ANTI-MICROBIAL PEPTIDE DATABASE

BY

ANUSHA SINGAMANENI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND

2022

MASTER OF SCIENCE THESIS IN COMPUTER SCIENCE

ANUSHA SINGAMANENI

APPROVED:

Thesis Committee:

Major Professor: Dr. Joan Peckham

(Major Professor's Name)

Dr. Lenore Martin

(Co-Major Professor's Name)

Dr. Abdeltawab Hendawi

(Internal Committee Member's Name)

Dr. Roberta King

(Outside Committee Member's Name)

Dr. Brenton DeBoef

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

(2022)

ABSTRACT

The goal of this thesis is to re-design and re-implement a functioning and efficient database with uniformity, coherence and most importantly transparency which clearly explains the system needs, methods and procedures openly to the users. The main users of the database are researchers, biologists, biochemists and engineers. We also developed a tool called MIC (the Minimal Inhibitory Concentration which is defined as the minimum concentration of the antimicrobial agent that will inhibit the visible growth of an organism after an overnight incubation (>24 hours)), for the users of the database by which, researchers can upload their lab data, perform the calculation for average peptides with timepoints and see results using graphical visualizations.

In addition, we developed a user-friendly web interface to support transparency. We revisited the normalization of the database, as this is an editable database and as it evolves, we need to add and delete attributes and change the database structure to assure consistency, correctness and efficiency. Finally, this work developed a secure web interface that assures the privacy of the users' data.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my Major advisor Dr. Joan Peckham and Co-major advisor Dr. Lenore Martin who made this work possible. Their expertise, guidance and advice carried me through all the stages of doing this project. I would also like to thank my internal committee member Dr. Abdeltawab Hendawi for his brilliant suggestions and comments.

I would also like to give my special thanks to my husband for his continuous support and understanding. Your prayers for me was what sustained me this far.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

PAGE

LIST OF FIGURES

FIGURES

PAGE

CHAPTER 1

Introduction

Since Penicillin was discovered and isolated in 1928 by Alexander Fleming, antibiotics have been the most important treatment to defeat bacterial infection. However, as time has passed, drug-resistance to many commonly used antibiotics has become a critical health issue of global concern. As per the CDC's (Centers for Disease Control and Prevention) AR (Antibiotic Resistance) threats in United States, 2019 report, "more than 2.8 million antibiotic-resistant infections occur in the U.S. each year, and more than 35,000 people die as a result." [1] In response, researchers are investigating alternative approaches for the treatment of ailments caused by antibiotic resistant microbes. One active area of investigation includes the use of peptides as an alternative to antibiotics.

We have designed and implemented a functioning and efficient database, the Anti-Microbial Peptide Database (AMPed) that supports the analysis of the effectiveness of various antimicrobial peptides on antibiotic resistant pathogens. This includes a database, a secure web interface, and algorithm to support the analysis and communication of experimental results. Central to this work is uniformity, coherence and most importantly transparency. By transparency, we mean that the data and system needs, methods and procedures are openly explained and clearly understood by the users of the system [2]. To achieve this, we programmed the algorithm, a tool called MIC (Minimum Inhibitory Concentration) for the users of the database to calculate and plot the average levels of microbes in response to peptides that are applied at various experimental timepoints. The experimental data is provided as input, and the algorithm produces informative output. We

also developed a user-friendly and secure web interface to support transparency and re-designed the database for correctness and consistency. Dr. Lenore Martin's AMPed research group at the University of Rhode Island (URI) previously developed a similar web application with backend MYSQL [3] database, and that is integrated with analytical algorithms (MIC, MassSpec) and that provides all valuable information about antimicrobial peptides in one place. However, missing were clear descriptions of suitable data formats for archival in the database and for input to the MIC algorithm, the design and provenance of the experimental data, and the appropriate use of the algorithms. To accomplish this, we provided a coherent and improved application with transparency including information about appropriate use of the system (http://amped.uri.edu/) We also redesigned the database for greater efficiency and correctness, and reprogrammed elements of the interface and MIC algorithm, with a special focus on providing transparency of the system to potential users.

The AMPed database has been developed and expanded for a number of years and is now a vast project that was initiated as a small class project by David Ryder [4], a biochemist, and Daniel Ducharme, a computer scientist, in CMB/CSC 522. George Konstantinidis [5] then designed the first general database structure for his MS thesis. After that Tripti Garg [6], enhanced this structure on her thesis project and created AMPed's first Web Interface. Following her, Abraham Herrera [7] worked on this project as his MS thesis with the help and guidance of Professors Martin and Peckham, and students in subsequent CMB/CSC 522 classes. He created a data uploading feature for the user of the website, and also enhanced the database and GUI (Graphical User Interface). The site is written in PHP scripts with MySQL commands improved with Cascading Style Sheets (CSS),

Asynchronous JavaScript and XML (AJAX), a client-side script that communicates to and from a server/database. In spite of all this work we still needed some tools for the websites which were relevant to the main users of the website, research biologists. So, with the help of Dr. Martin, Dr. Peckham (Computer Science at URI) and the students in the CMB/CSC 522 we were able to identify important tools, normalize the database, to reflect changes in the experimental data and processes, assure greater data correctness and consistency, and add more new features to the website, all with a focus on system transparency.

REFERENCES:

[1] U.S Centers for Disease Control and Prevention, Antibiotic Resistance Threats in United States, 2019, DOI - http://dx.doi.org/10.15620/cdc:82532, Publication date - November 2019, Number of pages: 140.

URL: https://www.cdc.gov/drugresistance/biggest-threats.html

[2] Joshi, M. & Bhardwaj, P. Impact of data transparency: Scientific publications. Perspectives Clin. Res 9, 31(2018).

URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5799949/

[3] Zoratti, I. Mysql security best practices. In 2006 IET accessed on 01/25/2022

URL: https://ieeexplore.ieee.org/abstract/document/4123759

[4] David E Ryder, "Synthesis and bioassay of novel hybrid antimicrobial peptides (AMPs), designed to increase therapeutic activity and decrease hemolysis" (2012). Dissertations and Master's Theses (Campus Access). Paper AAI1516162.

URL: https://digitalcommons.uri.edu/dissertations/AAI1516162

[5] George Konstantinidis, "Antimicrobial peptide editable database" (2015). Dissertations and Master's Theses (Campus Access). Paper AAI1586542.

URL: https://digitalcommons.uri.edu/dissertations/AAI1586542

[6] Tripti Garg, "Graphical user interface for antimicrobial peptide database" (2016). Dissertations and Master's Theses (Campus Access). Paper AAI10100281.

URL: https://digitalcommons.uri.edu/dissertations/AAI10100281

[7] Abraham Herrera, "Web based bioinformatics tool for contributing content to the Anti-Microbial Peptide Editable Database" (2016). Dissertations and Master's

Theses (Campus Access). Paper AAI10242575.

URL: https://digitalcommons.uri.edu/dissertations/AAI10242575

CHAPTER 2

The Structure of this Manuscript Style Thesis

This thesis contains one manuscript, a paper entitled “Antimicrobial Peptide Database”, that will be submitted to the Nature journal Scientific Data, which solicits manuscripts that describe archived bioscience data. This thesis starts with an introduction of the Antimicrobial peptide database that has been expanding through the work of Dr.Martin’s students, with a brief description of the previous work done. Then we discuss the major goals of this work, where we started and how the application developed over time. I show the AMPed flow diagram and then describe the database design, algorithm development, and enhancements done to the database as a part of this thesis work. I talk about database normalization [1] and finally explain about how we achieved transparency. The next chapter of this thesis covers our entire project starting with experiment description, how the input files are generated, the nature of the data uploaded and also how data records are imported. The paper then describes the MIC analysis algorithm and how these input files are used to produce results. This is followed by the database plan and implementation, a description of the GUI and details on how we validated our web application and database for correctness.

REFERENCES:

[1] Hillyer, M. An introduction to database normalization. MySQL AB (2003)

URL: https://users.dcc.uchile.cl/~mnmonsal/cc42a/guias/intronorm.pdf

CHAPTER 3

MIC Tool Development

The improved MIC algorithm was developed in the R programming language [1]. The flow diagram of the MIC R algorithm is shown in Appendix G. R scripts are programmed to analyze the outcomes of anti-microbial peptide experiments and communicate the output data to the user through plots displayed in the web page.

The previous work done on this tool accepts only CSV input files containing data from 96 well plates as shown in Figure 1. Users had to manually convert the "txt" file to CSV. This conversion was automated as part of this work; now the MIC tool takes the input files in the format that the experiment scanner provides. Sample input files are shown in Appendix A (Raw data file) and Appendix B (Timepoints file). To attain transparency, we provided user access to additional options in the result page where they can easily click a button to check the extracted data from the raw input files and the calculations done in producing plots using an R script. Sample output is discussed below and shown in Figure 13.

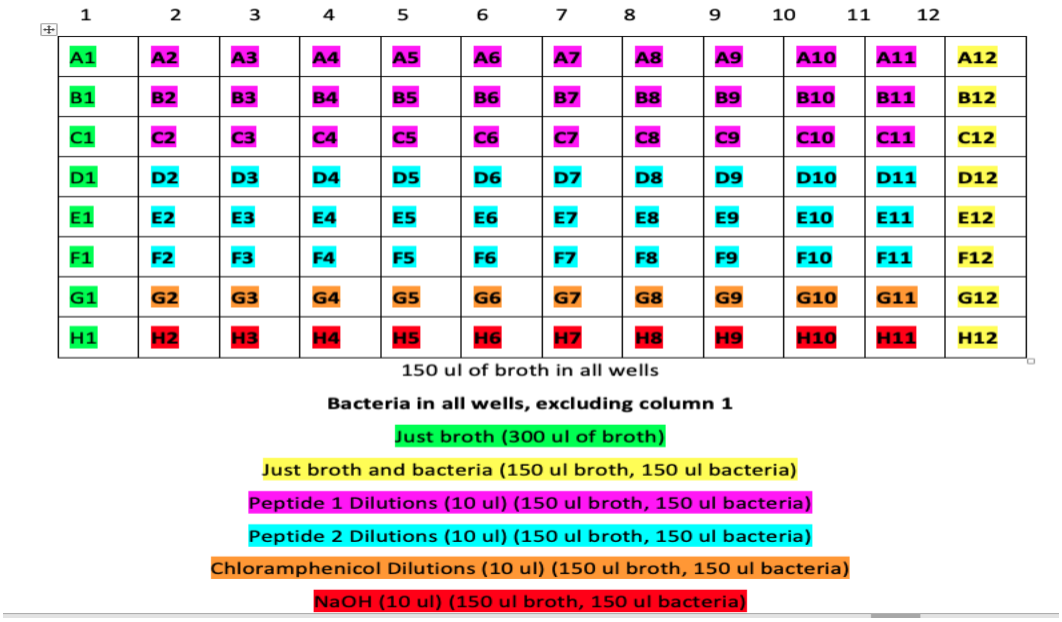


Figure 1: 96 well plates organization

The MIC (Minimal Inhibitory Concentration) [3] permits researchers to upload their lab data in the tools page as shown in Figure 2, calculate the average impact of peptides on microbial levels over various timepoints, and see results using graphical visualizations. The algorithm first prompts the user to upload unstructured raw data and raw timepoints taken from readings from their experiments, using text files.

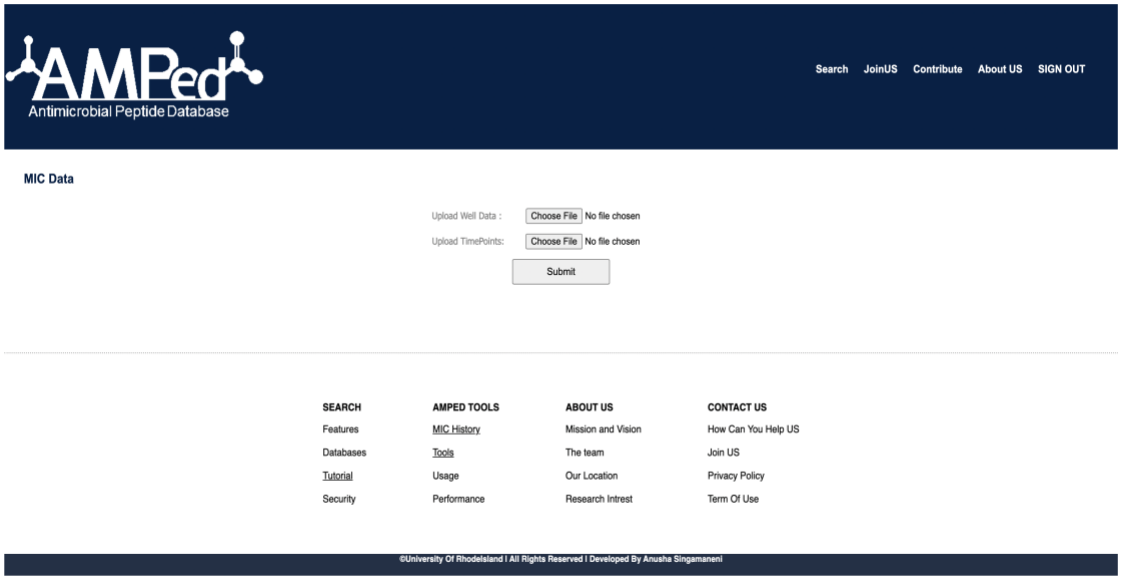


Figure 2: Tools page where users upload their input files

The algorithm retrieves only the data needed for the calculation from these input files and communicates the results of the analysis to the user using plots in JPEG format. Then the script identifies the position of bad wells mentioned by the user and eliminates them from all respective plates and calculates the mean and standard deviation of the microbial response to the peptides over the specified timepoints. The Sample code to identify and remove the bad wells is shown in Appendix F. The Sample mean data frame shown in Figure 3 is created with an input raw data file containing 18 plates (9 odd and 9 even). The first three rows (A2-A11, B2-B11, C2-C11) of every plate is considered as Peptide 1 (Pink

color shown in Figure 1). Peptide 1 of odd and even plates are bound together, and the column mean is calculated. X512, X256, X128,.., are the peptide concentrations

is the column mean of A2, B2, C2 of first odd and A2, B2, C2 of first even plates, X256 is the column mean of A3, B3, C3 of odd and even plates and so on., the data and time from the timepoints input file is also bound to the data frame.

""	"X512"	"X256"	"X128"	"X64"	"X32"	"X16"	"X8"	"X4"	"X2"	"X1"	"DATETIME"
"1"	0.0716666666666667	0.0718333333333333	0.0946666666666667	0.0708333333333333	0.0715	0.0718333333333333	0.072	0.0725	0.0703333333333333	0.0721666666666667	"12:15 2019/03/19"
"2"	0.0715	0.073	0.0728333333333333	0.0741666666666667	0.0745	0.0755	0.0751666666666667	0.0768333333333333	0.0746666666666667	0.0766666666666667	"12:45 2019/03/19"
"3"	0.0758333333333333	0.081	0.0853333333333333	0.0896666666666667	0.0918333333333333	0.0961666666666667	0.0973333333333333	0.0995	0.0968333333333333	0.0995	"13:15 2019/03/19"
"4"	0.0845	0.095	0.107333333333333	0.1145	0.118166666666667	0.125166666666667	0.126166666666667	0.1315	0.126833333333333	0.131833333333333	"13:48 2019/03/19"
"5"	0.104166666666667	0.142	0.159333333333333	0.171166666666667	0.176666666666667	0.186	0.186166666666667	0.1965	0.187	0.193333333333333	"14:43 2019/03/19"
"6"	0.1065	0.161666666666667	0.181666666666667	0.190666666666667	0.189166666666667	0.197666666666667	0.194833333333333	0.208	0.197833333333333	0.205166666666667	"15:15 2019/03/19"
"7"	0.1415	0.224833333333333	0.236166666666667	0.243333333333333	0.243833333333333	0.250833333333333	0.2425	0.260833333333333	0.2545	0.255666666666667	"17:10 2019/03/19"
"8"	0.439833333333333	0.4585	0.493333333333333	0.542833333333333	0.550666666666667	0.562	0.539166666666667	0.567833333333333	0.599333333333333	0.586333333333333	"8:16 2019/03/20"
"9"	0.534666666666667	0.556166666666667	0.556666666666667	0.610833333333333	0.605	0.7035	0.6065	0.711666666666667	0.688333333333333	0.619333333333333	"10:20 2019/03/20"

Figure 3: Sample mean data frame

To achieve transparency, the work of this thesis has added information to clearly explain the way in which the web interface and R algorithm are integrated into the MIC Tool, and the structure and flow of the data through the system. It is important for the user to understand how and in what form the data is to be input to ensure correct interpretation of experimental results. The R script requires that the data that comes from the web application for input to the algorithm is given by the user in a specific format for proper MIC analysis. This data consisting of information and timepoints regarding the plates used in the experiments is uploaded by the user into the web application, AMPed then saves the file in a specified location on the server, files shown in Figure 4, and provides arguments (file and MIC run number) for access to the data. The R script processes the data and explains to the user the result of the analysis of this data through graphical representation (See Figure 5).

