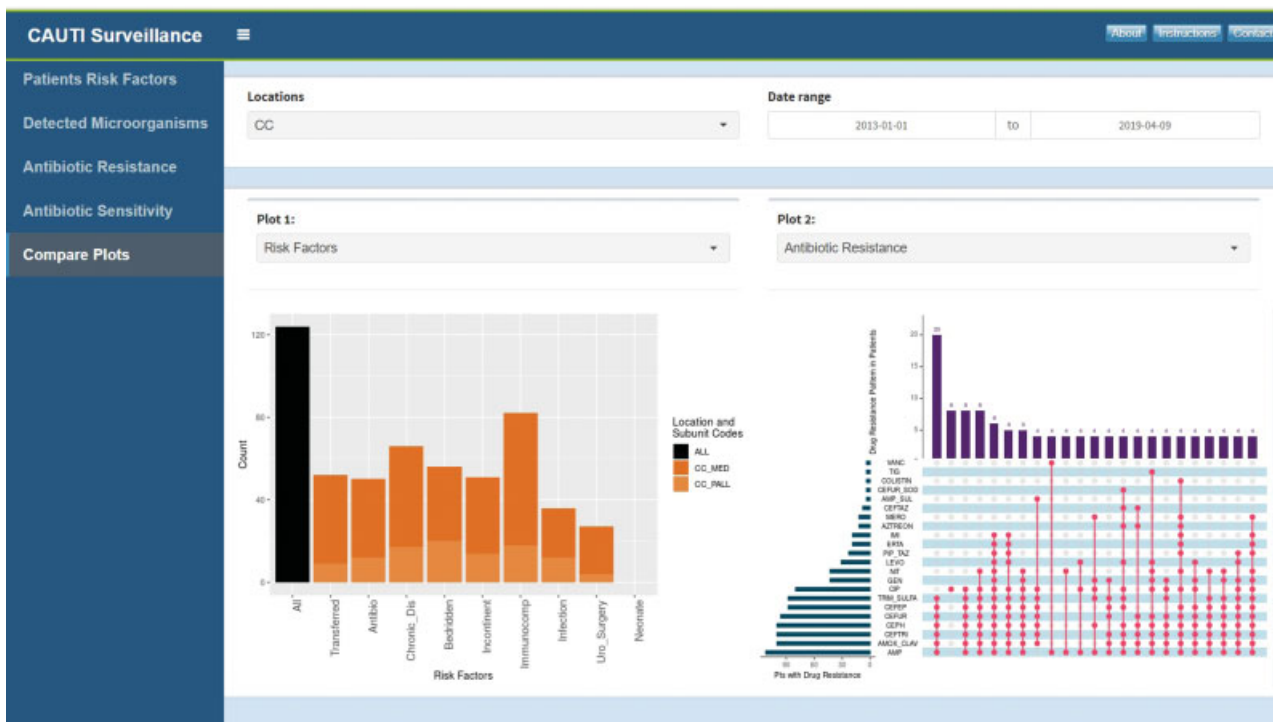
Fig. 4 Visualizing risk factors and antibiotic resistance patterns at one location. After comparing these plots at the Holiday Cancer Center (CC) location, it is evident that there is a high prevalence of catheter-associated urinary tract infection (CAUTI) patients that are immunocompromised and have chronic disease (bar plot) at this location, and also, CC has a serious problem with organisms with multiple resistances (upset plot). Notice the location and risk factor abbreviations from ►**Fig. 2** in the risk factor plot's x-axis. In the upset plot, drug resistances are visualized. A standard list of abbreviations for drugs in the surveillance system was established, and is shown on the y-axis of the upset plot.

Fig. 5 Comparing resistance and sensitivity patterns. This shows the compare plots tab being used to compare antibiotic resistance and sensitivity patterns filtered for selected drugs, allowing for policymakers to troubleshoot antibiotic policy and make decisions about the use, quarantine, or discontinuation of usage of certain drugs. The drug abbreviations used are standard in the surveillance system.

that remain sensitive, and discontinuing use of drugs for which microorganisms appear universally resistant.

Discussion

We were successful in developing a prototype solution for dashboarding CAUTI data to facilitate decision-making at the clinical and policy level in health care. Although our dashboard measured risk factors and other parameters important for the use-case, other organizations and settings may choose different parameters depending upon their focus. The use of R allowed us to publish our dashboard and make our prototype available for adaption to other systems. The use of Rshiny allows for reactive update and useful visualizations that provide information that cannot be easily gleaned from a static report. Further, the system developed represents a "best practices" system, of which the dashboard is an integral part. Future decision-making and changes in antibiotic policy can be reflected in dashboard functioning by keeping it updated with appropriate queries and visualizations.

This use-case presents a prototype for a dashboard to be used with a manual CAUTI surveillance system. For Organization A, the next steps involved upgrading all of their HCAI surveillance systems, including central line-associated blood stream infection, surgical site infection,[38] and ventilator-associated pneumonia. Each of these will require their own data and dashboard structures, but these will be similar to the ones for CAUTI. Once these systems are upgraded and dashboards made, baseline rates will be available. At that time, policymakers can set targets for

benchmarks, and the dashboards can be used for monitoring these targets.

