AMR-TV: Antimicrobial Resistance Transmission Visualizer

Ivan S. Gill

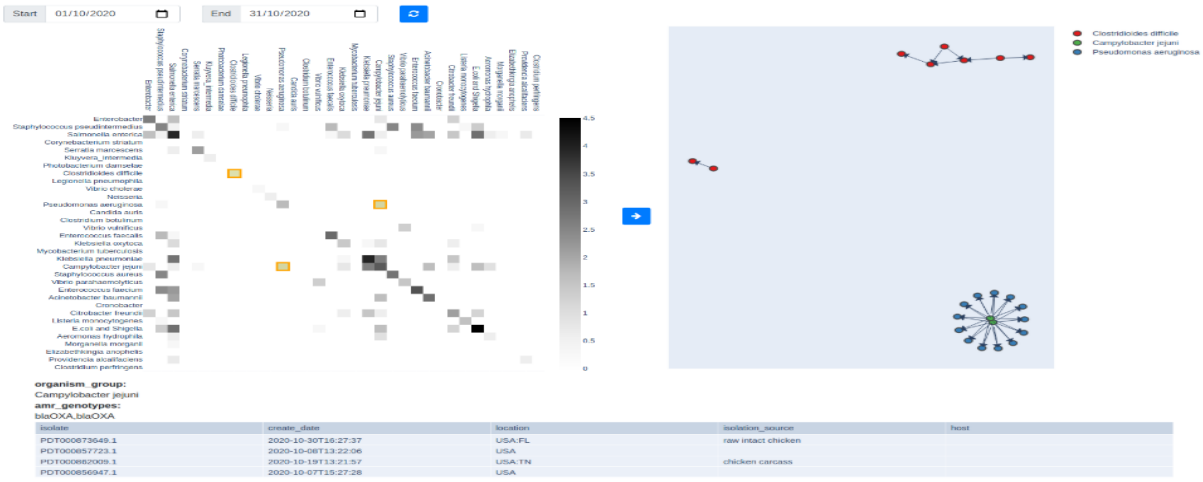

Fig. 1. The AMR-TV interface. Top left: high-level adjacency matrix view. Top right: mid-level NL diagram view. Bottom: low-level ND table view.

Abstract—We present the application AMR-TV, a visualization solution for identifying potential AMR transmission routes from the publicly available data in the NCBI Isolates Browser. AMR-TV uses three successive views with increasing levels of detail: an adjacency matrix view, a node link diagram view, and table view. We show how infection control healthcare workers can use these three views to elucidate the source and extent of specific AMR outbreaks, and in some cases, even hypothesize the geographic and host relationships facilitating potential AMR transmission routes. During the development of AMR-TV, we ran into some problems related to the inconsistent data standards of the NCBI Isolates Browser, as well as our limited domain knowledge on AMR transmission. We must fix these problems in the future to improve AMR-TV's real-world applicability.

Index Terms—AMR, AMR transmission

1 Introduction

Antimicrobial resistance (AMR) describes the resistance of microorganisms to antimicrobial medication [1]. One example of AMR, that is a critical healthcare challenge globally, is antibiotic-resistant bacteria. AMR genes evolve in bacterial populations due to antibiotic overuse [2]–[4], and upon evolving, spread quickly from population to population through a process known as horizontal gene transfer (hgt) [5]. We refer to this process as AMR transmission, and it can occur between distinct populations within and across bacterial species.

For infection control healthcare workers, monitoring AMR transmission routes between bacterial populations is critical to investigating the source and extent of AMR outbreaks. One potential workflow for identifying AMR transmission routes is to:

1. Identify potential AMR transmission routes by comparing AMR genes across bacterial populations
2. Validate potential AMR transmission routes using phylogenetic sequence comparison techniques [6]

Step one of this workflow can be difficult, as data on AMR genes identified in different bacteria is often aggregated into tables containing thousands of rows. Comparing AMR genes across so many rows for potential AMR transmission routes is difficult, so a tool for automatically summarizing potential AMR transmission routes from tabular data sources would make the above workflow much more efficient.

We present AMR-TV (antimicrobial transmission visualizer), a web application designed to visualize the AMR transmission routes of one such tabular data source, called the NCBI Isolates Browser [7]. AMR-TV derives a directed network from the ~700k rows available in the NCBI Isolates Browser to produce three tightly-coupled views: an adjacency matrix view, a node link (NL) diagram view, and a node detail (ND) table view. The adjacency matrix view summarizes the number of potential AMR transmission routes between bacterial species over a specified date range, while the NL diagram and ND table views display and contextualize individual routes more clearly.

2 Related Work

Several applications have either attempted to solve a domain problem or implement a visualization solution similar to AMR-TV.

2.1 Horizontal gene transfer (hgt) visualizations

Most visualization solutions for monitoring the transmission of genetic material across populations have focused on the vertical movement of genes through phylogenetic trees. The horizontal transmission of genetic material between contemporary populations through processes like hgt is either ignored, or provided an inadequate idiom tightly coupled to a phylogenetic tree representation. One example of this is T-REX [8], a web server that uses gene sequence comparison methods to construct a traditional phylogenetic tree for visualizing vertical transmission, and then superimposes dashed edges between branches to indicate hgt

Table 1. Details on the seven NCBI Isolate Browser data item attributes.