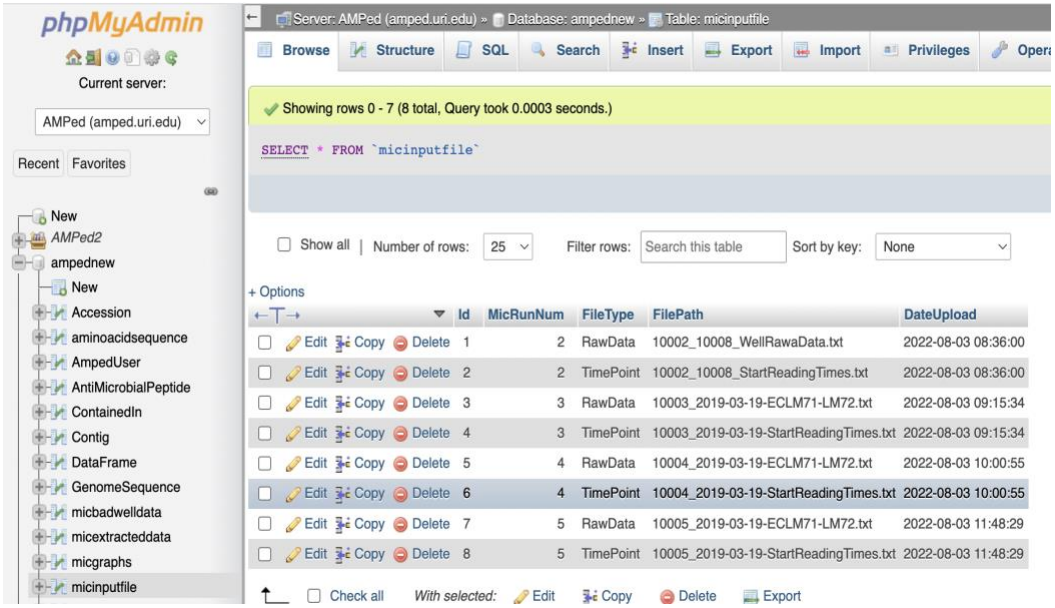


Figure 4: Input files stored in the database with MIC run number

The MIC algorithm analyzes and visualizes minimum inhibitory concentration (MIC) to predict the efficacy of AMPs [4] and accelerate the speed of analyzing the AMPs from complex natural and synthetic sources. Java initializes the R script (MIC) to securely process the data isolated from the application and produces the output. The AMPed application checks the address where the output plots will be stored in the database and displayed in the web page. We used Hibernate to map objects and relational properties to retrieve the data safely and correctly from the database and to prevent data security vulnerabilities like SQL injections [2]. Because the AMPed Java code explicitly uses SQL queries, usually a standard means to retrieve the data from the database, here we used Hibernate [5] queries which are essentially SQL embedded within the Java code to communicate with the database. This supports additional options on the website to show the user the data that is processed by the backend logic of the code.

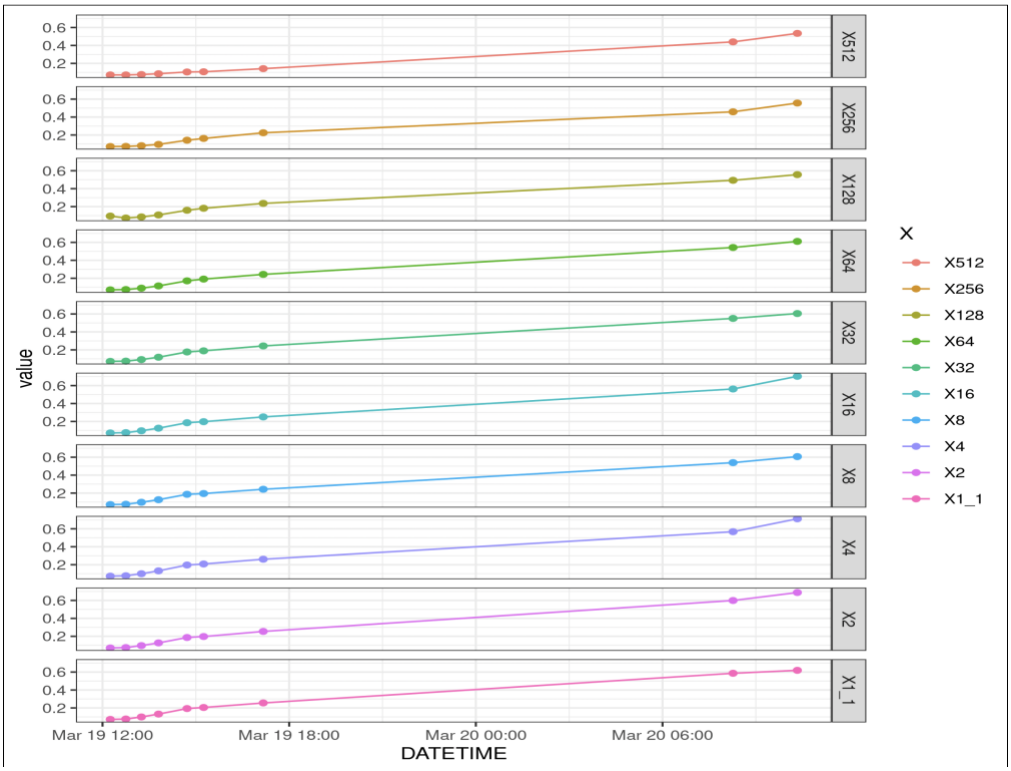


Figure 5: Sample growth plot for peptide 1 mean

REFERENCES:

[1] Gardener, M. Beginning R: The statistical programming language (John Wiley & Sons, 2012)

URL:https://books.google.com/books?hl=en&lr=&id=iJoKYSWCubEC&oi=fnd&pg=PR21&dq=r+programming+language&ots=5993FG7Bet&sig=bIdj5PmDT8tU5xdKdkQ6y3M_m5o#v=onepage&q=r%20programming%20language&f=false

[2] Janot, Etienne and Zavarsky, Pavol, Preventing SQL injections in online applications: Study, recommendations and java solution prototypr based on the SQL DOM, 2008.

URL: https://era.library.ualberta.ca/items/eced469a-75ac-4afc-94a7-24035ebfc516

[3] J. M. Andrews “Determination of minimum inhibitory concentrations.” J Antimicrob Chemother. 2001 Jul; 48 (Suppl 1): 5–16.

URL: https://www.ncbi.nlm.nih.gov/pubmed/11420333

[4] Wikipedia, Antimicrobial peptides, Last accessed 01/13/2022

URL: https://en.wikipedia.org/wiki/Antimicrobial_peptides

[5] Wikipedia, Hibernate (framework), Last accessed 02/01/2022

URL: https://en.wikipedia.org/wiki/Hibernate_(framework)

CHAPTER 4

Technical Overview

The web application is designed using a two-tier architecture, in which a client communicates directly with a server. The Java web application triggers the MIC R script by executing a Linux command in R (shown in Appendix E). The application build involves creation of multiple JSP pages. Each of these pages use HTML and CSS [1] to implement a simple interface and apply Core Java [2] to perform the logic that directly queries the database and inserts the retrieved data into the pages. The home page of the web application developed is shown in Figure 6.

A Flow chart of the R script can be found here:

http://amped.uri.edu/amped/micFlowChart.jsp (also in Appendix G)

Below is the list of software used on the server:

- R 3.5.2 → MIC algorithm developed using R programming language
- Java 1.8.0_332 → All core functionalities and web pages are developed using Java
- Apache Tomcat 9.0.37 → Apache Tomcat [3] is an open-source web server container used for implementation of java server pages.
- MySQL 5.5 → MySQL is a freely available open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).

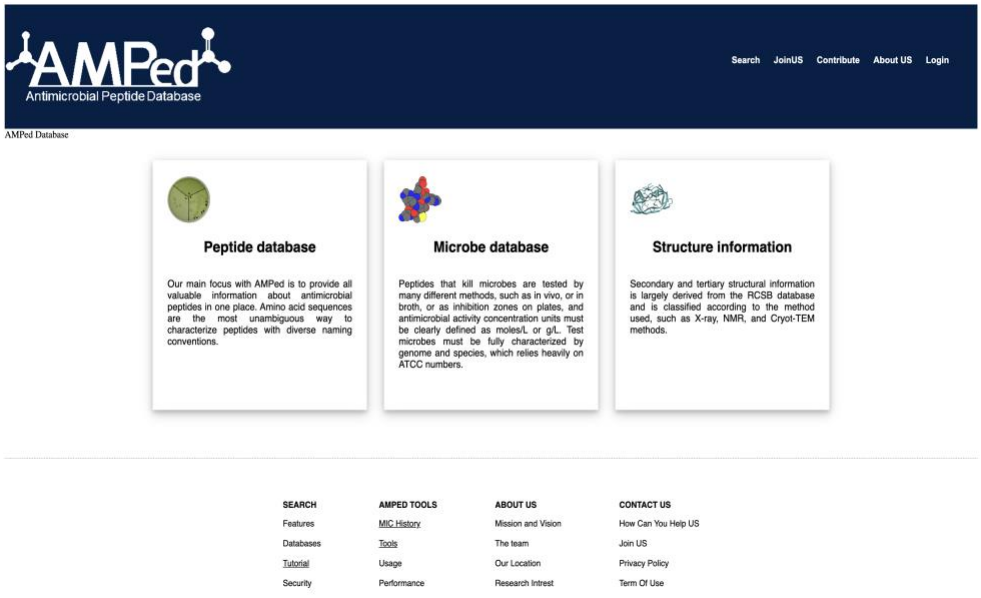


Figure 6: AMPed Home page

REFERENCES:

[1] Form content Type, Last accessed 01/20/2022

URL: http://www.w3.org/TR/html401/interact/forms.html%23h-17.13.4#h-17.13.4

[2] Horstmann, C. S. & Cornell, G. Core Java 2: Fundamentals, vol. 1 (Prentice Hall Professional, 2001).

URL: https://books.google.com/books?hl=en&lr=&id=W6bomXWB-TYC&oi=fnd&pg=PP1&dq=core+java&ots=4vJdWRwsqO&sig=bjkg-lvs1gBu9ja1pCclT9wNWXc#v=onepage&q=core%20java&f=false

[3] Wikipedia, Apache tomcat, Last accessed, 05/17/2022

URL: https://en.wikipedia.org/wiki/Apache_Tomcat

CHAPTER 5

AMPED Flow Diagram

AMPed [1] is a repository, so to access it we need an interface. We created a customized domain name mic.amped.uri.edu for the development of the application. Figure 7 shows the proposed flow diagram is shown below.

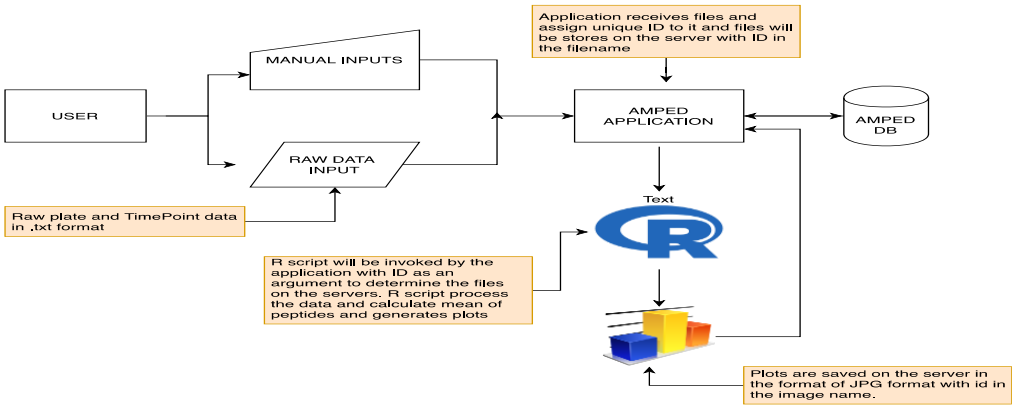


Figure 7: AMPed flow diagram

The user inputs two types of data, manual inputs (bacteria, assay, antibiotic concentration etc..,) and raw data. Manual inputs need to be entered in the text field of the web page and raw data is uploaded as text files. The AMPed Application receives these files and saves them in the server with a unique ID in the filename. The data in the input files and manual inputs are stored in the database (AMPed DB), and user logs are captured as logins and logouts. After storing and saving the user input data, the R script is invoked by the AMPed application and processes this data, generates the plots, sends the results in JPEG format to the application to display in the web page for the user. The analysis results are saved in the database as shown in Figure 8 below. This is a “MICGraphs” table in the database which stores the graphical results with “MICRunNumber” and date the results are generated.

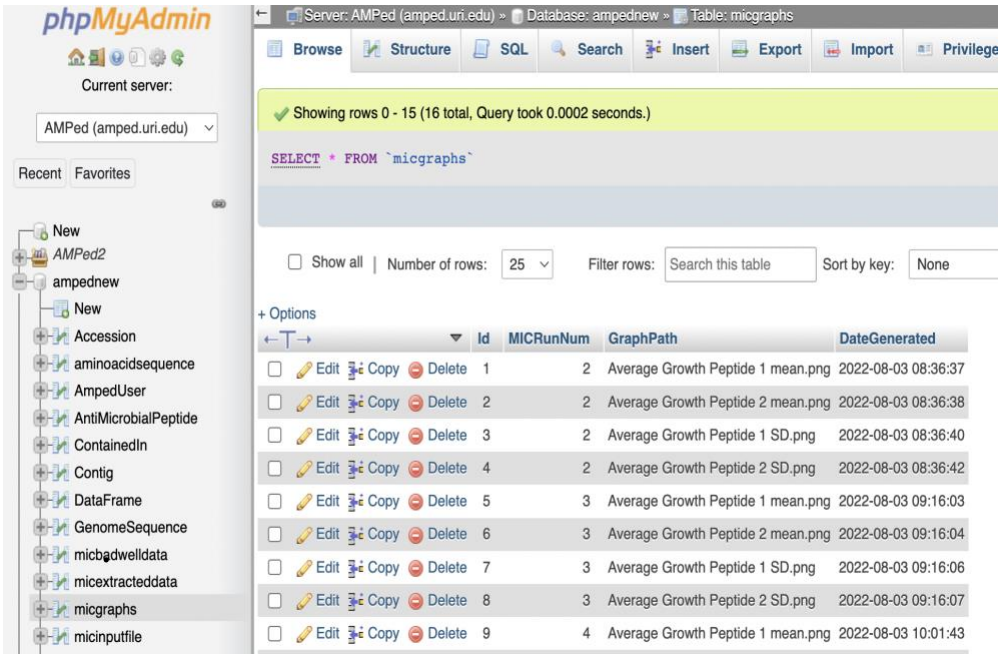


Figure 8: Results with their generated time stored in the backend database

This flow results in a clear understanding of the data, as the output of the algorithm shows all details about plates average, bad wells, means and standard deviation and also explains all the methods and procedures used to derive the output with required annotations in the plot.

REFERENCES:

[1] Lenore Martin - AMPed (Anti-microbial peptide database) - University of Rhode Island.

URL: https://amped.uri.edu

CHAPTER 6

AMPED Database

A database is a collection of information, or data that is organized in a manner so it can be easily and efficiently accessed, managed and updated. A Database Management System (DBMS) supports the design and development of database instances that persistently store, retrieve, define, and manage data in defined databases. AMPed uses a relational DBMS, so data is organized into rows, columns and tables. The database design process is very important to integrate all of the content and to assure that the data remains consistent and correct and can be accessed efficiently.

We analyzed the user requirements to help us identify the major components to be included in the database. Both biologists and computer scientists worked together in regular meetings for a semester to collect user requirements from many different user classes, figured out the connections among the experimental data produced in the lab, and determined the software tools that would need to be developed to reorganize the raw data into data frames. The conceptual data model was developed as an ERD (Entity-Relationship Diagram) that captures both accessed data and online data shared with other expert users. The team arrived at a database design that satisfies both biological and data science experts, as well as the best software design practices. The ERD (Entity Relationship Diagram) or conceptual model of Antimicrobial peptide database can be accessed at: http://amped.uri.edu/amped/erd.jsp

This logical representation of entities through the ERD, is mapped to database tables, the database schema, where the attributes become column names in the physical design of the

database. Some relationships are mapped as foreign key references from one table to another, and many to many relationships and/or relationships with attributes, are mapped to tables. The method of mapping uses standard database normalization theory (described in Chapter 7), and assures correctness and consistency of the data, and efficiency of data access.

The schema shown in Figure 9 defines the database tables. The schema also shows the relationships among the tables. The keys (unique identifiers of the rows of the table) and the foreign keys (references from one table to another used to combine information among tables) are also identified in the schema. The schema imposes integrity constraints that ensure that the changes made by authorized users do not result in loss of data and that information derived from multiple tables is correct and consistent.