Conclusion

In conclusion, Rshiny provided an optimal platform for a dashboarding solution for an upgraded HCAI surveillance program in a health care system. It also allowed for the dashboard to be shared so others may benefit from the development. As public health surveillance systems are upgraded, new dashboarding implementation should be proposed. Choosing Rshiny as a go-to tool for development will make it easier to share technological advances in visualizing public health surveillance data.

Clinical Relevance Statement

This dashboard prototype implementation aims to assist health care leaders in making decisions regarding CAUTI surveillance. It helps them leverage data they are likely already gathering as part of quality control systems by using open source visual analytics. Through use of this dashboard, health care leaders can establish baseline rates, set targets, and monitor benchmarks for HCAI.

Multiple Choice Questions

1. What functions in Rshiny were used in the dashboard to allow for reactive functioning?

a. UpsetR, ggplot, and Rshiny widgets.
b. observeEvent, updatePickerInput, and renderPlot.
c. Server-side application programming.
d. Data management functions in R.

Correct Answer: The correct answer is option b. Using these three functions together allowed for the dashboard to update dynamically as the parameters were changed by the user.

2. Why was the NHSN solution not adopted?
a. It was too expensive.
b. Information about how to implement the NHSN solution was not available.
c. Evidence-based reports showed the NHSN solution had a lot of defects, and may not represent the best solution.
d. The NHSN solution did not cover CAUTI.

Correct Answer: The correct answer is option c. Reviews of the NHSN system suggested that data entry was difficult and confusing, and resulting data reports from the system were found to be inaccurate.

Protection of Human and Animal Subjects
No human data was used in this prototype. All the data were generated for this project.

Funding
None.

Conflict of Interest
None declared.

References

1 Hassan KA, Fatima BK, Riffat M. Nosocomial infections: epidemiology, prevention, control and surveillance. Asian Pac J Trop Biomed 2017;7(05):478–482

2 Kontula KSK, Skogberg K, Ollgren J, Järvinen A, Lyytikäinen O. Early deaths in bloodstream infections: a population-based case series. Infect Dis (Lond) 2016;48(05):379–385

3 Rolnick J, Downing NL, Shepard J, et al. Validation of test performance and clinical time zero for an electronic health record embedded severe sepsis alert. Appl Clin Inform 2016;7(02):560–572

4 Westra BL, Landman S, Yadav P, Steinbach M. Secondary analysis of an electronic surveillance system combined with multi-focal interventions for early detection of sepsis. Appl Clin Inform 2017;8(01):47–66

5 Tatham M, Macfarlane G, MacRae M, Tully V, Craig K. Development and implementation of a catheter associated urinary tract infection (CAUTI) 'toolkit'. BMJ Qual Improv Rep 2015;4(01):u205441.w3668

6 Centers for Disease Control and Prevention. Urinary Tract Infection (Catheter-Associated Urinary Tract Infection [CAUTI] and Non-Catheter-Associated Urinary Tract Infection [UTI]) and Other Urinary System Infection [USI]) Events. National Healthcare Safety Network; 2017. Available at: https://www.cdc.gov/nhsn/acute-care-hospital/cauti/index.html. Accessed June 30, 2019

7 Vincitorio D, Barbadoro P, Pennacchietti L, et al. Risk factors for catheter-associated urinary tract infection in Italian elderly. Am J Infect Control 2014;42(08):898–901

8 World Health Organization. Prevention of Catheter-Associated Urinary Tract Infection (CAUTI): Student Handbook; 2018. Available at: https://www.who.int/infection-prevention/tools/core-components/CAUTI_student-handbook.pdf. Accessed June 30, 2019

9 Gulf Cooperation Council. Healthcare Associated Infections Surveillance Manual. Riyadh, Saudi Arabia: MInistry of National Guard Health Affairs; 2018

10 Centers for Disease Control and Prevention. About the National Healthcare Safety Network (NHSN). Published 2015. Available at: https://www.cdc.gov/nhsn/about-nhsn/index.html. Accessed July 30, 2018

11 Bordeianou L, Cauley CE, Antonelli D, et al. Truth in reporting: how data capture methods obfuscate actual surgical site infection rates within a health care network system. Dis Colon Rectum 2017;60(01):96–106

12 Neelakanta A, Sharma S, Kesani VP, et al. Impact of changes in the NHSN catheter-associated urinary tract infection (CAUTI) surveillance criteria on the frequency and epidemiology of CAUTI in intensive care units (ICUs). Infect Control Hosp Epidemiol 2015;36(03):346–349

13 Horwich-Scholefield S, Keller V, Kazerouni N, Janssen L. Validation of Hospital Healthcare-Associated Infections (HAI) Reporting via the National Healthcare Safety Network (NHSN) With a Focus on Improving Case-Finding, California 2014. Open Forum Infect Dis 2015;2(Suppl 1. Doi: 10.1093/ofid/ofv133.182