Attribute	Description	Example	Type	Cardinality	Sandardized?	
isolate	A unique code, like a primary key, used to identify the sampled pathogen in the NCBI Isolates Browser	PDT0003701 31.1	Categorical	~700k (a unique one for every data item)	Yes	
organism_group	Bacterial species of the sampled pathogen	Klebsiella pneumoniae	Categorical	33	Yes	Ranges from 2010-12-15 14:23:00 to 2020-10-31 05:26:45
create_date	Date and time the sampled pathogen was uploaded to the NCBI Isolates Browser	2018-08-25 08:22:36	Ordinal	~260k	Yes	
location	Geographic origin of the sampled pathogen	United Kingdom: (Scotland)	Categorical	~4200	No	
isolation_source	Physical or environmental source of the sampled pathogen	urine	Categorical	~1500	No	
host	Host species infected with sampled pathogen	Homo sapiens	Categorical	~1200	No	
amr_genotypes	AMR genes found in sampled pathogen	{aadA2, blaSHV-11, catA1}	Set of categorical values	~77k unique sets	Yes	Sets range in size from 1 to 53 items, and there are ~2600 unique AMR genes across ~77k unique sets

events. This is not an acceptable solution for visualizing more than a few hgt events per branch, as it would lead to multiple dashed edge crossings. Indeed, in the case of AMR genes, where the majority of movement is horizontal, it is likely that T-REX would produce a hairball of dashed edges over relatively small trees. To adequately visualize large-scale hgt networks, we must move beyond attempts to confine them within traditional phylogenetic representations.

2.2 Visualizing the movement of genes over geographic space

Although there has been a lack of solutions for visualizing hgt beyond traditional phylogenetic representations, there have been some distinct solutions for visualizing the transmission of genes across geographic space.

One such solution is GenGIS [9], which visualizes the geospatial distribution of genes by superimposing the corresponding network of nodes on a geographic map. Although this solution could support a larger network than one confined within a phylogenetic tree, particularly if the nodes are geographically far apart, link crossings and hairballing remain a concern if tasked with visualizing more than a few dozen nodes. AMR-TV uses a force-directed layout algorithm to minimize link crossings in its NL diagram view, but a similar solution is not possible with GenGIS, due to the positional encoding of nodes on the geographic map.

Microreact is another application that superimposes nodes on a geographic map [10], but improves scalability by only drawing links between nodes in a secondary view with no positional encodings. Not only does this two view solution minimize link crossings by allowing the use of force-directed layout algorithms, it also introduces the concept of filtering the network prior to visualizing node-link paths, as users must specify the nodes they want added to the secondary view. This is an important feature to consider when visualizing the paths in a network with thousands of nodes, as users may only be interested in visualizing the connections between a subset of nodes. AMR-TV accomplishes something similar by allowing users to filter the network by bacterial species through the adjacency matrix view, before visualizing a subset of nodes in the NL diagram view.

2.3 Social network visualizations

Similar to AMR transmission routes, the relationships between actors of a social network are most intuitively represented as a NL diagram, but the density of these networks necessitates some mechanism for summarizing and filtering data. Like AMR-TV, many social network visualization solutions accomplish this by coupling an adjacency matrix view to their NL diagram view. However, this coupling is configured in various ways. VAIM uses an adjacency matrix view to summarize the density of its coupled NL view at every x and y position [11]. Users can jump to positions of the NL view with specific node densities by selecting adjacency matrix cells. This functionality does not make sense for AMR-TV, where the goal is to visualize clear paths between individual nodes of interest, rather than detect clusters of nodes. In contrast, MatLink and MatrixExplorer both use their adjacency matrix views to summarize connections between different categories of nodes [12], [13]. This is similar to how AMR-TV's adjacency matrix view summarizes connections between nodes belonging to different bacterial species. However, MatLink uniquely concatenates its adjacency matrix and NL views by overlaying the links between nodes directly on the edges of the adjacency matrix view. This is not an acceptable solution for AMR-TV, as it obscures the directionality of links, and the topology of multiple paths represented by each adjacency matrix cell. MatrixExplorer offers the solution most similar to AMR-TV's proposed solution, by allowing users to filter a subset of data through adjacency matrix cell selection prior to generating the NL diagram view.

3 Data Abstractions

The domain data that infection control healthcare workers can use to infer potential AMR transmission events is usually available in tabular format. AMR-TV derives a network from one example of this domain data, called the NCBI Isolates Browser, to provide a more intuitive representation of AMR transmission routes.

3.1 Domain data: AMR detection

Microbiologists identify AMR genes in sampled bacteria genomes by referencing external databases for previously identified AMR gene sequences [14]. Information on sampled AMR bacteria is then submitted to external databases like NCBI and the European Nucleotide Archive [15], where it is available in tabular format. Infection control healthcare workers can compare the AMR genes identified across rows of different bacteria to determine potential AMR transmission routes between bacteria populations, based on AMR gene overlap. However, due to the decreasing costs of whole genome sequencing [16], and increasing spread of AMR bacteria across the globe [17], thousands of new rows are being added to