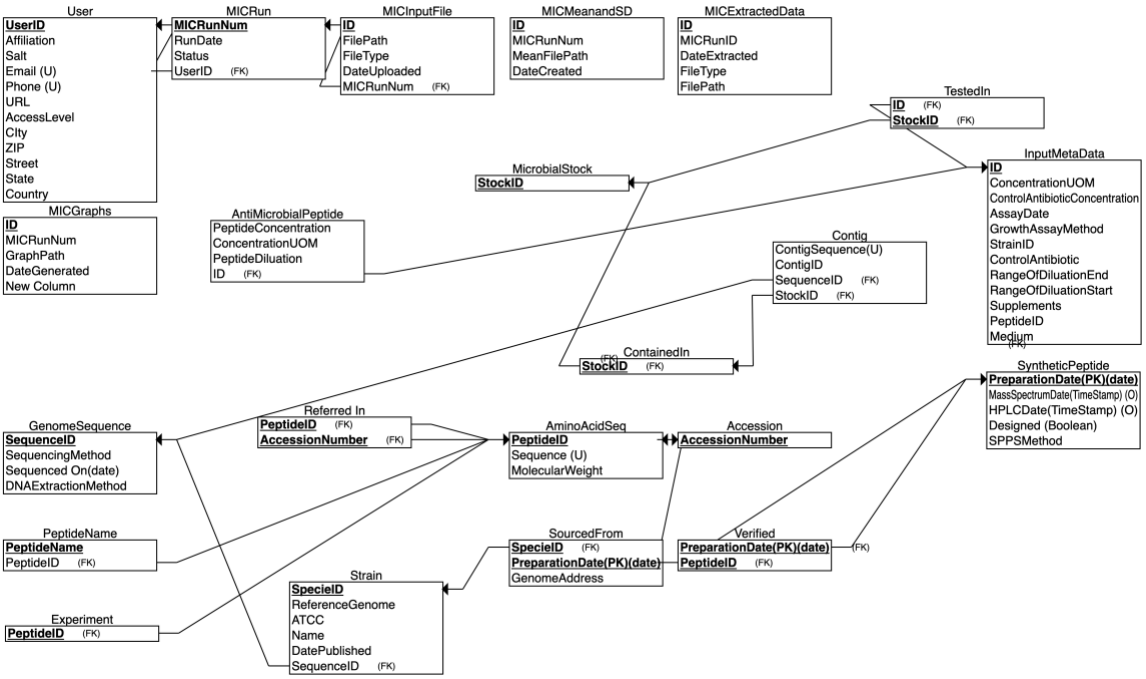


Figure 9: Schema diagram of AMPed database

The AMPed database was developed over time, and new attributes and tables were added as shown in Figure 10 below:

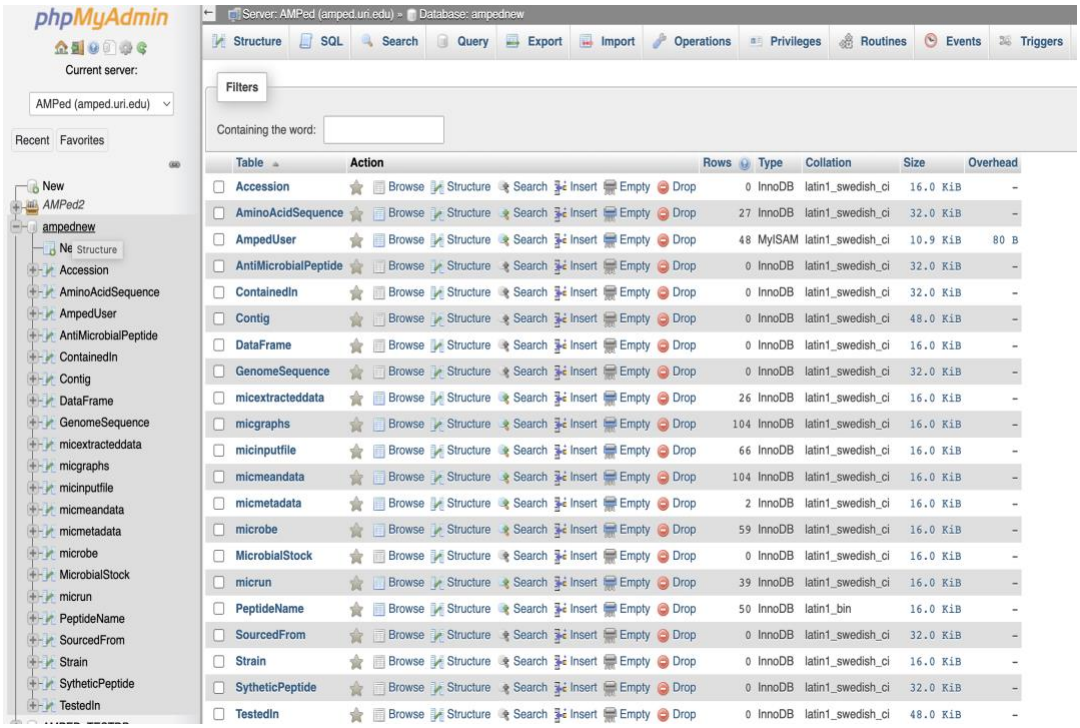


Figure 10: AMPed Tables

For example, “AminoAcidSequence” is a table in our database that was mapped from an ER Entity of the same name.

The requirements given from the biologists are

- This should contain the amino acid sequence describing a particular peptide.
- The sequences will be arbitrarily long:

To address these requirements,

We chose the VARCHAR[155] data type to represent the sequence. We maintained no duplicate sequences are added to the table by assigning Unique key to sequence attribute.

Attributes: PeptideID (PK), Sequence(U), MolecularWeight,

The Amped database is an editable database where values and attributes can be updated as we add information and need to make changes. However, the database design process is very important to integrate all of the content and to assure that the data remains consistent and correct and can be accessed efficiently, and that changes do not cause errors. So, each time the schema is updated, we need to revisit the design process, from ER diagram to specification of the schema. Addition or deletion of attributes and tables after a database is populated is usually possible if done with care. Other schema updates at this stage are difficult and may require removal and restructuring of all data, and then re-population of the database.

REFERENCES:

[1] Database, Last accessed: 02/01/2022

URL: https://www.techtarget.com/searchdatamanagement/definition/database

CHAPTER 7

Normalization

Normalization [1] is the process of mapping from the ERD [2] to database schema for organizing data stored in a database. This includes creating tables and relationships among tables, setting rules that define the relationships to protect the data and to make the database more flexible, correct and consistent by eliminating redundancy and inconsistent data

Additionally, we (biologists and computer scientists) verified our ERD with the following semantic checks and concluded that it is valid for our data.

- We have a central entity which maps to a table that participates in most relationships

Ex: Antimicrobialpeptide table, User table, InputMetaData table

- We ensured cohesivity, by identifying the weak entities or entities that need an owning entity to provide unique identification because they cannot be uniquely identified through their attribute values alone. We also checked that relationships among the entities and tables are necessary for the functionality of the database, as they represent the connections needed for combining data in response to queries upon the database.

- We also checked that the internal and external transactions among the tables are carried out properly and can properly combine information from related tables to provide needed output.

For example: The tables created to input MIC input files were split into multiple tables.

Every MIC run needs an input and transforms the input file into 96 rows data frame which is further used for calculations and produces mean, SD and growth plots which are stored in multiple tables.

1. MICRun Table shown in Figure 11 stores MICRunNumber (starts from 10000 and auto increments) which is a primary key and records can be uniquely identified.
2. MICInputFile Table shown in Figure 12 stores input files. This table has an attribute "MICRunNumber" which is a foreign key reference from the MICRun table.

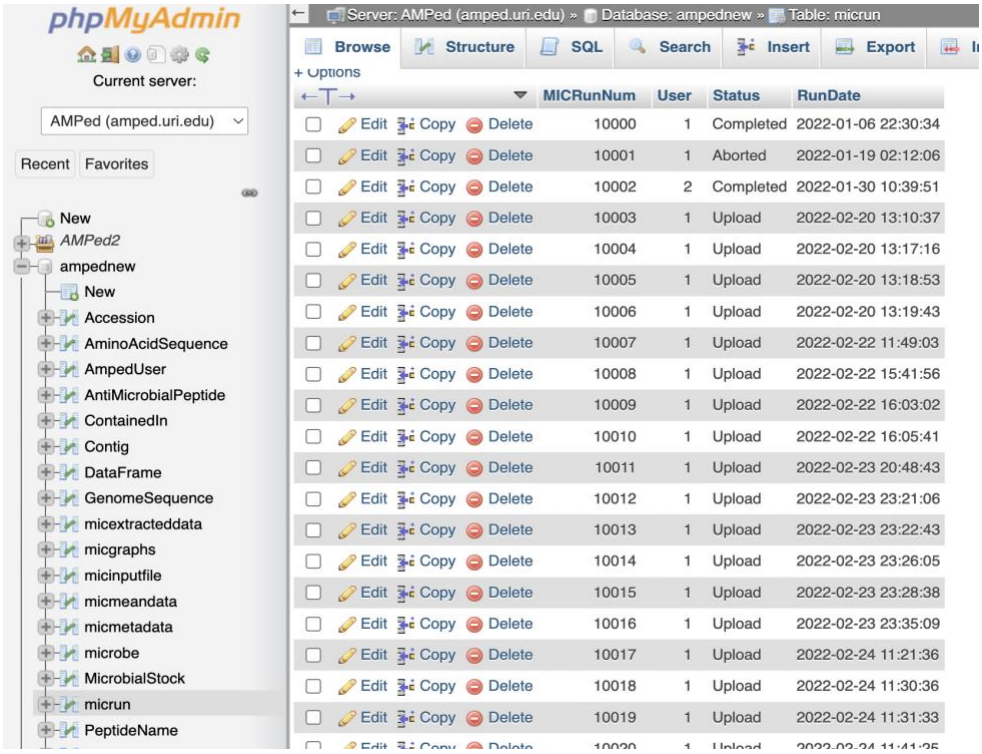


Figure 11: micrun Table

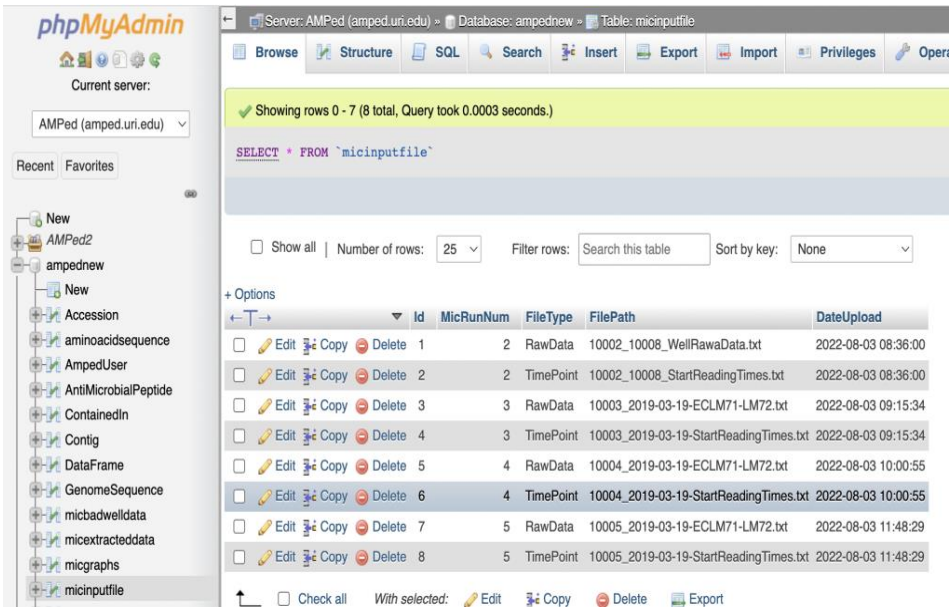


Figure 12: MICInputFile Table

REFERENCES:

[1] Hillyer, M. An introduction to database normalization. MySQL AB (2003).

URL: https://users.dcc.uchile.cl/~mnmonsal/cc42a/guias/intronorm.pdf

[2] SmartDraw (1994 – 2020) – Entity Relationship diagram, Last accessed October 2021

URL: https://www.smartdraw.com/entity-relationship-diagram/

CHAPTER 8

Transparency

Data transparency [1] has become an important aspect of research that helps in enabling evidence-based decisions and helps increase trust in the users of the data. So, one of our primary goals is to achieve data transparency, which means presenting data to the user in a format that its easy to see and easy to use without hiding anything in the shadow, and explaining our methods and procedures on how the raw data that is coming from the experiment scanner is transformed to get the result. We are successful in achieving this goal by showing the user step by step transformation of the data through our separate webpages. For example, once the user uploads their input ".txt" files from the tools page, the MIC algorithms runs and produces growth plots. Then a new record will be created in the MIC history page (shown in Figure 13) with a new MIC run number with run date and time. This MIC run number contains all the details about input files, and the transformed file (shown in Appendix C) needed for the algorithm calculations, final mean, SD and growth plots. The figure below shows the MIC History page which is created for the user to check their inputs and corresponding outputs

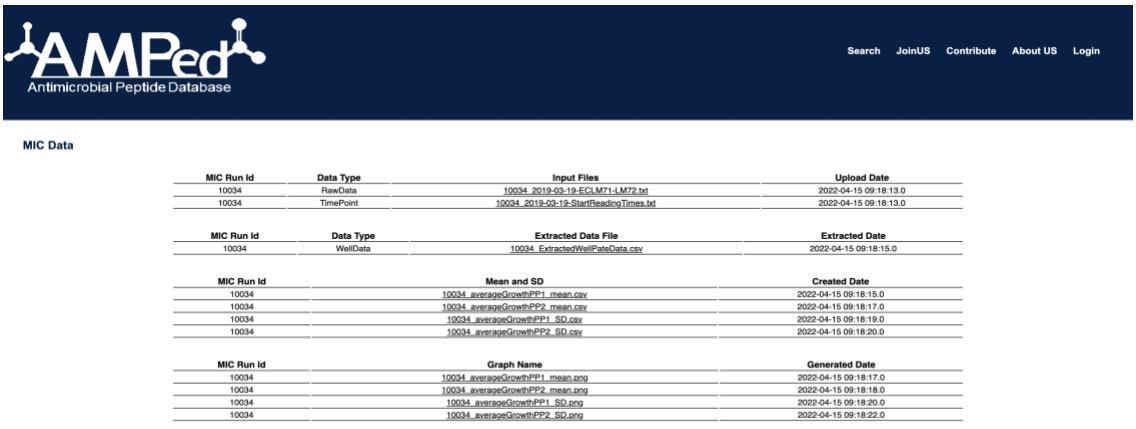


Figure 13: MIC History page

REFERENCES:

[1] Callaghan, Sarah, Editorial On the Importance of Data Transparency, Patterns, 1, Cell Press Open Access, July 2020.

URL: https://doi.org/10.1016/j.patter.2020.100070.

[2] Harvard Business Review. by Timothy Morey - A version of this article appeared in the May 2015 issue.

URL: https://hbr.org/2015/05/customer-data-designing-for-transparency-and-trust

CHAPTER 9

The Paper Manuscript

Anusha Singamaneni, Joan Peckham, and Lenore M. Martin

- Graduate Programs in Computer Science and Cell and Molecular Biology, University of Rhode Island, Kingston, RI, 02881, USA
- Department of Cell and Molecular Biology, University of Rhode Island, Kingston, RI, 02881, USA
- Department of Computer Science and Statistics, University of Rhode Island, Kingston, RI, 02881, USA

ABSTRACT

The Antimicrobial Peptide Database (AMPed.uri.edu) is a publicly accessible bioinformatics database that specializes in organizing and sharing raw antimicrobial peptide datasets for an interdisciplinary group of users that include Biologists, Biochemists, Engineers, and Computational Biologists. AMPed facilitates exchange of diverse but connected datasets such as amino acid sequences, bacterial genome sequencing results, and the results of biological and chemical assays. It also preserves metadata about the experimental conditions, such as details about the synthesis of peptides and the origins of the microbes these AMPs are being screened against, for example strain types, genomic sequences, and where to acquire identical samples. As a public database, it allows for contributions by several registered labs with demonstrated expertise, but more importantly, it also allows students and mature scientists alike to access large

amounts of valuable raw data about AMPs for bioinformatics experiments. The goal of this paper is to describe how the antimicrobial peptide data is originated, analyzed, transformed and presented; specifically describing the behavior and characteristics of the data derived from such experiments, and how it transparently mapped to a useful data schema for easy access. Central to this work is uniformity, coherence and most importantly interdisciplinary transparency [1]. By transparency, we mean that the data and the system needs, methods and procedures are openly explained and clearly understood by the diverse users of the system [2]. To attain this, included are the following: A tool called MIC (Minimal Inhibitory Concentration), which can be constantly reprogrammed and augmented as experimental design changes unfold, and cope with the reality that data formats output by the data recorders used in experiments always evolve over time. The tool is essential for users of the database who need to accurately calculate the peptide concentrations where microbial growth is inhibited, or microbes are killed, or who wish to evaluate the microbial growth rate changes as a function of time following antimicrobial peptide (or other drug) treatment. To maintain interdisciplinary transparency, we needed to provide a clear explanation of the transformation of raw experimental data into reported MIC values and design the algorithm to reflect current trends in experimental design and data analysis for bacterial growth datasets.

Introduction

Since Penicillin was discovered and isolated in 1928 by Alexander Fleming, antibiotics have been the most important drug to defeat bacterial infection.

However, over time drug-resistance to many commonly used antibiotics has become a critical health issue of global concern. This leads to a growing need to develop new types of drugs that can inhibit or kill antibiotic-resistant bacterial pathogens. In recent years scientists began using antimicrobial peptides (AMPs) as a solution to the antibiotic-resistance issue. AMPs are short proteins that are naturally secreted by many organisms as part of the host defense system against infection. They show great potential as a solution due to their wide number of potential combinations of 24 amino acids which is approximately 19^{24} or 4×10^{30} combinations. Acquiring large amounts of data is essential for further development of AMPs as a treatment for antibiotic-resistant bacteria. Scholars doing research in the field of antimicrobial peptides use many antibacterial assays. However, the results obtained are often not reproducible but the peptides can be used with different pathogens simultaneously on duplicate 96-well plates.