14 Farrell L, Williams K, Satchell L. Creating consensus: ensuring inter-rater reliability for reporting infections using NHSN surveillance criteria. Am J Infect Control 2018;46(06):S6

15 Allen-Bridson K, Pollock D, Gould CV. Promoting prevention through meaningful measures: improving the Centers for Disease Control and Prevention's National Healthcare Safety Network urinary tract infection surveillance definitions. Am J Infect Control 2015;43(10):1096–1098

16 Talaat M, El-Shokry M, El-Kholy J, et al. National surveillance of health care-associated infections in Egypt: developing a sustainable program in a resource-limited country. Am J Infect Control 2016;44(11):1296–1301

17 Salama MF, Jamal W, Al Mousa H, Rotimi V. Implementation of central venous catheter bundle in an intensive care unit in Kuwait: effect on central line-associated bloodstream infections. J Infect Public Health 2016;9(01):34–41

18 Jahani-Sherafat S, Razaghi M, Rosenthal VD, et al. Device-associated infection rates and bacterial resistance in six academic teaching hospitals of Iran: findings from the International Nocosomial Infection Control Consortium (INICC). J Infect Public Health 2015;8(06):553–561

19 Simpao AF, Ahumada LM, Larru Martinez B, et al. Design and implementation of a visual analytics electronic antibiogram within an electronic health record system at a tertiary pediatric hospital. Appl Clin Inform 2018;9(01):37–45

20 Isikgoz Tasbakan M, Durusoy R, Pullukcu H, Sipahi OR, Ulusoy S; 2011 Turkish Nosocomial Urinary Tract Infection Study Group. Hospital-acquired urinary tract infection point prevalence in Turkey: differences in risk factors among patient groups. Ann Clin Microbiol Antimicrob 2013;12:31

21 Weiner LM, Fridkin SK, Aponte-Torres Z, et al. Vital signs: preventing antibiotic-resistant infections in hospitals - United States, 2014. MMWR Morb Mortal Wkly Rep 2016;65(09):235–241

22 Riley L, Guthold R, Cowan M, et al. The World Health Organization STEPwise approach to noncommunicable disease risk-factor surveillance: methods, challenges, and opportunities. Am J Public Health 2016;106(01):74–78

23 Wahi MM, Seebach P. Analyzing Health Data in R for SAS Users. Boca Raton, FL: CRC Press; 2017

24 Tableau. R for Statistical Computing & Analysis. Tableau Software. Available at: https://www.tableau.com/solutions/r. Accessed May 29, 2019

25 Roosan D, Del Fiol G, Butler J, et al. Feasibility of population health analytics and data visualization for decision support in the infectious diseases domain: a pilot study. Appl Clin Inform 2016;7(02):604–623

26 RStudio. RStudio. Available at: https://www.rstudio.com/. Accessed March 22, 2019

27 Chang W, Cheng J, Allaire JJ, et al. Shiny: Web Application Framework for R; 2018. Available at: https://CRAN.R-project.org/package=shiny. Accessed March 12, 2019

28 RStudio. Modularizing Shiny app code. Published 2017. Available at: https://shiny.rstudio.com/articles/modules.html. Accessed March 22, 2019

29 Chang W, Ribeiro BB. Shinydashboard: Create Dashboards with "Shiny"; 2018. Available at: https://CRAN.R-project.org/package=shinydashboard. Accessed March 25, 2019

30 Wickham H, Chang W, Henry L, et al. Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics; 2018. Available at: https://CRAN.R-project.org/package=ggplot2. Accessed April 3, 2019

31 Conway J, Gehlenborg N. UpSetR: A More Scalable Alternative to Venn and Euler Diagrams for Visualizing Intersecting Sets; 2017. Available at: https://CRAN.R-project.org/package=UpSetR. Accessed April 3, 2019

32 RStudio. Shiny from RStudio. Available at: https://shiny.rstudio.com/. Accessed March 13, 2019

33 RStudio. ShinyApps.io. Available at: https://www.shinyapps.io/. Accessed March 22, 2019

34 Dukach N, Wahi MM. CAUTI Surveillance Shiny dashboard app. Published 2019. Available at: https://cauti.shinyapps.io/CAUTI-dashboard/. Accessed March 22, 2019

35 shinyWidgets.pdf. Available at: https://cran.r-project.org/web/packages/shinyWidgets/shinyWidgets.pdf. Accessed March 13, 2019

36 Shiny - How to understand reactivity in R. Available at: https://shiny.rstudio.com/articles/understanding-reactivity.html. Accessed March 18, 2019

37 Adding Reactive Expressions in Shiny Application and Customizing the Shiny Appearance | Analytics Profile. Available at: https://analyticsprofile.com/r-shiny-dashboard-tutorial/tutorial-3-adding-reactivity-and-customizing-appearance/. Accessed March 22, 2019

38 Karavite DJ, Miller MW, Ramos MJ, et al. User testing an information foraging tool for ambulatory surgical site infection surveillance. Appl Clin Inform 2018;9(04):791–802