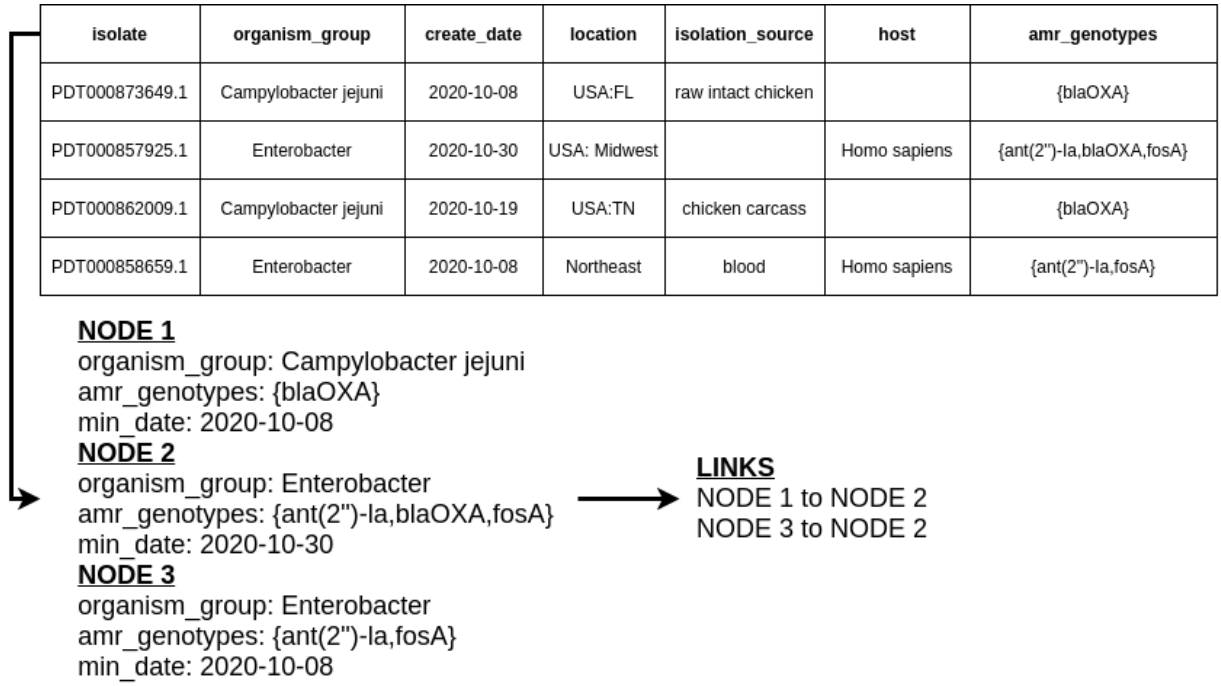

Fig. 2. Example illustrating the generation of three DN nodes and two DN links from four NCBI Isolates Browser data items.

NCBI and the European Nucleotide Archive every month. It is difficult to track and compare AMR genes across that many rows, especially if some rows are found in multiple potential transmission routes. Filtering by contextual attributes that are of relevance to a particular AMR outbreak, like sampling date and bacterial species, does not always help, and can still yield hundreds to thousands of rows.

3.2 Application data: NCBI Isolates Browser

The NCBI Isolates Browser data source used by AMR-TV is an accurate representation of the domain data usually available to infection control healthcare workers. It is a static table dataset that we downloaded from the publicly available NCBI repository on Oct 31, 2020. The NCBI Isolates Browser data source has ~700k data items representing a variety of sampled bacterial pathogens with different AMR genes. Each data item has seven attributes, as seen in Table 1. More attributes were available in the publicly available NCBI repository, but were not relevant to AMR-TV's goal of visualizing AMR transmission routes. Of the seven attributes in each data item, only organism_group, isolate, create_date, and amr_genotypes are standardized. The rest have both missing and inconsistent values. However, only the clean attributes were used to construct the derived network in the next section.

3.3 Application data: Derived Network (DN)

The derived network (DN) data source is a network dataset derived from the NCBI Isolates Browser data source. A single DN node is generated for every unique combination of organism_group and amr_genotypes values, as seen in Fig. 2. This process results in a one-to-many relationship between DN nodes and NCBI Isolates Browser data items. Details on the cardinality of this relationship can be seen in Table 2.

Table 2. Cardinality of many-to-one relationships between NCBI Isolates Browser data items and DN nodes, without any filters on create_date and organism_group values.

___ number of NCBI Isolates Browser data items DN nodes are mapped to	
Minimum	1
Maximum	93
Average	~9

Each DN node is assigned three attributes:

1. organism_group
2. amr_genotypes
3. min_date

organism_group and amr_genotypes are the unique combination of values corresponding to the DN node. min_date is the minimum create_date value across the >1 NCBI Isolates Browser data items mapped to the DN node.

As seen in Fig. 2, directional links are formed between any two DN nodes A and B where:

1. A.min_date < B.min_date
2. A.amr_genotypes ⊆ B.amr_genotypes

Each link represents a potential AMR transmission route between all the NCBI Isolates Browser data items mapped to A and all the NCBI Isolates Browser data items mapped to B. We do not produce a one-to-one mapping of DN nodes to NCBI Isolates Browser data items, because this criteria for deriving links would result in many NCBI Isolates Browser data items with identical amr_genotypes values producing hundreds of identical node-link paths. We choose to emphasize the variety of potential AMR transmission routes across different species and amr genotypes with our one-to-many mapping instead.

Unlike the NCBI Isolates Browser data source, the DN data source is dynamic. DN nodes and links can be derived from any subset of NCBI Isolates Browser data items. This can be useful when investigating specific AMR outbreaks, when certain NCBI Isolates Browser data items could be irrelevant due to their create_date and organism_group values. As a result, the size of the DN data source can vary. The size of DN data sources filtered by different create_date and organism_group values is described in Table 3.

Table 3. The cardinality of different DN data source generated from different subsets of the NCBI Isolates Browser data source. The filters refer to create_date ranges and organism_group values used to filter the NCBI Isolates Browser data source.

Filters	# of nodes	Avg # of links per node
None	~77k	Too many to compute
Oct 1, 2020 to Oct 31, 2020	~5k	~17
Oct 1, 2020 to Oct 31, 2020 *Enterobacter* isolates only	190	~5
Oct 1, 2020 to Oct 31, 2020 *Enterobacter* and *Campylobacter jejuni* isolates only	384	~8
Oct 1, 2020 to Oct 31, 2020 *Salmonella Enterica* isolates only	439	~20

4 Task abstractions

AMR-TV has three distinct categories of tasks that require three different views of the application data described in the previous section.

4.1 High-level tasks

Infection control healthcare workers must first be able to filter the NCBI Isolates Browser data items by a create_date range of relevance to a particular AMR outbreak they are investigating. AMR-TV must then produce a DN from the filtered subset of NCBI Isolates Browser data items, and summarize the number of links between DN nodes of different organism_group values. Ideally, infection control healthcare workers would already have organism_group values in mind when investigating an AMR outbreak, since they would know the bacterial species that have been observed during the outbreak. But comparing the number of incoming links from DN nodes of other organism_group values would also allow infection control healthcare workers to identify bacterial species that, although not directly observed during the AMR outbreak, are part of potential AMR transmission routes leading to the outbreak.