This interdisciplinary effort by computer scientists and biologists to understand how cells process information may yield new insights for both fields. If the computing scholars understand how the raw data has been generated by the experiment, they can help to identify the key characteristics for designing and implementing an algorithm that converts the raw data into required output. If the interdisciplinary communication is effective, the output data should make sense to the biologists. Attending to all stages of data collection, analysis, and communication with transparency, also addresses the ethics of AI [3]. The Minimal Inhibitory Concentration (MIC) is defined as the minimum concentration of the antimicrobial agent that will inhibit the visible growth of an organism after an

overnight incubation (>24 hours). Dr.Martin's lab designed an assay in such a way that all four organisms under study could be tested on the same day, in several 96 well plates simultaneously. In this way the effects of environmental factors can be avoided, which can affect the growth of the organisms and thus the results of MIC's. The next sections of the paper describe how the experiment was conducted and received the scanned raw data files, how the files are prepared as input to the algorithm, and how the algorithm produces the output. Figure 1 Below shows the flow diagram for AMPed

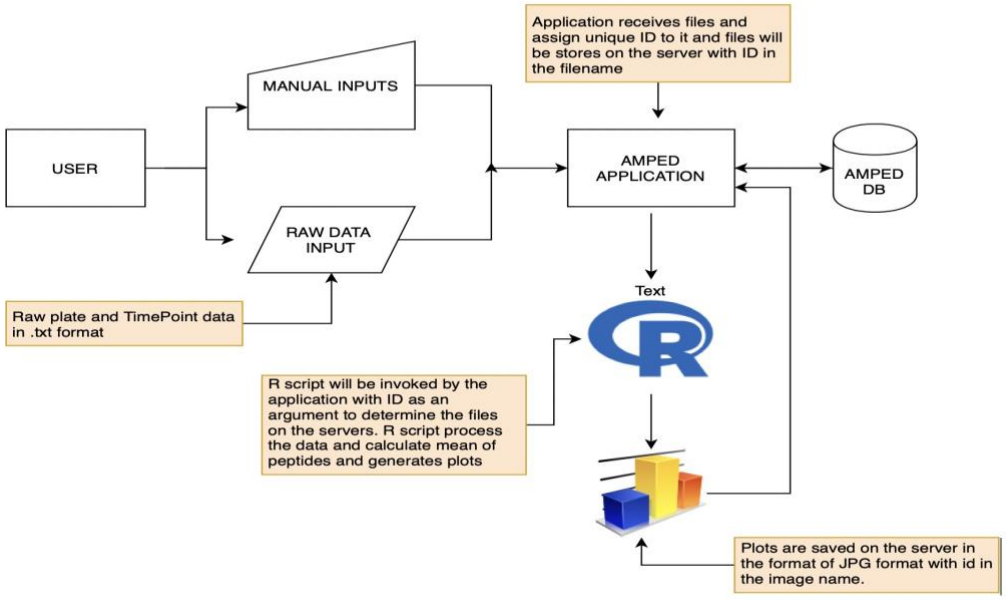


Figure 1. AMPed Flow diagram

Methods

0.1 Data Sources

Experiment Description:

The experimental methods used to capture bacterial growth data are described below:

Bacterial growth data that is to be used to calculate the MIC/MLC (Minimum Lethal Concentration) values is obtained as raw csv-formatted output from bacterial plate scanners that are used in the lab. An example of proprietary scanner software commonly used in biotechnology is SoftMax Pro. To facilitate interoperability and ensure future data access, most labs output a copy of all data in non-binary formats for archiving. In Dr. Martin's Lab a standard, optimized 96-well plate format was identified that facilitates consistent data analysis from most bacterial growth experiments. In this experimental format, the wells are set up according to a specific pattern containing either plain bacteria, bacteria plus standard antibiotics, or bacteria plus test peptides in ascending or descending concentration order. To develop data collection and analysis algorithms, we have set up our data cleaning process to incorporate this 96-well plate template. Figure 2 below shows how the 96-well plates are arranged.

In the 96-well plate template shown in Figure 2, purple represents wells containing test Peptide 1 in descending concentrations, similarly blue represents test Peptide 2, and orange wells contain an antibiotic of known MIC, in descending concentrations. Green wells contain broth only with no bacteria, and yellow represents broth containing only untreated bacteria. At the beginning of the peptide dilutions, the initial starting concentrations of the peptides can be changed

according to the needs of the experiment. It depends upon what type of lethality is necessary for the biological investigation and how potent the researchers are expecting a test peptide to be. The overall pattern of serial dilutions should be maintained, however, to ensure a range of concentrations is adequately sampled.

Each set of peptide dilutions are tested in triplicate, so the first 3 rows of the plate will all have the same peptide dilutions of Peptide 1 as shown in Figure 2. Rows 4-6 will contain dilutions of a second test peptide, for example Peptide 2, set up analogous to the first 3 rows with Peptide1. Tests are repeated in triplicate for each peptide to see what kind of changes the dilution brings to each peptide, and the two peptides are compared with the standard antibiotic in Row 7. The last row, Row 8, will have NaOH as a positive lethal control. Columns 2-11 will contain varying dilutions of test peptides. Column 12 (yellow) will be bacteria and broth only as the negative control (no peptide/antibiotic added).

A single colony of each different bacterial test species is grown overnight in a shaking culture in Mueller-Hinton (MH) broth. Optical density (OD) measurements at 630 nm, taken the next day, are used to calculate the stock bacterial CFUs (Colony forming Units). Bacterial stock cultures are then diluted in MH broth to give a target colony-forming unit (CFU) concentrations of 2 − 7x105CFU/ml 100ul of each bacterial suspension is needed for each well of the test plate, to be added to rows A-H and from columns 2 to 12. Column 1, Rows A-H are reserved as a sterility control and serves as a blank column for monitoring bacterial growth in the plate

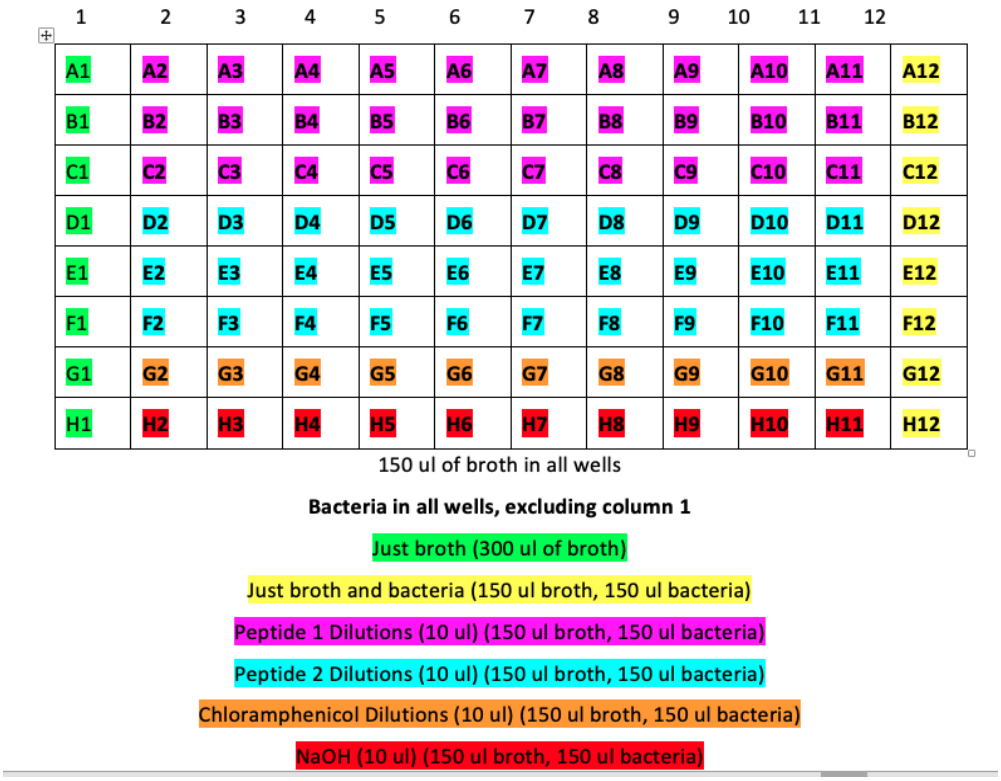


Figure 2. 96 Well Plates Organization

scanner by OD at 630 nm. Column 12 will be used as a control to monitor bacterial growth alone, with no peptide or antibiotics added. Stock solutions of the test peptides are made by weighing out accurate amounts of lyophilized purified peptides to give 10X, if possible, of the maximum required peptide test concentrations and dissolved into "peptide dilution buffer" (0.01v/v acetic acid, 0.2w/v BSA), glacial acetic acid, or DMSO using polypropylene or glass tubes. Ten serial dilutions of each test peptide are then prepared by serial 500 uL transfers into 10 micro-centrifuge tubes, each containing 500 uL of the peptide dilution buffer, vortexed, and spun down before transferring along to the next tube. Ten micro-liters of individual test peptide dilution is added sequentially (in triplicate, rows A-C or D-F) into each well of the test plate, from columns 2 to 11. A robot pipettor is used to aliquot peptide and antibiotic dilutions to the 8 96-well plates needed for each assay of 2 peptides against 4 bacterial species. Plates with peptide and

bacteria added are incubated at 37 degrees Celsius for 18-24 hours, and OD630 data is obtained by repeated scanning at time points using an automated scanner and saved as text files, and the time points are recorded when each scan is performed. At the end of each experimental run, bacterial CFUs are determined by plating out 10ul of 3 serial dilutions of the bacterial cultures from plate wells at the end of the experiment that did not show cloudiness, spread onto agar plates and incubated at 37 degrees Celsius overnight to confirm killing by viability counts of any wells that showed growth inhibition.

0.2 Tools (MIC):

The MIC (Minimal Inhibitory Concentration) [4] is an online tool to which researchers can upload their lab data through the AMPed portal website; it performs growth curve and MIC calculations for their peptides (See Section 0.4 for detailed description) and users will see results as bacterial growth curves using graphical visualizations. The algorithm first prompts the user to upload the txt-formatted raw data and a separate text file containing raw timepoints reflecting the dates and times of readings from their experiments represented by the raw data files; see Figure 3.

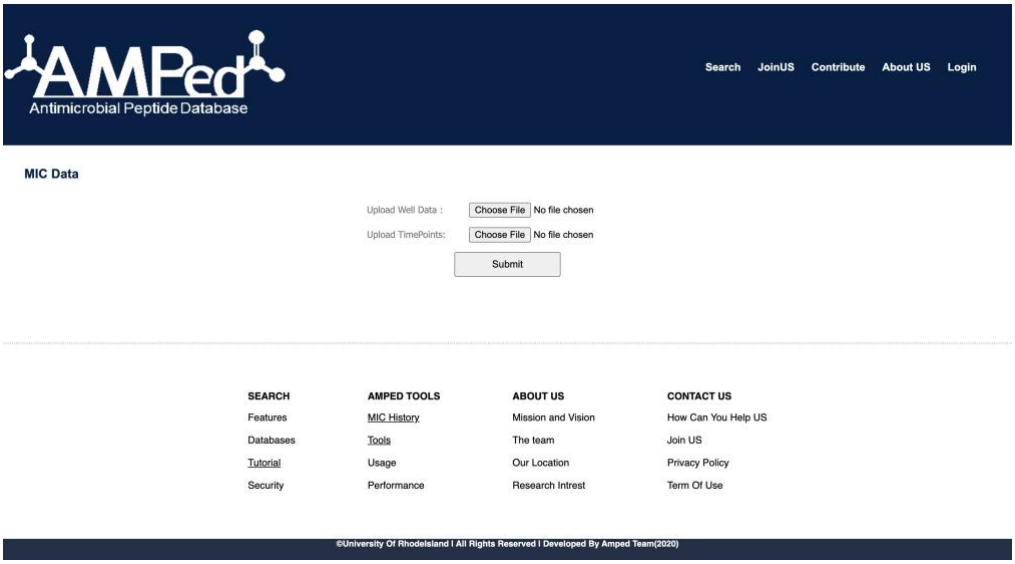


Figure 3. MIC Experiment Plan: Set-Up and Organization of inputs

The algorithm extracts and cleans the numerical data needed for the bacterial growth calculations from these raw data input files, communicates the results of the analysis to the user through plots in JPEG format, and archives the data in a unified format. We have further developed this tool, so that it takes as input, two text files, cleans and retrieves the required data and searches for bad wells, if any exist, for exclusion from data analysis. This script eliminates the bad wells from all respective plates (Sample code shown below), calculates the means and standard deviations of the timepoints of respective peptides and also plots the 24 hours growth curve versus concentrations.

The R [5] code below first asks the user to enter the bad well position and if its odd or even. Then it reads the position from the txt file using “read.csv” function

```
well <−  readline ( prompt =" Enter  Excluded  Well ID :  " )
if ( well ! =  0 ){
odd_even <−  readline ( prompt =" Enter  odd / even :  " )
```

```
well _ position <− read . csv ( " Well _ positions . csv " )
m1 <− as . matrix ( well _ position )
D1 <− which ( m1== well ,  arr.ind =TRUE)
ROWNum <−D1 [ , 1 ]
ColNum <− D1 [ , 2 ]
exlValue <− as . numeric ( ’NA’ )
oddPlates [ [ 1 ] ] [ ROWNum, ColNum ]
length ( oddPlates )
if ( odd_even == " odd " ) {
for ( i  i n  1 : length ( odd Plates ) ) {
oddPlates [ [ i ] ] [ ROWNum, ColNum ] <− exlValue
}
} else {
for ( i  in  1 : length ( evenPlates ) ) {
evenPlates [ [ i ] ] [ ROWNum, ColNum ] <− exlValue
}
}
}
```

Data Records:

0.3 Uploaded Data:

In Figure 1, only one block is shown. There can be many blocks like this depending upon the experiment. Each block represents one single time point in the growth

curve of the bacteria. The antibiotic represented in Row 7 will have a known antibiotic at the same dilutions as the test peptides. The data we input to the open-source coded algorithm is the "raw" data contained in a text or XML file that is exported from the scanner by the manufacturer's proprietary software. We incorporate minimal changes into the "raw" format of the data as originally collected, because incorporating "metadata" describing experimental details prior to uploading is essential for proper interpretation of results.

0.4 Data Processing Tools: R scripts

The MIC and MLC calculators determine the minimum concentration of a given AMP that will either be inhibitory or lethal against some specified species of bacteria. The system for the 96 well plate (shown in Figure 1) project is based on the assumption that 96 well plates are being used, with two different peptides per plate in a standard sequence. It will compare two different peptides and one known antibiotic on one known bacteria. It assumes that Peptide 1 is found in rows A, B, and C and Peptide 2 is found in rows D, E, and F. There are two sets of plates, so starting from the beginning of the file to the end, odd numbered plates go together, and even numbered plates go together. The plates are duplicates so they will contain the same peptides. Once all of the average values for the peptides are calculated along with the timepoints a growth curve can be plotted for each peptide (rows A-C and D-F) and each concentration (columns 2-11). The Script written as part of the thesis work is in R programming which collects data and timepoint txt. files as an input (96 well plate data) and miscellaneous data points from user via a web

interface, and exclude well number, even/odd, (in case any bad well is found). The backend then executes an algorithm written in R script to calculate the MIC and returns it to a different web interface and displays graphical results using JavaScript. If the user selects the MIC calculator, they will be asked a few details about the test peptides and the control antibiotic. The R script receives two types of input files, data and times in text input files and manual user inputs. The required data from the data text input file is retrieved and plate data is extracted into a data frame. In the same way, the experimental timepoints are also extracted into a data frame from a text file. Then we calculate the number of plates and check for bad wells in all the plates. If a bad well is found, we delete it from the respective plate. Once all the plates are free from bad wells, we split the plates and timepoints into even and odd. Then we bind peptide1 of odd and even plates together and Peptide 2 of odd and even plates together and calculate the mean and standard deviation of combined odd-even plates of Peptide1 and combined odd-even plates of Peptide 2. The code also calculates the average of antibiotic (Row 7) and bacteria and broth (Column 12) of even and odd plates. Then it merges the timepoints with the combined means of Peptide 1 and Peptide 2. So, now we have a complete data frame with the averages and timepoints that are required to plot a graph and save the graph in JPEG format.

Database Plan: Schema of Data entity relationships

A database is a collection of information, data that is organized in a manner so it can be easily and correctly accessed, managed and updated. A Database Management System

(DBMS) supports the design and development of database instances, and persistently stores, retrieves, defines, and manages data in defined databases. AMPed uses a relational DBMS, so data is organized into rows, columns and tables. The database design process is very important to integrate all of the content and to assure that the data remains consistent and correct and can be accessed efficiently.

We analyzed the user requirements which helped us in identifying the major components to be included in the database. Both biologists and computer scientists worked together in regular meetings for a semester to collect user requirements from many different user groups, figured out the connections among the experimental data produced in the lab, and determined the software tools that would need to be developed to reorganize the raw data into data frames. The conceptual scheme was developed as an ERD (Entity-Relationship Diagram) that also addresses the needs for data sharing online through the database with other expert users. The team arrived at a database design that satisfies both Biological and Data Science-Software design disciplinary perspectives.

0.5 Database design : The ERD

The ERD [6] is a graphical representation of the data entities and their relationships to each other. The ERD can be accessed using this URL: http://amped.uri.edu/amped/erd.jsp. The purpose of ERDs is to conceptualize the data model, and work with the peptide researchers (clients) to fully explore the relationships among data entities. The ERD is then mapped to the database schema using relational tables to structure the data in a way that preserves data correctness and efficiency. Figure 4 shows the relational tables used to develop the database schema.