To summarize, infection control healthcare workers must be able to analyze, produce, and derive a DN from a subset of the NCBI Isolates Browser data using the create_date attribute. They must then be able to analyze, consume, and discover all organism_group values of relevance to an outbreak, using the DN topology, through a summary of links between DN nodes of different organism_group values.

4.2 Mid-level tasks

Infection control healthcare workers must then be able to further filter the DN by the organism_group values they learned were of relevance to an AMR outbreak through the high-level tasks. AMR-TV must produce a clear view of the filtered DN topology, so infection control healthcare workers can identify potential AMR transmission routes through a visualization of the node-link paths. The links should be displayed with their directionality clearly visible, to clarify the order of potential AMR transmission routes.

To summarize, infection control healthcare workers must be able to analyze, produce, and derive a DN that has been further filtered by specified values of the organism_group attribute. AMR-TV must then analyze, consume, and present a visualization of the potential AMR transmission routes represented by the remaining DN paths.

4.3 Low-level tasks

Finally, infection control healthcare workers must be able to map the DN nodes, in the visualized node-link paths, back to the original NCBI Isolate Browser data items for further context on each potential AMR transmission route. For example, if the NCBI Isolate Browser data items representing two linked DN nodes are not missing their location and host values, infection control healthcare workers could use those values to identify the geographic and host relationships responsible for bringing the bacteria mapped to each DN node close enough together for AMR transmission to have potentially occurred.

To summarize, infection control workers must be able to query and identify NCBI Isolates Browser data items mapped to specific nodes in the visualized DN topology.

5 Solutions

As seen in Fig. 1, AMR-TV uses three juxtaposed and coordinated views with distinct visualization idioms for high-, mid-, and low-level tasks. These are the adjacency matrix, node-link (NL) diagram, and node-detail (ND) table views respectively.

5.1 High-level view: adjacency matrix

The high-level view allows users to set "start" and "end" date values, which AMR-TV then uses as a create_date range for filtering the NCBI Isolates Browser data items, creating a DN data source, and generating an adjacency matrix. The x and y coordinates of the adjacency matrix encode the 33 different organism_group values found in the NCBI Isolates Browser, and the cells of the adjacency matrix are assigned the log10 number of links between nodes of different organism_group values in the generated DN. For example, if the x and y coordinates of a cell are *Salmonella enterica* and *Enterobacter*, and there are 100 links in the DN between nodes of the organism_group values *Salmonella enterica* and *Enterobacter*, then the cell would be assigned a value of 2. The luminance channel is used to encode the values assigned to each cell, with darker colours representing bigger numbers, and a colour bar next to the adjacency matrix as reference. Hovering over cells also displays their values. Users can select cells by clicking on them, which highlights the cells in yellow. These selected cells are used to create the mid-level NL diagram view.

5.2 Mid-level view: node-link (NL) diagram

When creating the mid-level view, the DN is first filtered using the x and y organism_group values of user-selected cells in the high-level adjacency matrix view. Specifically, any node-link path between two nodes of organism_group values A and B is removed if the adjacency matrix cell at coordinates (A, B) was not selected. Then, a NL diagram is created using the Fruchterman-Reingold force-directed layout algorithm. A qualitative nine colour hue scheme is used to encode the organism_group value of different nodes in the NL diagram, with a legend on the top right available as reference. It is unlikely that infection control healthcare workers would need to visualize AMR transmission routes between more than a few bacterial species at a time, but if they do, the colours for different organism_group values will begin to repeat. A toolbar on the top of the NL diagram offers options for navigating the NL diagram through geometric zooming or panning. Arrowheads are placed on the links of the NL diagram to indicate directionality, and the min_date value of each node is visible on hover.

5.3 Low-level view: node-detail (ND) table

Clicking a node in the mid-level NL diagram view will generate a low-level ND table view of NCBI Isolates Browser data items mapped to that particular node. This table encodes every attribute originally found in the NCBI Isolates Browser data source, except organism_group and amr_genotypes, which are displayed above the table, as they are the same for every data item mapped to a single node.

5.4 Alternative considerations we considered

During implementation, we considered two more encodings in the NL diagram view that we originally pitched in our project proposal. First, we considered using the luminance channel to encode the min_date value of nodes, with darker colours representing more recent min_date values. However, we later realized that although this would be an effective channel for comparing min_date values between nodes of a single colour hue or organism group, it would not be an effective channel for comparing nodes across different colour hues. As a result, we instead decided to display the min_date

value of nodes when the user hovers over them. Second, we considered differentiating links between nodes of the same organism_group value, and links between nodes of different organism_group values, by using dashed links for the former. We thought this would be an effective way to emphasize potential AMR transmission routes between different bacterial species, as we originally thought any AMR gene overlap across individuals of the same species was likely due to common ancestry, and not hgt. We later learned that this was an unfair assumption, and hgt is the primary mechanism by which AMR genes spread between populations both within and across species. As a result, we decided against using the dashed link encoding to emphasize one type of transmission route over the other.

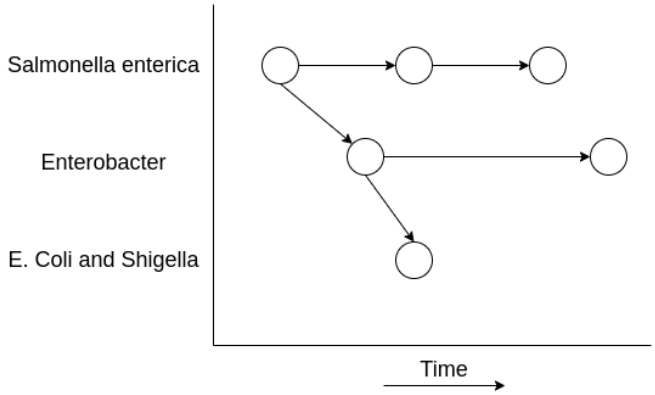

Fig. 3. Mock up of an alternative time series visualization we considered for the mid-level view.

During the earliest design stages of AMR-TV, we considered using a time series diagram for the mid-level view, instead of a NL diagram. The time series would have been segmented by organism_group on a chronologically linear axis, similar to Fig. 3. We were inspired by similar, smaller-scale time series visualizations we had previously seen used to represent AMR transmission [18]. However, we soon realized all these visualizations were manually rendered by humans to minimize link crossings, whereas AMR-TV would not have that benefit. We could not find any solutions to automatically minimize link crossings between hundreds of nodes with positional encodings, like the nodes that would be found in a segmented time series, so we switched to a force-directed NL diagram instead.

We also considered replacing the NL diagram in the mid-level view with a second adjacency matrix, next to the high-level adjacency matrix on the left. Some bacterial species, like *Salmonella enterica* and *E. Coli*, are so frequently sampled that the NL diagram often shows hairballing for DN nodes of those organism_group values. This made us worried about readability, and the ability of users to parse potential AMR transmission routes from said hairballs. We considered solving this problem by using a mid-level adjacency matrix view with the DN nodes as x and y coordinates, and a boolean value in each adjacency matrix cell encoding the presence or absence of a link. This solution would have supported thousands of DN nodes, without obscuring any direct paths between any two DN nodes. However, we ultimately decided against using a second adjacency matrix view for two reasons. First, it is more time consuming and less intuitive to identify sequential node-link paths between multiple DN nodes by jumping from cell to cell of an adjacency matrix, than it is to follow the paths in a NL diagram. This is particularly true when the adjacency matrix has hundreds of DN nodes along its axes. Second, most DN nodes are not affected by the level of hairballing seen in the few frequently sampled bacterial species. Replacing the NL diagram view with a less intuitive representation of the DN topology for a select few organism_group values seemed counter-productive. In the end, we decided the best solution for navigating any hairballs from the few frequently sampled bacterial species was to rely on the geometric zooming and panning features of the NL diagram.

6 Implementation

We implemented AMR-TV using a Django and Postgres framework, containerized through Docker. We supplemented the Django web framework with the front-end library JQuery, and the CSS framework Bootstrap. The boilerplate for the Django, Postgres, and Docker setup we used was inspired by the setup available from Cookiecutter [19]. One significant modification we made to the boilerplate was changing the OS used by the Django container from Alpine to Debian buster, as the former OS prevented us from installing several visualization library dependencies. We created a unique model in Django to store the NCBI Isolates Browser data in a Postgres table, and we wrote Django functions for rendering the high-, mid-, and low-level views. We used jQuery's ajax functionality to provide parameters to these functions, and insert the returned renders into the visible HTML. The two visualization libraries we used were Plotly and NetworkX.

6.1 High-level adjacency matrix view implementation

To implement the high-level adjacency matrix view, we wrote a Django function that runs a SQL query on the NCBI Isolates Browser data items stored in the Postgres database, to return a list of DN node-link relationships. We filter this list with a create_date range specified by the user, before iterating over said list to count the number of links between DN nodes of different organism_group values. We provide a log10 representation of these values to Plotly's heatmap function, which renders the adjacency matrix ultimately inserted into the HTML.

Plotly does not provide a built-in method for selecting or highlighting cells in heatmap objects, so we added that functionality ourselves. We use jQuery to obtain the coordinates of any click events on the adjacency matrix view, and then mapped these coordinates to the four corners of the nearest adjacency matrix cell. We then used these coordinates, and Plotly's add_shape function, to overlay a transparent yellow square over the clicked cell.

6.2 Mid-level NL diagram view implementation

To implement the mid-level NL diagram view, we wrote a Django function that receives a list of highlighted adjacency matrix cells to further filter the list of DN node-link relationships used previously. We then iterate over these filtered relationships to generate the data needed to construct a NetworkX graph object, with the appropriate node and link attributes. We then use the NetworkX built-in spring_layout function to run the Fruchterman-Reingold force-directed layout algorithm, which assigns each node a horizontal and vertical position. We finally render the NL diagram by passing the NetworkX node, link, and positional information to Plotly's scatter function.

6.3 Low-level ND table view implementation

For the low-level ND table view, we use jQuery to monitor the mid-level NL diagram view for click events, and we pass information on clicked nodes to a Django function, that uses Plotly's table function, to render the mapped data items found in the NCBI Isolates Browser Postgres table.

7 Milestones

1. Setup Django+Postgres framework in Docker container, with NetworkX and Plotly installed
- Estimated hours: 4
- Estimated completion: Oct 26
- Actual hours: 4
- Actual completion: Oct 26
2. Create Django fixture containing NCBI data, and load it into Postgres database
- Estimated hours: 8
- Estimated completion: Nov 3
- Actual hours: 8
- Actual completion: Nov 3
3. Create API calls for querying database that will be needed for user-specified visualizations

- Estimated hours: 16
- Estimated completion: Nov 13
- Actual hours: 16
- Actual completion: Nov 13

4. Using Plotly, create adjacency matrix with shaded cells and colour bar, using data from some stub, static period of time
- Estimated hours: 8
- Estimated completion: Nov 16
- Actual hours: 6
- Actual completion: Nov 16

5. Refine stub matrix to represent one-to-many mapping of DN nodes and NCBI Isolates Browser data items
- Estimated hours: 1
- Estimated completion: Nov 20
- Actual hours: 8
- Actual completion: Nov 23

6. Using Plotly and Networkx, generate directionless and colourless NL diagram for stub data used in adjacency matrix
- Estimated hours: 8
- Estimated completion: Nov 24
- Actual hours: 6
- Actual completion: Nov 26

7. Add colour, luminance, and directionality to node-link diagram
- Estimated hours: 8
- Estimated completion: Nov 26
- Actual hours: 4
- Actual completion: Nov 27

8. Set-up ability to to select nodes in stub node-link diagram, and see tables
- Estimated hours: 8
- Estimated completion: Nov 28
- Actual hours: 2
- Actual completion: Nov 28

9. Set-up interface to allow users to select time period of interest, dynamically update adjacency matrix, and dynamically update node-link diagram
- Estimated hours: 8
- Estimated completion: Nov 30
- Actual hours: 16
- Actual completion: Nov 30

8 Results

We consider three scenarios when using AMR-TV.