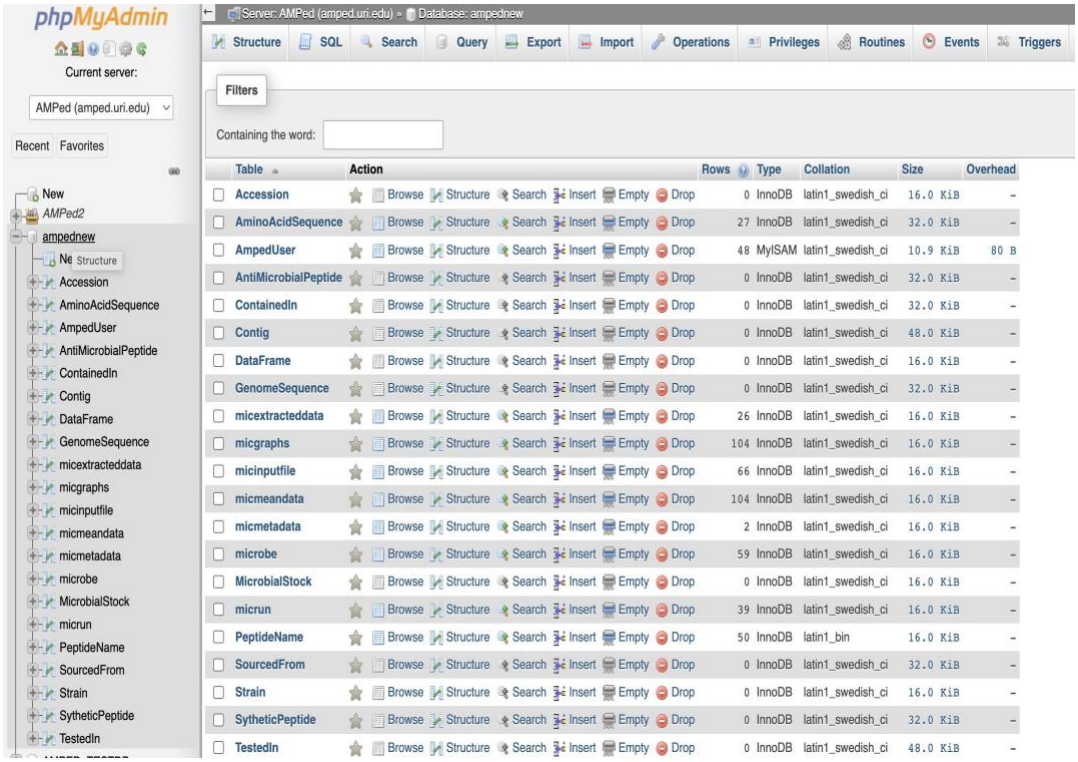


Figure 4. AMPed Tables

0.6 Database implementation Tables

The logical representation of ERD entities is mapped to database tables, where the attributes become column names in the physical design of the database. Some relationships are mapped as foreign key references from one table to another, and in case of many to many relationships and/or relationships with attributes, they are mapped as tables.

The schema defines the database tables. The fields (attributes) in each table show the data types of the associated data. The schema also shows the relationships between the tables. The keys (unique identifiers of the rows of the table) and the foreign keys

(references from one table to another, used to combine information among tables) are also identified in the schema. The schema imposes integrity constraints that ensure that the changes made by authorized users do not result in loss of data and that information derived from multiple tables is correct and consistent.

The AMPed database [7] has been developed and changed over time, and new attributes and tables have been added as needed. For example, Figure 5 shows "AminoAcidSequence", a table in our database.

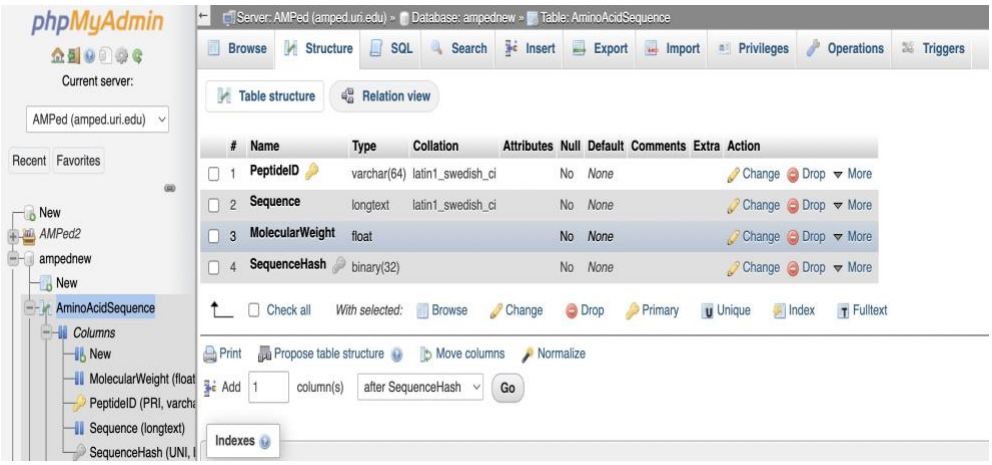


Figure 5. Amino Acid Sequence Table

The requirements given from the biologists are:

1) This table should contain the amino acid sequence describing a particular peptide, and

2) The sequences will be arbitrarily long. To address these requirements we choose the LONGTEXT data type to represent the sequence. However, we need to maintain that no duplicate sequences are added to the table, made it as unique ID.

The table's attributes are: PeptideID (PK), Sequence (U), MolecularWeight .

(Synopsis of SQL Query shown below). Because the AMPed database is editable in that tables and attributes can be updated as we need to make changes. Standard database design processes, including the use of ER Diagrams, and standard mappings to database schema are strictly followed during these changes to integrate all of the content and to assure that the data remains consistent and correct and can be accessed efficiently.
The below SQL [8] query describes the creation of Amino AcidSequence table with attributes as peptideID, Sequence, Molecular Weight. PeptideID is the primary key and Sequence is the Unique ID of the table.

```
CREATE TABLE AminoAcidSequence
    (
    PeptideID VARCHAR(64) NOT NULL,
    Sequence LONGTEXT NOT NULL,
    MolecularWeight FLOAT NOT NULL,
   PRIMARY KEY (PeptideID),
   UNIQUE (Sequence)
    );
```

0.7 Database implementation: Normalization

Normalization [9] is the process of organizing data stored in a database to maintain data integrity. This includes creating tables and relationships among tables, setting rules that define the relationships to protect the data and to make the database more flexible, correct and consistent by eliminating redundancy and inconsistent data We (biologists and computer scientists) verified our ERD with the following checks and concluded that it is valid for our data and then used standard normalization processes to map the

conceptual model (the ERD) to the relational database schema.

- Normalization often requires the decomposition of entities into tables to attend the dependencies among the attributes that could lead to inefficiencies and incorrectness when the data is later accessed [10].
- Normalization was accomplished by mapping the ERD properly to database tables with appropriate key and foreign key attributes. For example: Following normalization rules, the tables created to input MIC input files were split into multiple tables. Every MIC run needs an input and transforms the input file into a 96-row data frame which is further used for calculations and produce mean, SD (Standard Deviation) and growth plots which are stored in multiple tables
- The MICRun Table stores MICRunNumber (starts from 10000 and auto increments) which is a primary key and which assures that records can be uniquely identified.
- The MICInputFile Table stores input files. This table has a attribute "MICRunNumber" which is a foreign key reference from the MICRun table, thus providing a connection between the input file and the appropriate MIC run.

Figure 6 show the MICRun table, that stores the MICRun activity run by user. When the user uploads the input file the application assign a unique number (started with 10000 and auto increments) to the " MICRunNum" attribute which is stored in this table along with the user ID and also stores the status, date and time of the MIC Run

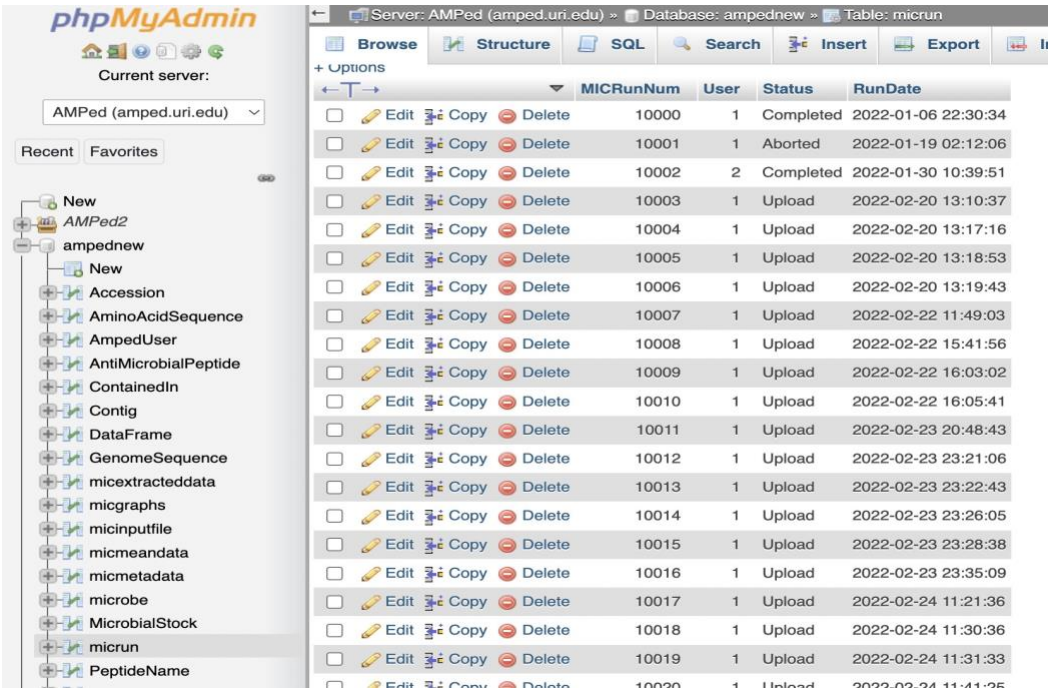


Figure 6. micrun Table

Figure 7 shows micinputfile table that stores the raw input data files uploaded to the MIC algorithm. This table has a foreign key reference from the micrun table

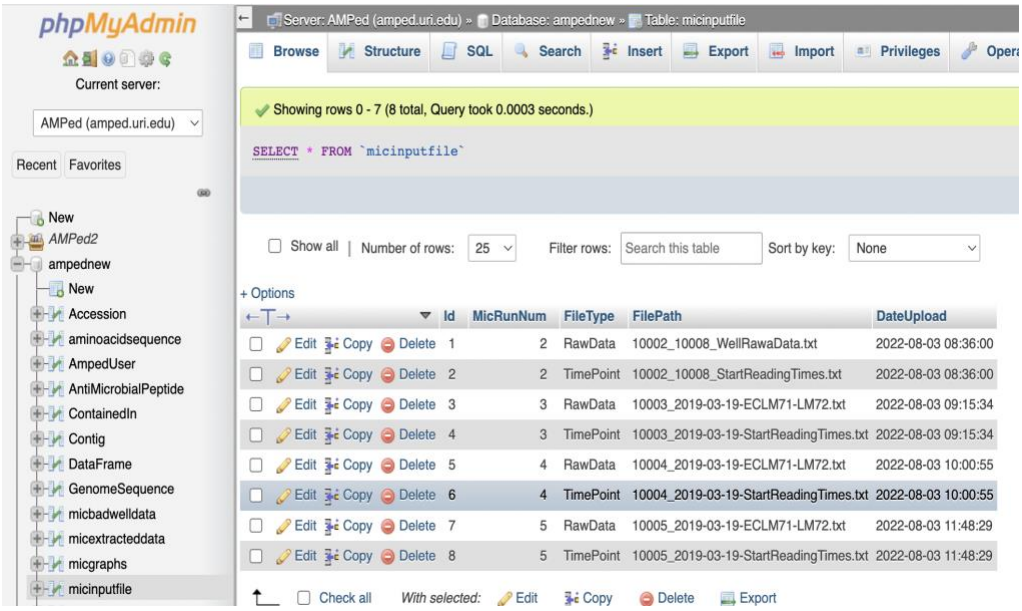


Figure 7. micinputfile Table

The SQL [11] command below describes the creation of micinputfile table with attributes ID, MICRunNumber, FileType, FilePath, Date Upload. ID, servers as the primary key and MICRunNum is the foreign key that is referencing from the micinputfile table

```
CREATE TABLE MIC Input File
(
ID INT ( 1 1 ) NOT NULL ,
MICRunNum VARCHAR( 128 ) NOT NULL,
FileType VARCHAR( 64 ) NOT NULL,
FilePath VARCHAR( 255 ) NOT NULL,
Date Upload DATETIME NOT NULL,
PRIMARY KEY ( ID , MICRunNum ) ,
FOREIGN KEY ( MICRunNum) REFERENCES MICRun ( MICRunNum)
) ;
```

We have a central activity trend that participates in most relationships. This includes the AntiMicrobialPeptide, User, InputMetaData entities.

• We ensured cohesivity, by identifying the weak entities (entities that need an owning entity to provide unique identification because they cannot be uniquely identified through their attribute values alone) and making sure the relationships among the entities are necessary for the functionality of the database.

• We also checked that the internal and external transactions between the tables

are carried out properly and can properly combine information from related tables to provide needed output. This was accomplished by populating the database with data and conducting a logical check with the interdisciplinary team, thereby testing that data access is working properly and returning the right results

Web User Interface Design

AMPed is a data repository, so to access it we need to have an interface [12], see Figure 8. We created a customized domain name amped.uri.edu, hosted in a secure GUI (Graphical User Interface) front end as http://amped.uri.edu. We have developed a GUI with the primary objectives of usability and overall good performance. The new system build involves creation of multiple JSP (Java Server Pages) pages [13]. Each of these page use HTML (Hyper Text Markup Language) [14] and CSS (Cascading Style Sheets) [15] to implement simple interface and apply Core Java [16] to perform business logic, with an MYSQL (A relational database management system) backend see Figure 4 . This provides efficiency as well as strong functional and technical capabilities. The GUI also supports key strategic priorities such as transparency and explainability so that users can understand how the system functions.

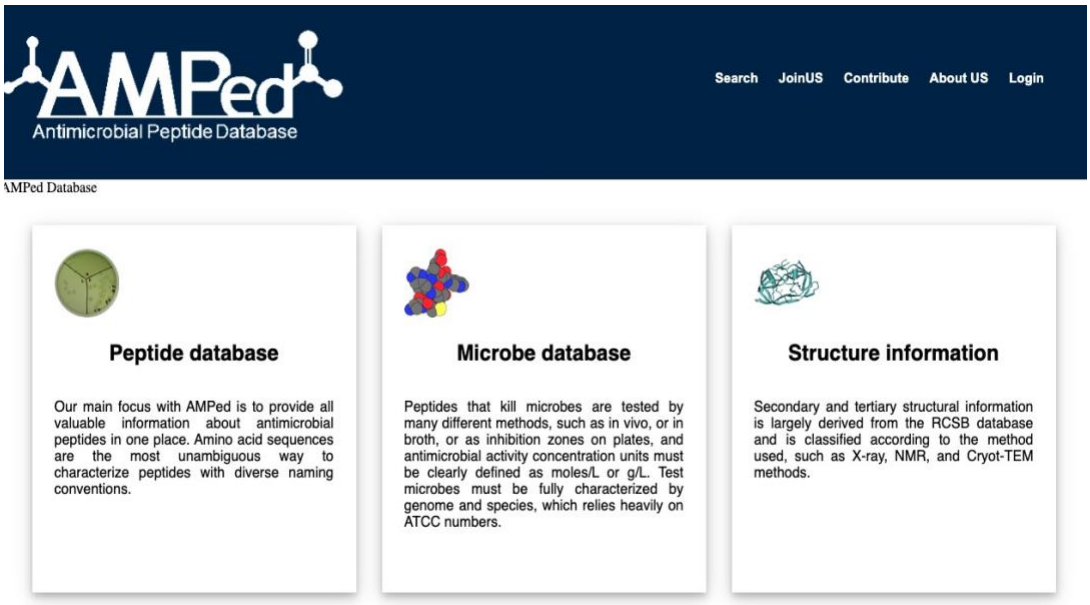


Figure 8. Home page of AMPed application

0.8 User Profile creation page:

AMPed is a self sign up application, see Figure 9. The user has a separate web page developed using JSP code (See Appendix D) where he/she can create their profile, then a request is initiated for approval from the owner (Dr. Martin), once the owner approves the request the user can log in.

Steps to create a profile:

1. First click on sign up

2. After that, the sign-up form will appear. Users need to fill in all the details it requests:

- First name,
- Last name,
- a new username, and a new password,

• Email id,

• Contact number,

• Job title,

• Affiliation,

• City,

• State,

• Zip code.