8.1 Scenario 1: sparse AMR transmission network

There has been an outbreak of AMR Pseudomonas aeruginosa across humans living in San Diego, CA, during the month of Oct 2020. An infection control healthcare worker seeking to investigate the outbreak enters a create_date range of Oct 1, 2020 to Oct 31, 2020 in the AMR-TV interface. The resulting adjacency matrix view alerts the healthcare worker to a significant number of potential AMR transmission events between Pseudomonas aeruginosa and Pseudomonas aeruginosa nodes, but also interestingly, between Pseudomonas aeruginosa and *Campylobacter jejuni* nodes. The healthcare worker selects the adjacency matrix cells representing both relationships, and retrieves the NL diagram seen in Fig. 4. They observe multiple potential AMR transmission routes between only a few *Campylobacter jejuni* nodes and many Pseudomonas aeruginosa nodes. By cycling through the ND tables of these nodes, they realize most of the NCBI samples mapped to the few *Campylobacter jejuni* nodes are from raw poultry, and most of the NCBI samples mapped to the many Pseudomonas aeruginosa nodes are from humans living in San Diego, CA. The healthcare worker considers this a potential AMR transmission route worth validating through further sequence comparisons, and wonders if the San Diego, CA population of Pseudomonas aeruginosa picked up AMR genes from Campylobacter jejuni bacteria at a local poultry processing plant, before infecting individuals that possibly worked there.

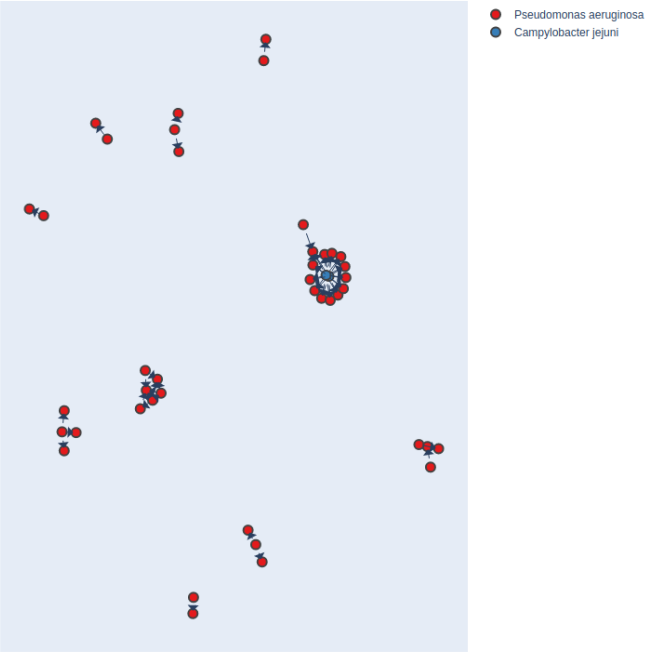

Fig. 4. NL diagram view described in Scenario 1.

8.2 Scenario 2: dense AMR transmission network

There has been an outbreak of various AMR *E. Coli* across humans in the United Kingdom, over the weekend of Oct 10, 2020 to Oct 13, 2020. An infection control healthcare worker seeking to investigate the outbreak enters this create_date range in the AMR-TV interface. The resulting adjacency matrix view shows a large number of potential AMR transmission events between *E. Coli* and *E. Coli* nodes, but none between *E. Coli* and nodes of other species. As a result, the healthcare worker only selects the adjacency matrix cell representing potential AMR transmission events within *E. Coli*, and is horrified to observe the resulting NL diagram hairball, as seen in Fig. 5A. However, by zooming into the hairball, as seen in Fig. 5B, the healthcare worker observes a node at the center of the hairball with many outgoing links. This node seems like it could be the source of the AMR *E. Coli* outbreak, and by clicking on this node to see its ND table, the healthcare worker learns that most of the NCBI samples mapped to the node are from cattle farms in NC, USA. Therefore, there may be a route of AMR transmission between *E. Coli* in NC cattle, and *E. Coli* in UK humans--perhaps linked to the import of American beef, or a NC cattle farmer that visited the UK.

8.3 Scenario 3: missing contextual data causes a scare

There have been outbreaks of AMR Enterococcus faecalis across multiple livestock farms in the USA, over a period of several weeks in Sep 2020. An infection control healthcare worker seeking to investigate the outbreaks enters a relevant create_date range of Sep 8, 2020 to Sep 22, 2020 in the AMR-TV interface. The resulting adjacency matrix view shows the presence of some potential AMR transmission events between Enterococcus faecalis and *Listeria monocytogenes* nodes. The healthcare worker begins to panic at the thought of Listeria monocytogenes potentially being in close enough proximity to the livestock, currently infected with Enterococcus faecalis, for AMR transmission to occur. *Listeria monocytogenes* is more dangerous to humans than Enterococcus faecalis, and if it infects the livestock like Enterococcus faecalis, consumers of the livestock meat could become severely ill. The healthcare worker selects the cells representing potential AMR transmission routes between Enterococcus faecalis and *Listeria monocytogenes*, to generate a NL diagram view. They then click on the single *Listeria monocytogenes* node with multiple outgoing links to Enterococcus faecalis to generate a ND table, and are frustrated to learn that due to a lack of standardization in several

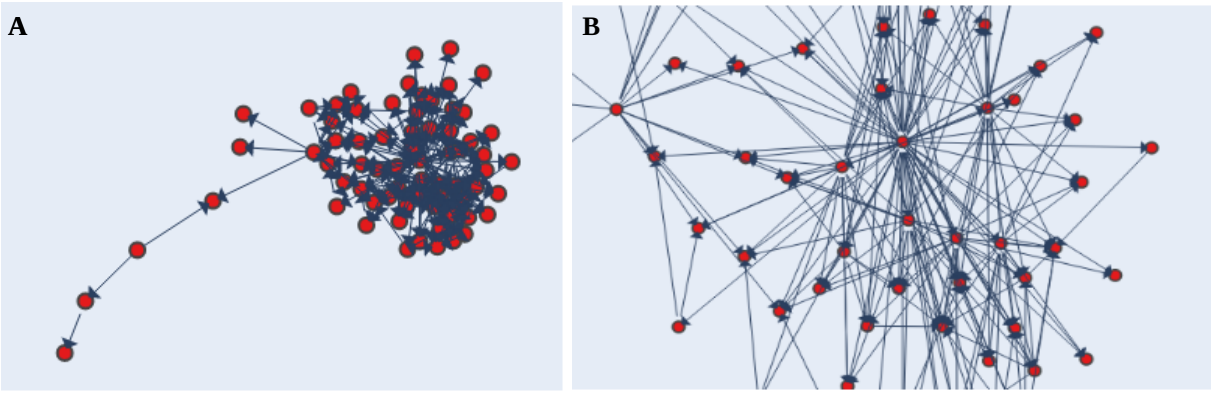

Fig. 5. NL diagram view described in Scenario 2 A) when not zoomed in, B) and when zoomed in.

organism_group:
Listeria monocytogenes
amr_genotypes:
tet(M)

isolate	create_date	location	isolation_source	host
PDT000835656.2	2020-09-11T14:14:42	Not available	Not available	Not available
PDT000835659.2	2020-09-11T14:15:12	Not available	Not available	Not available

Fig. 6. ND table view described in Scenario 3.

NCBI Isolate Browser data item attributes, there is no information on location, isolation_source, and host for the two *Listeria monocytogenes* NCBI samples mapped to that node, as seen in Fig. 6. The healthcare worker has identified a potential AMR transmission route, but due to missing contextual data in the ND table view, is unable to determine the geographic or host relationships responsible for potentially bringing *Listeria monocytogenes* in close contact to livestock meant for human consumption.

9 Discussion

In practice, AMR-TV has several benefits. First, AMR-TV definitely allows for faster identification of potential AMR transmissions routes than tabular data alone. It is much easier to automatically quantify and visualize potential AMR transmission routes through the adjacency matrix and NL diagram views, than it is to obtain the same information by flipping through thousands of rows in a table. Second, AMR-TV attempts to visualize potential hgt events at a scale we have not seen other hgt visualization solutions attempt. The adjacency matrix view can summarize thousands of potential AMR transmission routes, and the NL diagram view can explicitly visualize hundreds. Third, the containerization framework of AMR-TV allows users to deploy the application on a local server, if the privacy of data being visualized is ever a concern.

However, the real-world applicability of AMR-TV is limited by several weaknesses in its design and implementation. First, the unstandardized values of several attributes in the NCBI Isolates Browser data source prevents users from identifying the geographic and host relationships responsible for facilitating potential AMR transmission routes. Second, potential AMR transmission routes between NCBI Isolates Browser data items mapped to the same DN node are never considered. If every NCBI Isolate Browser data item mapped to a DN node was from the same population, this would not be an issue, as infection control healthcare workers are more interested in AMR transmission between distinct populations. However, without standardized values for host, location, and isolation_source to consider when generating DN nodes, in addition to the values of organism_group and amr_genotypes already being considered, we could not realistically achieve such a mapping between nodes and populations.

Third, due to a lack of AMR domain knowledge at the onset of this project, we made two inaccurate domain assumptions:

1. Bacteria transmit all their AMR genes during transmission
2. The directionality of AMR transmission can be inferred from bacteria sampling date

Neither assumption is true, however neither assumption can be safely ignored without significant changes in implementation. Removing domain assumption 1 adds hundreds of links per DN node across a variety of create_date ranges, which makes the NL diagram view completely unreadable. And in the case of domain assumption 2, we have learned the only way to infer the directionality of AMR transmission accurately is by actually comparing AMR gene sequences. Indeed, if we were to compare AMR gene sequences prior to visualization, as opposed to leaving it as a validating step afterwards, we could not only infer directionality, but also mitigate the drastic increase in links caused by ignoring domain assumption 1. Comparing AMR genes sequences calculates the likelihood of hgt between any two samples, so we could reduce the number of links by only visualizing potential AMR transmission events with high likelihood values.

9.1 Future work

We want to allow users to upload their own tabular datasets, as opposed to only ever visualizing AMR transmission routes from the NCBI Isolates Browser. However, it has become clear that for AMR-TV to have any real-world applicability, we must do two things. First, we must enforce a common vocabulary on user-uploaded files to ensure there are no unstandardized or missing values for any contextual attributes. Second, we must require users to upload the accompanying AMR gene sequence datasets for preliminary comparison prior to visualization.