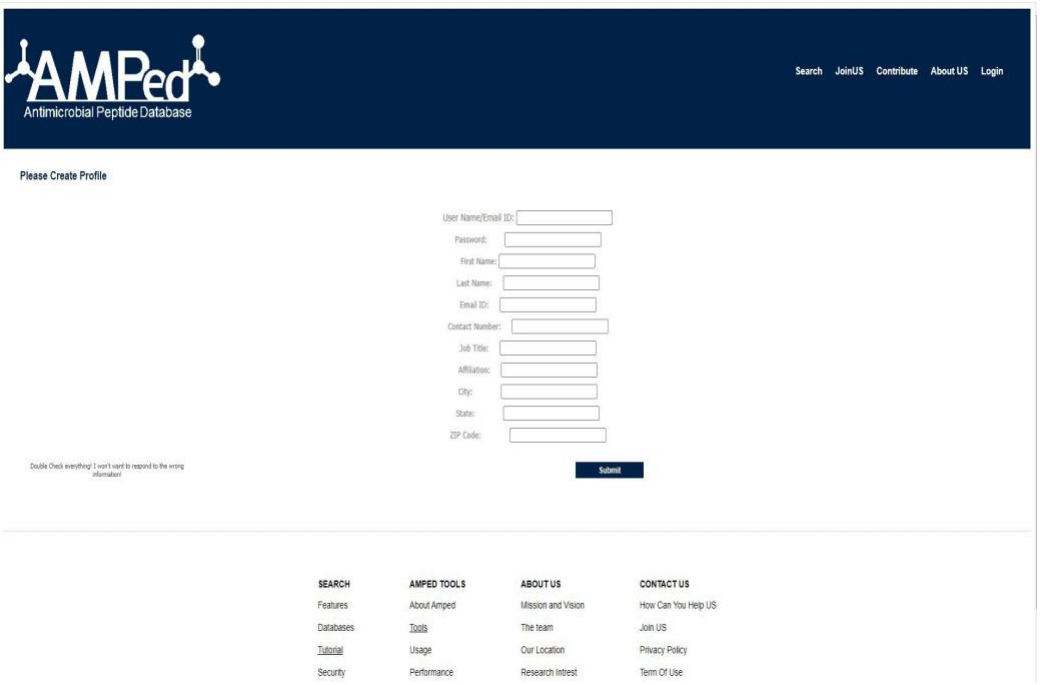


Figure 9. User Profile creation page

0.9 MIC Tool page:

The MIC Tool page, see Figure 3, is developed to upload input files which contain raw data and time point files. Once the user uploads the text files, the application triggers the R script (Sample code shown below) that outputs the means and standard deviations of the MIC values for Peptides 1 and 2 in that assay, and also produces detailed bacterial

growth plots for evaluating the integrity of the assay data more completely. The Mean and SD block for each assay run, output in Figure 10 below shows the input files uploaded by the user and the extracted file that used for calculations, the mean and SDs (standard deviations) of the two peptides tested and growth plots for those input files.

Figure 10. MIC Output page

The first two MIC run id's represent the mean of Peptide 1 and Peptide 2 respectively, and the other two represent the SD's. Figure 11 below shows a sample "mean dataframe" which can be seen by the user who clicks on the MIC Run Id link shown in Figure 10. "X512", "X256",., are Peptide concentrations, and the first column represents number of odd or even plates.

""	"X512"	"X256"	"X128"	"X64"	"X32"	"X16"	"X8"	"X4"	"X2"	"X1"	"DATETIME"
"1"	0.0716666666666667	0.0718333333333333	0.0946666666666667	0.0708333333333333	0.0715	0.0718333333333333	0.072	0.0725	0.0703333333333333	0.0721666666666667	"12:15 2019/03/19"
"2"	0.0715	0.073	0.0728333333333333	0.0741666666666667	0.0745	0.0755	0.0751666666666667	0.0768333333333333	0.0746666666666667	0.0766666666666667	"12:45 2019/03/19"
"3"	0.0758333333333333	0.081	0.0853333333333333	0.0896666666666667	0.0918333333333333	0.0961666666666667	0.0973333333333333	0.0995	0.0968333333333333	0.0995	"13:15 2019/03/19"
"4"	0.0845	0.095	0.107333333333333	0.1145	0.118166666666667	0.125166666666667	0.126166666666667	0.1315	0.126833333333333	0.131833333333333	"13:48 2019/03/19"
"5"	0.104166666666667	0.142	0.159333333333333	0.171166666666667	0.176666666666667	0.186	0.186166666666667	0.1965	0.187	0.193333333333333	"14:43 2019/03/19"
"6"	0.1065	0.161666666666667	0.181666666666667	0.190666666666667	0.189166666666667	0.197666666666667	0.194833333333333	0.208	0.197833333333333	0.205166666666667	"15:15 2019/03/19"
"7"	0.1415	0.224833333333333	0.236166666666667	0.243333333333333	0.243833333333333	0.250833333333333	0.2425	0.260833333333333	0.2545	0.255666666666667	"17:10 2019/03/19"
"8"	0.439833333333333	0.4585	0.493333333333333	0.542833333333333	0.550666666666667	0.562	0.539166666666667	0.567833333333333	0.599333333333333	0.586333333333333	"8:16 2019/03/20"
"9"	0.534666666666667	0.556166666666667	0.556666666666667	0.610833333333333	0.605	0.7035	0.6065	0.711666666666667	0.688333333333333	0.619333333333333	"10:20 2019/03/20"

Figure 11. Sample mean Dataframe

The input files block in the Figure 13 also lists the input raw data files uploaded by the

user used to calculate these MICs. The extracted data file contains the transformed data shown in Figure 12 (cleaned and sorted raw data) file that is actually used as the input file used for subsequent MIC calculations.

""	"12:15 2019/03/19"	"12:15 2019/03/19"	"12:45 2019/03/19"	"12:45 2019/03/19"	"13:15 2019/03/19"	"13:16 2019/03/19"	"13:48 2019/03/19"	"13:49 2019/03/19"	"14:43 2019/03/19"	"14:44 2019/03/19"	"15:15 2019/03/19"	"15:16 2019/03/19"	"17:10 2019/03/19"	"17:11 2019/03/19"	"8:16 2019/03/20"	"8:17 2019/03/20"	"10:20 2019/03/20"	"10:20 2019/03/20"
"1"	0.037	0.039	0.037	0.038	0.037	0.038	0.037	0.038	0.036	0.038	0.036	0.036	0.036	0.037	0.033	0.371	0.034	0.346
"2"	0.066	0.075	0.064	0.072	0.061	0.067	0.06	0.066	0.056	0.063	0.055	0.06	0.056	0.132	0.335	0.447	0.296	0.51
"3"	0.069	0.069	0.069	0.069	0.069	0.069	0.075	0.077	0.109	0.119	0.132	0.136	0.2	0.197	0.394	0.45	0.449	0.557
"4"	0.151	0.127	0.068	0.068	0.073	0.074	0.094	0.09	0.136	0.133	0.17	0.152	0.227	0.2	0.602	0.406	0.589	0.39
"5"	0.065	0.068	0.069	0.071	0.078	0.084	0.097	0.109	0.138	0.164	0.161	0.18	0.211	0.229	0.565	0.65	0.712	0.412
"6"	0.067	0.07	0.071	0.072	0.085	0.087	0.108	0.112	0.154	0.17	0.17	0.183	0.218	0.235	0.676	0.571	0.44	0.744
"7"	0.068	0.069	0.072	0.072	0.092	0.091	0.122	0.118	0.173	0.175	0.188	0.183	0.24	0.229	0.622	0.517	0.838	0.584
"8"	0.068	0.072	0.073	0.073	0.091	0.097	0.118	0.128	0.166	0.184	0.172	0.185	0.219	0.22	0.497	0.702	0.675	0.507
"9"	0.067	0.069	0.072	0.072	0.093	0.095	0.125	0.125	0.176	0.185	0.183	0.194	0.232	0.235	0.593	0.582	0.86	0.791
"10"	0.069	0.07	0.074	0.072	0.095	0.095	0.127	0.125	0.176	0.18	0.184	0.182	0.244	0.223	0.581	0.595	0.839	0.747
"11"	0.068	0.074	0.073	0.076	0.093	0.097	0.124	0.128	0.17	0.185	0.179	0.192	0.234	0.237	0.546	0.698	0.546	0.53
"12"	0.068	0.068	0.076	0.075	0.102	0.099	0.133	0.134	0.192	0.204	0.203	0.234	0.256	0.275	0.571	0.732	0.77	0.517
"13"	0.038	0.04	0.039	0.039	0.038	0.039	0.038	0.039	0.039	0.04	0.037	0.037	0.038	0.035	0.038	0.378	0.039	0.36
"14"	0.071	0.074	0.076	0.07	0.098	0.066	0.128	0.065	0.193	0.062	0.207	0.059	0.257	0.102	0.496	0.398	0.667	0.5
"15"	0.071	0.077	0.076	0.074	0.1	0.076	0.129	0.085	0.193	0.126	0.208	0.152	0.261	0.216	0.48	0.448	0.572	0.54
"16"	0.073	0.071	0.077	0.072	0.102	0.081	0.131	0.098	0.194	0.146	0.209	0.173	0.256	0.234	0.472	0.494	0.549	0.685
"17"	0.071	0.073	0.076	0.074	0.096	0.089	0.125	0.112	0.189	0.173	0.208	0.198	0.263	0.248	0.518	0.487	0.642	0.549
"18"	0.074	0.076	0.078	0.075	0.097	0.095	0.125	0.124	0.19	0.185	0.203	0.196	0.26	0.252	0.502	0.477	0.586	0.536
"19"	0.072	0.074	0.076	0.076	0.097	0.097	0.125	0.127	0.188	0.191	0.204	0.203	0.254	0.261	0.548	0.523	0.772	0.596
"20"	0.072	0.074	0.077	0.075	0.1	0.1	0.128	0.135	0.192	0.202	0.209	0.21	0.255	0.262	0.497	0.515	0.559	0.686
"21"	0.073	0.076	0.079	0.077	0.103	0.103	0.136	0.138	0.207	0.205	0.227	0.213	0.28	0.268	0.479	0.598	0.584	0.699
"22"	0.067	0.072	0.074	0.074	0.096	0.096	0.124	0.126	0.19	0.187	0.208	0.197	0.273	0.251	0.686	0.539	0.586	0.629
"23"	0.071	0.077	0.079	0.079	0.104	0.104	0.136	0.138	0.204	0.203	0.22	0.21	0.272	0.261	0.587	0.574	0.736	0.62
"24"	0.07	0.07	0.078	0.074	0.101	0.098	0.131	0.136	0.194	0.204	0.204	0.226	0.263	0.275	0.595	0.574	0.821	0.695
"25"	0.038	0.04	0.041	0.042	0.04	0.047	0.038	0.05	0.041	0.041	0.034	0.037	0.07	0.045	0.565	0.467	0.611	0.395
"26"	0.07	0.074	0.077	0.07	0.097	0.066	0.124	0.064	0.189	0.062	0.202	0.056	0.23	0.072	0.529	0.434	0.646	0.589
"27"	0.07	0.075	0.078	0.072	0.099	0.073	0.127	0.077	0.193	0.112	0.207	0.135	0.264	0.211	0.525	0.454	0.651	0.568
"28"	0.071	0.075	0.079	0.073	0.099	0.083	0.128	0.103	0.194	0.153	0.208	0.178	0.265	0.235	0.499	0.487	0.578	0.549
"29"	0.071	0.077	0.08	0.075	0.1	0.091	0.127	0.117	0.191	0.172	0.204	0.193	0.26	0.249	0.547	0.49	0.819	0.531
"30"	0.068	0.074	0.077	0.074	0.093	0.094	0.12	0.12	0.183	0.178	0.194	0.189	0.253	0.245	0.537	0.541	0.705	0.619
"31"	0.071	0.077	0.081	0.076	0.099	0.101	0.127	0.132	0.193	0.196	0.207	0.201	0.264	0.257	0.56	0.602	0.701	0.73
"32"	0.07	0.076	0.079	0.074	0.098	0.098	0.122	0.126	0.184	0.189	0.194	0.199	0.242	0.257	0.492	0.532	0.585	0.627
"33"	0.073	0.077	0.084	0.077	0.103	0.1	0.133	0.132	0.206	0.2	0.222	0.209	0.281	0.269	0.55	0.605	0.623	0.713
"34"	0.067	0.077	0.077	0.077	0.096	0.103	0.124	0.135	0.188	0.201	0.202	0.214	0.27	0.266	0.609	0.586	0.673	0.656
"35"	0.068	0.075	0.078	0.075	0.1	0.099	0.134	0.131	0.203	0.195	0.22	0.21	0.273	0.257	0.573	0.54	0.739	0.545
"36"	0.068	0.073	0.08	0.074	0.1	0.101	0.131	0.137	0.196	0.209	0.206	0.232	0.27	0.279	0.536	0.564	0.727	0.655
"37"	0.034	0.044	0.04	0.038	0.037	0.039	0.037	0.04	0.038	0.04	0.031	0.038	0.039	0.051	0.036	0.598	0.035	0.511
"38"	0.064	0.075	0.074	0.073	0.089	0.097	0.114	0.128	0.174	0.191	0.188	0.198	0.239	0.262	0.528	0.56	0.715	0.746
"39"	0.066	0.072	0.074	0.07	0.088	0.091	0.111	0.118	0.163	0.178	0.177	0.186	0.225	0.245	0.449	0.541	0.527	0.729
"40"	0.068	0.07	0.077	0.07	0.095	0.087	0.123	0.112	0.182	0.162	0.198	0.173	0.244	0.233	0.48	0.583	0.605	0.416
"41"	0.068	0.074	0.077	0.073	0.093	0.096	0.118	0.127	0.181	0.195	0.2	0.205	0.251	0.268	0.62	0.595	0.734	0.705
"42"	0.064	0.071	0.073	0.069	0.089	0.09	0.114	0.117	0.173	0.181	0.191	0.198	0.251	0.25	0.539	0.59	0.675	0.66
"43"	0.069	0.07	0.078	0.067	0.096	0.088	0.125	0.114	0.187	0.176	0.206	0.184	0.255	0.236	0.556	0.643	0.705	0.718
"44"	0.065	0.069	0.074	0.066	0.089	0.084	0.115	0.105	0.175	0.16	0.192	0.179	0.244	0.243	0.504	0.687	0.609	0.59
"45"	0.063	0.072	0.073	0.07	0.087	0.093	0.11	0.122	0.169	0.187	0.184	0.2	0.241	0.259	0.537	0.707	0.646	0.661
"46"	0.065	0.073	0.075	0.072	0.094	0.093	0.122	0.12	0.187	0.184	0.206	0.199	0.264	0.248	0.689	0.684	0.808	0.774

Figure 12. Sample Transformed 96 well plates Dataframe

The code below calculates the mean and standard deviation of Peptides1 and Peptide 2. Firstly, it assigns 4 empty lists to store the means and SD of Peptide 1 and Peptide 2. Then the for-loop binds the first three rows (Peptide 1) of odd and even plates and then takes the column mean of all the 6 rows (3 odd+ 3 even) which is the mean for the Peptide 1. The is repeated for Peptide 2. The loop executes for all the plates and creates a dataframe which contains all the means of both peptides as shown in Figure 11.

```
# Peptides mean and  SD

    plateCount = 1
```

```
averageGrowthPP1_mean = list()
averageGrowthPP2_mean = list()
averageGrowthPP1_SD = list()
averageGrowthPP2_SD = list()
for(i in 1:length(evenPlates))
{
averageGrowthPP1_mean[[plateCount]] <-
 colMeans(rbind(oddPlates[[i]][1:3,2:11], evenPlates[[i]][1:3,2:11])
, na.rm=TRUE)
averageGrowthPP2_mean[[plateCount]] <-
 colMeans(rbind(oddPlates[[i]][4:6,2:11], evenPlates[[i]][4:6,2:11])
, na.rm=TRUE)
averageGrowthPP1_SD[[plateCount]] <-
 sapply(rbind(oddPlates[[i]][1:3,2:11], evenPlates[[i]][1:3,2:11]),
sd, na.rm=TRUE)
averageGrowthPP2_SD[[plateCount]] <-
 sapply(rbind(oddPlates[[i]][4:6,2:11], evenPlates[[i]][4:6,2:11]),
sd, na.rm=TRUE)
plateCount = plateCount + 1
}
average Growth PP 1_mean
average Growth PP 2_mean
average Growth PP 1_SD
average Growth PP 2_SD

  if(num_plates %% 2 != 0){
averageGrowthPP1_mean[[length(averageGrowthPP1_mean)+1]] <-
 colMeans(platesRawData[[num_plates]][1:3,2:11], na.rm=TRUE)
```

```
averageGrowthPP2_mean[[length(averageGrowthPP2_mean)+1]] <-
  colMeans(platesRawData[[num_plates]][4:6,2:11], na.rm=TRUE)
averageGrowthPP1_SD[[length(averageGrowthPP1_SD)+1]] <-
  sapply(platesRawData[[num_plates]][1:3,2:11], sd, na.rm=TRUE)
averageGrowthPP2_SD[[length(averageGrowthPP2_SD)+1]] <-
sapply(platesRawData[[num_plates]][4:6,2:11], sd, na.rm=TRUE)
}
averageGrowthPP1_mean
averageGrowthPP2_mean
averageGrowthPP1_SD
averageGrowthPP2_SD
```

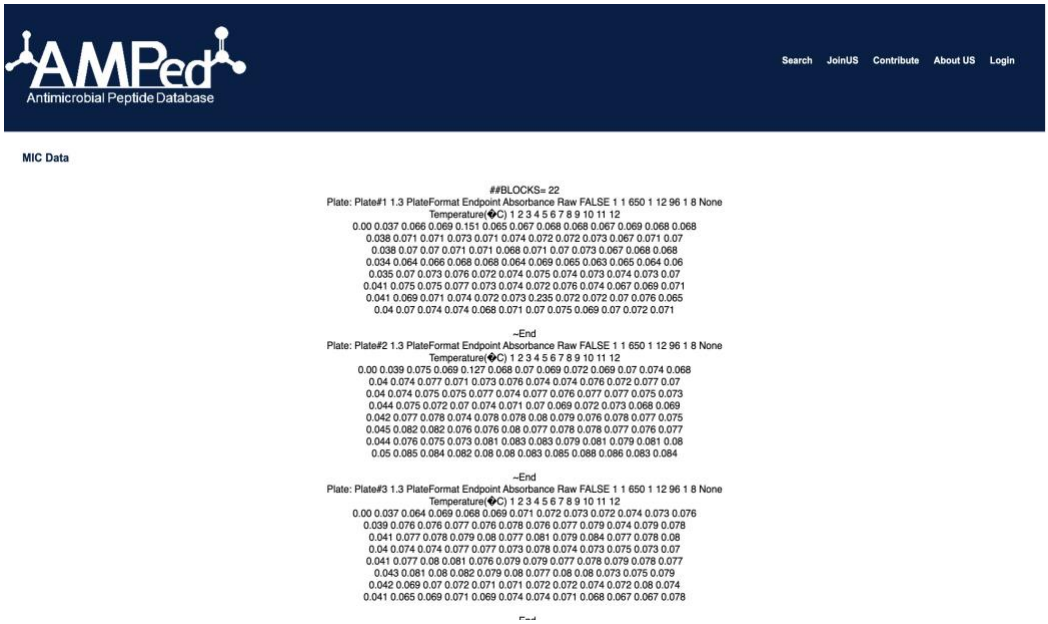


Figure 13. Sample input raw data

0.10 MIC History page:

To achieve transparency, and integrity of the application, MIC pages have been created to display the details about the MIC run activity such as uploading of input files, extracting the data for necessary calculations and displaying means, SD and growth

plots, and bad wells found during the data cleaning process, Figure 14.