We intend to create a three part pipeline. Part one of the pipeline will create a DN from the tabular data source as AMR-TV does now, but with a few adjustments. First, we will consider the now standardized location, host, and isolation_source values when creating DN nodes, so all bacteria mapped to a single node are more likely to come from the same population. Second, we will remove the two domain assumptions. Part two of the pipeline will then compare the AMR gene sequences of DN nodes to infer link directionality, and remove any node-link paths representing transmission events with low likelihood values for hgt. Part three of the pipeline will visualize the DN in a high-, mid-, and low-level view similar to the current AMR-TV solution.

10 Conclusion

We presented AMR-TV, a visualization solution that identifies potential AMR transmission routes from the NCBI Isolates Browser by generating dynamic DN datasets over user-specified create_date ranges. Using a high-level adjacency matrix view, mid-level NL diagram view, and low-level ND table view, AMR-TV allows infection control healthcare workers to parse and identify potential AMR transmission routes of interest when investigating a particular AMR outbreak. However, the real-world applicability of AMR-TV is limited due to missing values in the NCBI Isolates Browser dataset, and faulty assumptions made about AMR transmission during development. To mitigate these limitations, we must modify AMR-TV in the future by standardizing all application data sources, and integrating AMR gene sequence comparisons.

References

[1] S. B. Levy, "Microbial resistance to antibiotics. An evolving and persistent problem.," Lancet, vol. 2, pp. 83–88, 1982.

[2] I. A. Rather, B.-C. Kim, V. K. Bajpai, and Y.-H. Park, "Self-medication and antibiotic resistance: Crisis, current challenges, and prevention," Saudi J. Biol. Sci., vol. 24, no. 4, p. 808, May 2017, doi: 10.1016/j.sjbs.2017.01.004.

[3] C.-E. Luyt, N. Bréchot, J.-L. Trouillet, and J. Chastre, "Antibiotic stewardship in the intensive care unit," Crit. Care, vol. 18, no. 5, p. 480, Aug. 2014, doi: 10.1186/s13054-014-0480-6.

[4] K. L. Tang et al., "Restricting the use of antibiotics in food-producing animals and its associations with antibiotic resistance in food-producing animals and human beings: a systematic review and meta-analysis," Lancet Planet. Health, vol. 1, no. 8, p. e316, doi: 10.1016/S2542-5196(17)30141-9.

[5] K. A. Moser et al., "The Role of Mobile Genetic Elements in the Spread of Antimicrobial-Resistant Escherichia coli From Chickens to Humans in Small-Scale Production Poultry Operations in Rural Ecuador," Am. J. Epidemiol., vol. 187, no. 3, pp. 558–567, Mar. 2018, doi: 10.1093/aje/kwx286.

[6] M. Ravenhall, N. Škunca, F. Lassalle, and C. Dessimoz, "Inferring Horizontal Gene Transfer," PLOS Comput. Biol., vol. 11, no. 5, p. e1004095, May 2015, doi: 10.1371/journal.pcbi.1004095.

[7] National Center for Biotechnology Information, Isolates Browser. Bethesda (MD): National Library of Medicine (US), 2020.

[8] A. Boc, A. B. Diallo, and V. Makarenkov, "T-REX: a web server for inferring, validating and visualizing phylogenetic trees and networks," Nucleic Acids Res., vol. 40, no. Web Server issue, pp. W573–W579, Jul. 2012, doi: 10.1093/nar/gks485.

[9] D. H. Parks et al., "GenGIS: A geospatial information system for genomic data," Genome Res., vol. 19, no. 10, pp. 1896–1904, Jan. 2009, doi: 10.1101/gr.095612.109.

[10] S. Argimón et al., "Microreact: visualizing and sharing data for genomic epidemiology and phylogeography," Microb. Genomics, vol. 2, no. 11, Nov. 2016, doi: 10.1099/mgen.0.000093.

[11] A. Arleo, W. Didimo, G. Liotta, S. Miksch, and F. Montecchiani, "VAIM: Visual Analytics for Influence Maximization," ArXiv200808821 Cs, Aug. 2020, Accessed: Oct. 23, 2020. [Online]. Available: http://arxiv.org/abs/2008.08821.

[12] N. Henry and J. Fekete, "MatrixExplorer: a Dual-Representation System to Explore Social Networks," IEEE Trans. Vis. Comput. Graph., vol. 12, no. 5, pp. 677–684, Sep. 2006, doi: 10.1109/TVCG.2006.160.

[13] N. Henry and J.-D. Fekete, "MatLink: Enhanced Matrix Visualization for Analyzing Social Networks," in Human-Computer Interaction – INTERACT 2007, Berlin, Heidelberg, 2007, pp. 288–302, doi: 10.1007/978-3-540-74800-7_24.

[14] M. Boolchandani, A. W. D'Souza, and G. Dantas, "Sequencing-based methods and resources to study antimicrobial resistance," Nat. Rev. Genet., vol. 20, no. 6, Art. no. 6, Jun. 2019, doi: 10.1038/s41576-019-0108-4.

[15] EMBL-EBI, European Nucleotide Archive (ENA). 2020.

[16] E. W. Brown, N. Gonzalez-Escalona, R. Stones, R. Timme, and M. W. Allard, "The Rise of Genomics and the Promise of Whole Genome Sequencing for Understanding Microbial Foodborne Pathogens," in Foodborne Pathogens: Virulence Factors and Host Susceptibility, J. B. Gurtler, M. P. Doyle, and J. L. Kornacki, Eds. Cham: Springer International Publishing, 2017, pp. 333–351.

[17] O'Neill, Jim, "Tackling Drug-resistant Infections Globally: Final Report and Recommendations.," HM Government and the Wellcome Trust, London, 2016.

[18] G. H. Van domselaar, D. M. Patrick, and W. Hsiao, "Bioinformatics Tools to Improve Data Sharing and Re-use in Public Health - applications in antimicrobial resistance profiling and source tracking.," [Online]. Available: https://app.dimensions.ai/details/grant/grant.7638542.

[19] D. Feldroy, Cookiecutter Django. 2020.