Technical Validation:

Web applications are more complex these days employing multiple technologies and having complex front ends, which makes the back ends more intricate. So, we validated our database and web applications effectively to ensure security and quality. Data often travels back and forth from the UI (User Interface) to the backend DB and vice versa, so we carried out the following integrity checks:

1. We made sure that the fields in the UI are mapped consistently with the corresponding fields in the database tables, and that the database actions are consistent between the UI and database. Whenever a certain action is taken at the front end, a create, retrieve, update or delete action gets invoked at the back end. When a user wants to save any data, the create operation should be invoked. If the user wants to search for something in the application, the retrieve operation should be invoked. Similarly edit should invoke an update operation and if the user wants to remove something, the delete operation should be invoked. Any operation performed in the database will always be one of these four operations.

2. We created primary keys before any other fields are created to ensure data consistency. Foreign keys are completely indexed for easy retrieval and search. The normalization process sometimes fractures conceptual entities into multiple tables for consistency, correctness and efficiency, but this information needs to be recomposed correctly upon query, and foreign keys enable this process.

3. The key fields in the tables are defined with a unique index to avoid the

duplicate entries into the tables.

4. To make sure that the data uploaded to the algorithm is correct, we checked that the number of plates imported into the plate data matches the given number of time points.

5. We ensured the overall functionalities of the web applications software is defect free under successive updates. For example, when a user creates their profile, their name is entered into the application and into a field that will not accept numbers or other special characters.

6. We checked the functionality of the overall system, by executing the R algorithm independently and comparing the results with the output obtained by executing the R script through the application. x

Usage Notes

1. Users mainly interact with the system through the web portal (http://amped.uri.edu). However, internal users may interact with the database directly. This includes people from the genomics group, technical administrators, and others.

2. Amped uses a MySQL database which is also hosted on the same server where the application resides.Users can get all the details regarding the database from the URL below: https://wiki.ucs.uri.edu/dokuwiki/server_amped

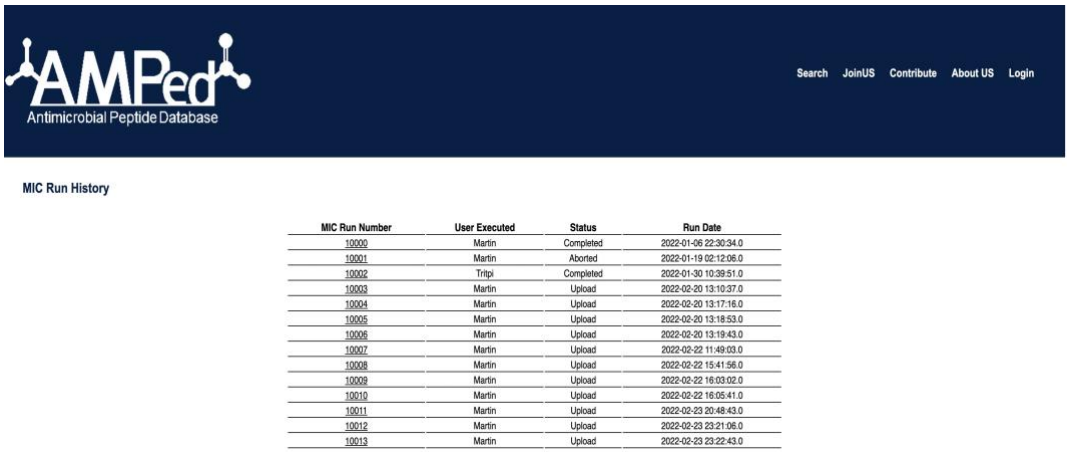
3. The phpMyAdmin web interface is available to access the database. Login credentials are needed for accessing the database. These credentials are available in the URL mentioned above.

4. Software versions used are:

- Perl 5.16.3
- PHP 5.4.16
- MariaDB 5.5.60
- Apache 2.4.6
- Python 2.7.5
- R 3.5.2
- RStudio Server 1.1.463

5. To use the MIC tool, the user has to login into the AMPed web application, where the MIC tool is integrated in the separate tools web page. Users upload the input files and click on submit. The java business logic processes the complete MIC workflow, which includes executing R script and displaying the OD plots with annotations in the result web page.

6. To achieve transparency [17], an additional web page named "MIC History" has been created where the user can see the original format of their input files and how these input files have been transformed into 96 row data frames and how the means,SDs and growth plots are calculated, see Figure 14



MIC Run Number	User Executed	Status	Run Date
10000	Martin	Completed	2022-01-06 22:30:34.0
10001	Martin	Aborted	2022-01-19 02:12:06.0
10002	Tritpi	Completed	2022-01-30 10:39:51.0
10003	Martin	Upload	2022-02-20 13:10:37.0
10004	Martin	Upload	2022-02-20 13:17:16.0
10005	Martin	Upload	2022-02-20 13:18:53.0
10006	Martin	Upload	2022-02-20 13:19:43.0
10007	Martin	Upload	2022-02-22 11:49:03.0
10008	Martin	Upload	2022-02-22 15:41:56.0
10009	Martin	Upload	2022-02-22 16:03:02.0
10010	Martin	Upload	2022-02-22 16:05:41.0
10011	Martin	Upload	2022-02-23 20:48:43.0
10012	Martin	Upload	2022-02-23 23:21:06.0
10013	Martin	Upload	2022-02-23 23:22:43.0

Figure 14. MIC History page

Code availability

All the code developed for the thesis is available in the GitHub link provided below:

https://github.com/lmmartin/AMPed-URI-Martin

References

[1] Joshi, M. & Bhardwaj, P. Impact of data transparency: Scientific publications. Perspectives Clin. Res. 9, 31 (2018).

[2] Morey, T., Forbath, T. & Schoop, A. Customer data: Designing for transparency and trust. Harv. Bus. Rev. 93, 96–105 (2015).

[3] Obafemi-Ajayi, T. et al. No-boundary thinking: a viable solution to ethical data-driven ai in precision medicine. AI Ethics 1–9 (2021).

[4] Acharya, S. S. Tools for anti-microbial peptide editable database. DigitalCommonsURI (2017).

[5] Gardener, M. Beginning R: The statistical programming language (John Wiley & Sons, 2012).

[6] Elmasri, R. & Navathe, S. B. Database systems, vol. 9 (Pearson Education Boston, MA, 2011).

[7] Cagiltay, N. E., Tokdemir, G., Kilic, O. & Topalli, D. Performing and analyzing non-formal inspections of entityrelationship diagram (erd). J. Syst. Softw. 86, 2184–2195 (2013).

[8] Rouse, M. database (db). SearchSQLServer, accessed Oct. 9 (2017).

[9] Zoratti, I. Mysql security best practices. In 2006 IET Conference on Crime and Security, 183–198 (IET, 2006).

[10] Hillyer, M. An introduction to database normalization. MySQL AB (2003).

[11] Subali, M. A. P. & Rochimah, S. A new model for measuring the complexity of sql commands. In 2018 10th International Conference on

Information Technology and Electrical Engineering (ICITEE), 1–5 (IEEE, 2018).

[12] Garg, T. Graphical user interface for antimicrobial peptide database. DigitalCommonsURI (2016).

[13] Kurniawan, B. Servlet & jsp: A tutorial. (2015).

[14] Castro, E. HTML for the world wide web (Peachpit Press, 2003)

[15] Lie, H. W. & Bos, B. Cascading style sheets: Designing for the web, Portable Documents (Addison-Wesley Professional,365 2005)

[16] Horstmann, C. S. & Cornell, G. Core Java 2: Fundamentals, vol. 1 (Prentice Hall Professional, 2001).

[17] Callaghan, S. On the importance of data transparency. Patterns 1 (2020).

Acknowledgements

I would like to sincerely acknowledge and give my warmest thanks to major advisor Dr. Joan Peckham and co-major advisor Dr. Lenore Martin without whom this work wouldn't have been possible. Their guidance and advice carried me through all the stages of completing this research. I would also like to thank my internal committee member Dr. Abdeltawab Hendawi for his brilliant comments and suggestions.

Author contributions statement

Dr. Lenore Martin and Dr. Joan Peckham conceived the presented idea. Dr.Martin with her students performed experiments in her lab at University of Rhode Island. Anusha Singamaneni designed the algorithm that analyses and perform calculations on the data, updated the database design, and developed new user web interface. All authors contributed to the interpretation of the results provided critical feedback and helped shape the research, analysis and manuscript.

Competing interests

We have no conflict of interests to disclose.

CHAPTER 9

Conclusion and Future Work

The AMPed project is a collective effort over the past few years to create a uniform and complete collection of antimicrobial peptides by storing them in a persistent database. It has been in development for several years under the direction of Professors Joan Peckham and Lenore Martin. The database design was developed by our team in such a way that all the requirements from the rest of the sources will be stored uniformly into one database application

The primary goals for this project were to build the MIC tool that would be helpful for researchers to calculate and visualize analysis results, to normalize the database, and to design a new website with important and secure enhancements, such as the MIC analysis results and a Glossary.

We are also successful in achieving another overarching primary goal which is transparency of data, by providing clear explanations about methods and procedures used in the MIC algorithm analysis, and the management of the data that is input to and output from the algorithm in separate webpages.

The MIC/MLC calculator was developed and tested by the research group in Dr. Martin's lab. The tool is working as expected. It was developed and tested using sample data with unexpected constraints that may occur while testing on real data. The tool was developed keeping in mind that authorized users may need to extend or modify it.

In the end with the proper collaboration and communication between computer scientists and biologists, a result that will benefit different people has been reached. In the near future this opens new doors for experimentation and research while utilizing the knowledge of both parties.

Future Work:

The major parts of this thesis are the initial design of the database, the implementation of the database, the web interface for user interaction, and finally calculation tools (MIC\MLC, MassSpec) that will have to pull from the data in order to produce an outcome. Almost all parts have been completed through this thesis. A Mass Spectrometry tool is envisioned and has not yet been completed. This part can be done either by the AMPed team, or by future graduate students that will work on the continuation of this project. Also important will be the development of clear guidance for updating the design of the database as needed when experimental designs and data formats change with the introduction of new analysis equipment and procedures.

APPENDICES

Appendix A

Sample input data file. The figure below shows only 3 plates but there can be many blocks as generated by the experimental data

##BLOCKS= 22

Plate:	Plate#1 1.3	PlateFormat		Endpoint		Absorbance		Raw	FALSE	1			
	Temperature(°C)	1	2	3	4	5	6	7	8	9	10	11	12
	0.00	0.037	0.066	0.069	0.151	0.065	0.067	0.068	0.068	0.067	0.069	0.068	0.068
		0.038	0.071	0.071	0.073	0.071	0.074	0.072	0.072	0.073	0.067	0.071	0.07
		0.038	0.07	0.07	0.071	0.071	0.068	0.071	0.07	0.073	0.067	0.068	0.068
		0.034	0.064	0.066	0.068	0.068	0.064	0.069	0.065	0.063	0.065	0.064	0.06
		0.035	0.07	0.073	0.076	0.072	0.074	0.075	0.074	0.073	0.074	0.073	0.07
		0.041	0.075	0.075	0.077	0.073	0.074	0.072	0.076	0.074	0.067	0.069	0.071
		0.041	0.069	0.071	0.074	0.072	0.073	0.235	0.072	0.072	0.07	0.076	0.065
		0.04	0.07	0.074	0.074	0.068	0.071	0.07	0.075	0.069	0.07	0.072	0.071
~End													
Plate:	Plate#2 1.3	PlateFormat		Endpoint		Absorbance		Raw	FALSE	1			
	Temperature(°C)	1	2	3	4	5	6	7	8	9	10	11	12
	0.00	0.039	0.075	0.069	0.127	0.068	0.07	0.069	0.072	0.069	0.07	0.074	0.068
		0.04	0.074	0.077	0.071	0.073	0.076	0.074	0.074	0.076	0.072	0.077	0.07
		0.04	0.074	0.075	0.075	0.077	0.074	0.077	0.076	0.077	0.077	0.075	0.073
		0.044	0.075	0.072	0.07	0.074	0.071	0.07	0.069	0.072	0.073	0.068	0.069
		0.042	0.077	0.078	0.074	0.078	0.078	0.08	0.079	0.076	0.078	0.077	0.075
		0.045	0.082	0.082	0.076	0.076	0.08	0.077	0.078	0.078	0.077	0.076	0.077
		0.044	0.076	0.075	0.073	0.081	0.083	0.083	0.079	0.081	0.079	0.081	0.08
		0.05	0.085	0.084	0.082	0.08	0.08	0.083	0.085	0.088	0.086	0.083	0.084
~End													
Plate:	Plate#3 1.3	PlateFormat		Endpoint		Absorbance		Raw	FALSE	1			
	Temperature(°C)	1	2	3	4	5	6	7	8	9	10	11	12
	0.00	0.037	0.064	0.069	0.068	0.069	0.071	0.072	0.073	0.072	0.074	0.073	0.076
		0.039	0.076	0.076	0.077	0.076	0.078	0.076	0.077	0.079	0.074	0.079	0.078
		0.041	0.077	0.078	0.079	0.08	0.077	0.081	0.079	0.084	0.077	0.078	0.08
		0.04	0.074	0.074	0.077	0.077	0.073	0.078	0.074	0.073	0.075	0.073	0.07
		0.041	0.077	0.08	0.081	0.076	0.079	0.079	0.077	0.078	0.079	0.078	0.077
		0.043	0.081	0.08	0.082	0.079	0.08	0.077	0.08	0.08	0.073	0.075	0.079
		0.042	0.069	0.07	0.072	0.071	0.071	0.072	0.072	0.074	0.072	0.08	0.074
		0.041	0.065	0.069	0.071	0.069	0.074	0.074	0.071	0.068	0.067	0.067	0.078
~End													

Appendix B

Sample input times file. Below figure shows only times for 4 plates but there can be as many blocks as generated by experiment

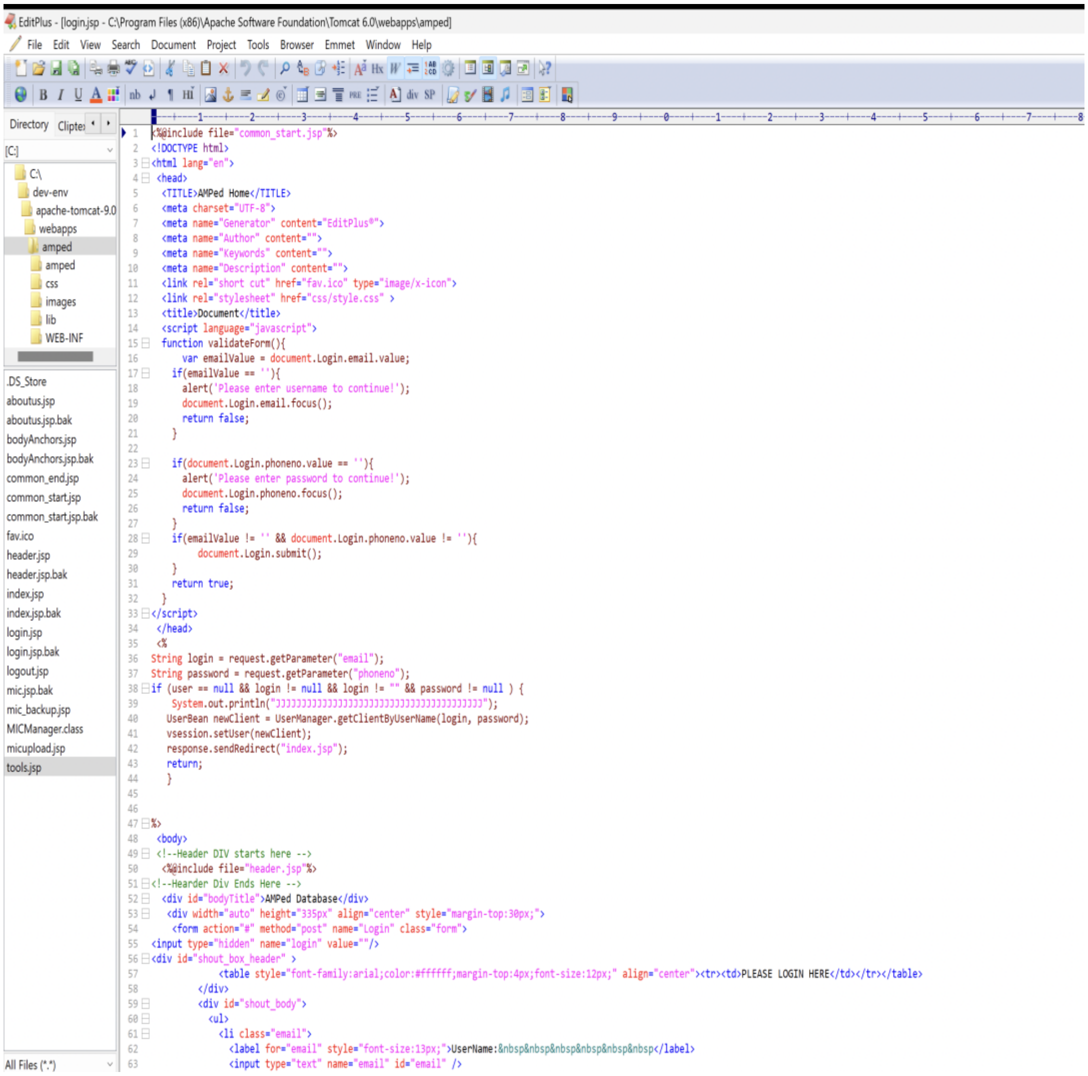
Appendix C

Transformed the input file into 96 rows data frame with time as columns. The figure below shows only up to 45 rows.

""	"12:15 2019/03/19"	"12:15 2019/03/19"	"12:45 2019/03/19"	"12:45 2019/03/19"	"13:15 2019/03/19"	"13:16 2019/03/19"	"13:48 2019/03/19"	"13:49 2019/03/19"	"14:43 2019/03/19"	"14:44 2019/03/19"	"15:15 2019/03/19"	"15:16 2019/03/19"	"17:10 2019/03/19"	"17:11 2019/03/19"	"8:16 2019/03/20"	"8:17 2019/03/20"	"10:20 2019/03/20"	"10:20 2019/03/20"
"1"	0.037	0.039	0.037	0.038	0.037	0.038	0.037	0.038	0.036	0.038	0.036	0.036	0.036	0.037	0.033	0.371	0.034	0.346
"2"	0.066	0.075	0.064	0.072	0.061	0.067	0.06	0.066	0.056	0.063	0.055	0.06	0.056	0.132	0.335	0.447	0.296	0.51
"3"	0.069	0.069	0.069	0.069	0.069	0.069	0.075	0.077	0.109	0.119	0.132	0.136	0.2	0.197	0.394	0.45	0.449	0.557
"4"	0.151	0.127	0.068	0.068	0.073	0.074	0.094	0.09	0.136	0.133	0.17	0.152	0.227	0.2	0.602	0.406	0.589	0.39
"5"	0.065	0.068	0.069	0.071	0.078	0.084	0.097	0.109	0.138	0.164	0.161	0.18	0.211	0.229	0.565	0.65	0.712	0.412
"6"	0.067	0.07	0.071	0.072	0.085	0.087	0.108	0.112	0.154	0.17	0.17	0.183	0.218	0.235	0.676	0.571	0.44	0.744
"7"	0.068	0.069	0.072	0.072	0.092	0.091	0.122	0.118	0.173	0.175	0.188	0.183	0.24	0.229	0.622	0.517	0.838	0.584
"8"	0.068	0.072	0.073	0.073	0.091	0.097	0.118	0.128	0.166	0.184	0.172	0.185	0.219	0.22	0.497	0.702	0.675	0.507
"9"	0.067	0.069	0.072	0.072	0.093	0.095	0.125	0.125	0.176	0.185	0.183	0.194	0.232	0.235	0.593	0.582	0.86	0.791
"10"	0.069	0.07	0.074	0.072	0.095	0.095	0.127	0.125	0.176	0.18	0.184	0.182	0.244	0.223	0.581	0.595	0.839	0.747
"11"	0.068	0.074	0.073	0.076	0.093	0.097	0.124	0.128	0.17	0.185	0.179	0.192	0.234	0.237	0.546	0.698	0.546	0.53
"12"	0.068	0.068	0.076	0.075	0.102	0.099	0.133	0.134	0.192	0.204	0.203	0.234	0.256	0.275	0.571	0.732	0.77	0.517
"13"	0.038	0.04	0.039	0.039	0.038	0.039	0.038	0.039	0.039	0.04	0.037	0.037	0.038	0.035	0.038	0.378	0.039	0.36
"14"	0.071	0.074	0.076	0.07	0.098	0.066	0.128	0.065	0.193	0.062	0.207	0.059	0.257	0.102	0.496	0.398	0.667	0.5
"15"	0.071	0.077	0.076	0.074	0.1	0.076	0.129	0.085	0.193	0.126	0.208	0.152	0.261	0.216	0.48	0.448	0.572	0.54
"16"	0.073	0.071	0.077	0.072	0.102	0.081	0.131	0.098	0.194	0.146	0.209	0.173	0.256	0.234	0.472	0.494	0.549	0.685
"17"	0.071	0.073	0.076	0.074	0.096	0.089	0.125	0.112	0.189	0.173	0.208	0.198	0.263	0.248	0.518	0.487	0.642	0.549
"18"	0.074	0.076	0.078	0.075	0.097	0.095	0.125	0.124	0.19	0.185	0.203	0.196	0.26	0.252	0.502	0.477	0.586	0.536
"19"	0.072	0.074	0.076	0.076	0.097	0.097	0.125	0.127	0.188	0.191	0.204	0.203	0.254	0.261	0.548	0.523	0.772	0.596
"20"	0.072	0.074	0.077	0.075	0.1	0.1	0.128	0.135	0.192	0.202	0.209	0.21	0.255	0.262	0.497	0.515	0.559	0.686
"21"	0.073	0.076	0.079	0.077	0.103	0.103	0.136	0.138	0.207	0.205	0.227	0.213	0.28	0.268	0.479	0.598	0.584	0.699
"22"	0.067	0.072	0.074	0.074	0.096	0.096	0.124	0.126	0.19	0.187	0.208	0.197	0.273	0.251	0.686	0.539	0.586	0.629
"23"	0.071	0.077	0.079	0.079	0.104	0.104	0.136	0.138	0.204	0.203	0.22	0.21	0.272	0.261	0.587	0.574	0.736	0.62
"24"	0.07	0.07	0.078	0.074	0.101	0.098	0.131	0.136	0.194	0.204	0.204	0.226	0.263	0.275	0.595	0.574	0.821	0.695
"25"	0.038	0.04	0.041	0.042	0.04	0.047	0.038	0.05	0.041	0.041	0.034	0.037	0.07	0.045	0.565	0.467	0.611	0.395
"26"	0.07	0.074	0.077	0.07	0.097	0.066	0.124	0.064	0.189	0.062	0.202	0.056	0.23	0.072	0.529	0.434	0.646	0.589
"27"	0.07	0.075	0.078	0.072	0.099	0.073	0.127	0.077	0.193	0.112	0.207	0.135	0.264	0.211	0.525	0.454	0.651	0.568
"28"	0.071	0.075	0.079	0.073	0.099	0.083	0.128	0.103	0.194	0.153	0.208	0.178	0.265	0.235	0.499	0.487	0.578	0.549
"29"	0.071	0.077	0.08	0.075	0.1	0.091	0.127	0.117	0.191	0.172	0.204	0.193	0.26	0.249	0.547	0.49	0.819	0.531
"30"	0.068	0.074	0.077	0.074	0.093	0.094	0.12	0.12	0.183	0.178	0.194	0.189	0.253	0.245	0.537	0.541	0.705	0.619
"31"	0.071	0.077	0.081	0.076	0.099	0.101	0.127	0.132	0.193	0.196	0.207	0.201	0.264	0.257	0.56	0.602	0.701	0.73
"32"	0.07	0.076	0.079	0.074	0.098	0.098	0.122	0.126	0.184	0.189	0.194	0.199	0.242	0.257	0.492	0.532	0.585	0.627
"33"	0.073	0.077	0.084	0.077	0.103	0.1	0.133	0.132	0.206	0.2	0.222	0.209	0.281	0.269	0.55	0.605	0.623	0.713
"34"	0.067	0.077	0.077	0.077	0.096	0.103	0.124	0.135	0.188	0.201	0.202	0.214	0.27	0.266	0.609	0.586	0.673	0.656
"35"	0.068	0.075	0.078	0.075	0.1	0.099	0.134	0.131	0.203	0.195	0.22	0.21	0.273	0.257	0.573	0.54	0.739	0.545
"36"	0.068	0.073	0.08	0.074	0.1	0.101	0.131	0.137	0.196	0.209	0.206	0.232	0.27	0.279	0.536	0.564	0.727	0.655
"37"	0.034	0.044	0.04	0.038	0.037	0.039	0.037	0.04	0.038	0.04	0.031	0.038	0.039	0.051	0.036	0.598	0.035	0.511
"38"	0.064	0.075	0.074	0.073	0.089	0.097	0.114	0.128	0.174	0.191	0.188	0.198	0.239	0.262	0.528	0.56	0.715	0.746
"39"	0.066	0.072	0.074	0.07	0.088	0.091	0.111	0.118	0.163	0.178	0.177	0.186	0.225	0.245	0.449	0.541	0.527	0.729
"40"	0.068	0.07	0.077	0.07	0.095	0.087	0.123	0.112	0.182	0.162	0.198	0.173	0.244	0.233	0.48	0.583	0.605	0.416
"41"	0.068	0.074	0.077	0.073	0.093	0.096	0.118	0.127	0.181	0.195	0.2	0.205	0.251	0.268	0.62	0.595	0.734	0.705
"42"	0.064	0.071	0.073	0.069	0.089	0.09	0.114	0.117	0.173	0.181	0.191	0.198	0.251	0.25	0.539	0.59	0.675	0.66
"43"	0.069	0.07	0.078	0.067	0.096	0.088	0.125	0.114	0.187	0.176	0.206	0.184	0.255	0.236	0.556	0.643	0.705	0.718
"44"	0.065	0.069	0.074	0.066	0.089	0.084	0.115	0.105	0.175	0.16	0.192	0.179	0.244	0.243	0.504	0.687	0.609	0.59
"45"	0.063	0.072	0.073	0.07	0.087	0.093	0.11	0.122	0.169	0.187	0.184	0.2	0.241	0.259	0.537	0.707	0.646	0.661
"46"	0.065	0.073	0.075	0.072	0.094	0.093	0.122	0.12	0.187	0.184	0.206	0.199	0.264	0.248	0.689	0.684	0.808	0.774

Appendix D

UI JSP Code



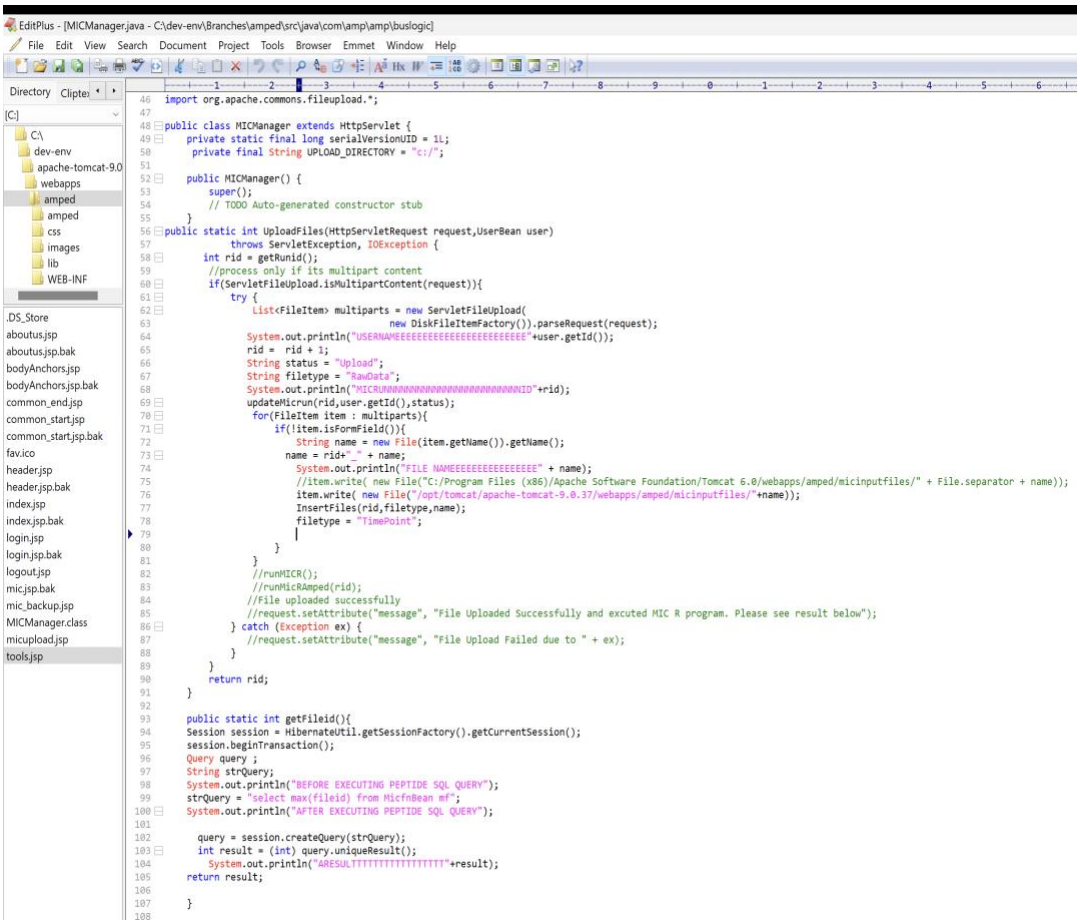
```
<%@include file="common_start.jsp"%>
<!DOCTYPE html>
<html lang="en">
 <head>
  <TITLE>AMPed Home</TITLE>
  <meta charset="UTF-8">
  <meta name="Generator" content="EditPlus®">
  <meta name="Author" content="">
  <meta name="Keywords" content="">
  <meta name="Description" content="">
  <link rel="short cut" href="fav.ico" type="image/x-icon">
  <link rel="stylesheet" href="css/style.css" >
  <title>Document</title>
  <script language="javascript">
  function validateForm(){
      var emailValue = document.Login.email.value;
    if(emailValue == ''){
      alert('Please enter username to continue!');
      document.Login.email.focus();
      return false;
    }

    if(document.Login.phoneno.value == ''){
      alert('Please enter password to continue!');
      document.Login.phoneno.focus();
      return false;
    }
    if(emailValue != '' && document.Login.phoneno.value != ''){
        document.Login.submit();
    }
    return true;
  }
</script>
 </head>
 <%
String login = request.getParameter("email");
String password = request.getParameter("phoneno");
if (user == null && login != null && login != "" && password != null ) {
    System.out.println("JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ");
   UserBean newClient = UserManager.getClientByUserName(login, password);
   vsession.setUser(newClient);
   response.sendRedirect("index.jsp");
   return;
   }


%>
 <body>
 <!--Header DIV starts here -->
  <%@include file="header.jsp"%>
<!--Hearder Div Ends Here -->
  <div id="bodyTitle">AMPed Database</div>
   <div width="auto" height="335px" align="center" style="margin-top:30px;">
    <form action="#" method="post" name="Login" class="form">
<input type="hidden" name="login" value=""/>
<div id="shout_box_header" >
              <table style="font-family:arial;color:#ffffff;margin-top:4px;font-size:12px;" align="center"><tr><td>PLEASE LOGIN HERE</td></tr></table>
          </div>
          <div id="shout_body">
            <ul>
              <li class="email">
                <label for="email" style="font-size:13px;">UserName:      </label>
                <input type="text" name="email" id="email" />
```

Appendix E

Java code that triggers the R script



Appendix F

R script to eliminate bad wells if any

```
well <- readline(prompt="Enter Excluded Well ID: ")
if(well!= 0){
  odd_even <- readline(prompt="Enter odd/even: ")
  well_position <- read.csv("Well_positions.csv")
  m1 <- as.matrix(well_position)
  D1 <- which(m1==well, arr.ind=TRUE)
  ROWNum <-D1[,1]
  ColNum <- D1[,2]
  exlValue <- as.numeric('NA')
  oddPlates[[1]][ROWNum,ColNum]
  length(oddPlates)
  if(odd_even == "odd"){
    for(i in 1:length(oddPlates)){
      oddPlates[[i]][ROWNum,ColNum] <- exlValue
    }
  }else{
    for(i in 1:length(evenPlates)){
      evenPlates[[i]][ROWNum,ColNum] <- exlValue
    }
  }
}
```

Appendix G

R script flow chart