Investigating Correlations between Environmental Factors and Bacteriophage Genomes

Lena Armstrong, Jasmine Cao, Allison Chang, Devin Golla, Connie Liu, Marina Mancoridis, Vassiliki Mancoridis, Ethan Moyer, Luke Patton and Elena Swecker

Abstract

The purpose of this research project was to discover relationships between bacteriophage genomes and the environment in which they were found. In order to gather the environmental data, we parsed through a database from the National Oceanic and Atmospheric Administration. Likewise, to gather genetic data, we parsed through a database from GenBank and the Actinobacteriophage Database. We implemented both the k-means clustering algorithm and the affinity propagation clustering algorithm to group bacteriophages by similar attributes. We cross-verified the validity of clusters by comparing the output of both algorithms. Once we were able to successfully cluster bacteriophages, we were able to implement various chi-squared tests of homogeneity. These determined whether genetic attributes were distributed in the same way across environmental clusters. We found that the guanine-cytosine content of a bacteriophage was strongly related to the average low temperature and record high temperature of the location in which that bacteriophage was discovered. In addition, we used statistical analyses to identify three proteins whose occurrences within a bacteriophage genome highly correlate to the temperature of that bacteriophage's environment. Upon further analysis, we speculate those three hypothetical proteins could have co-occurred or even coevolved.

I. Introduction

A. Background

1. Bacteriophages

Bacteriophages, or phages, are viruses that infect bacteria.[1] In 1915, Frederick Twort first hypothesized the existence of an "ultra-microscopic virus" after observing a contaminant in his vaccinia virus propagation experiment;[2] however, he was unable to confirm their existence. Two years later, microbiologist Felix d'Herelle reported similar findings and coined the term bacteriophage, which roughly translates to "bacteria eater."[3]

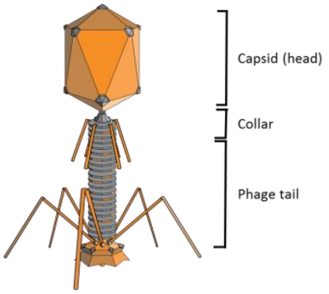

Figure 1: Bacteriophage Structure[4]

Most bacteriophages have a relatively simple structure, as depicted in Figure 1. Generally, the genetic material of phages that infect actinobacteria is encapsulated in a protein capsid, which is connected to a collar and a tail. Their genes can be encoded by either DNA or RNA, which may be single or double stranded. When a bacteriophage attaches to a host, it uses its tail to inject its genetic material through the host's cell membrane. Through either a lytic or lysogenic life cycle, the bacteriophages replicate themselves using the host as a platform (see Figure 2). Virulent bacteriophages multiply through the lytic cycle. The bacteria treat the bacteriophage DNA as if it were their own; in this way, bacteriophage DNA is replicated, transcribed, and translated into viral DNA. They continue propagating within the host until enough bacteriophages have been formed to burst the host's cell membrane. At this point, the bacteriophages are released. Temperate bacteriophages take another approach- the lysogenic cycle. In this cycle, they incorporate their genetic material into the host cell's chromosomes and replicate without harming their hosts. Specific environmental stressors induce the bacteriophage to exit the host's genome, enter the lytic cycle, and lyse the host cell.

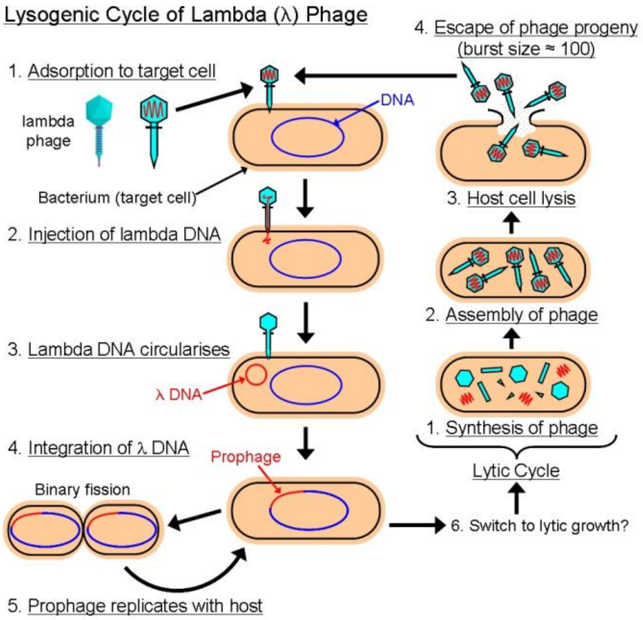
Lysogenic Cycle of Lambda (λ) Phage
4. Escape of phage progeny
(burst size ≈ 100)
1. Adsorption to target cell
lambda phage
DNA
Bacterium (target cell)
3. Host cell lysis
2. Injection of lambda DNA
2. Assembly of phage
3. Lambda DNA circularises
λ DNA
1. Synthesis of phage
Lytic Cycle
4. Integration of λ DNA
Prophage
Binary fission
6. Switch to lytic growth?
5. Prophage replicates with host

Figure 2: Lysogenic and Lytic Life Cycles[5]

Bacteriophages emerged over three billion years ago. As a result of rapid reproductive cycles, bacteriophage mutations are frequently introduced. A rapid mutation rate causes rapid evolution amongst the phages, allowing their differences in different environments to be studied. Bacteriophage genomes range from four to hundreds of genes, so they are relatively simple to research in the context of genomics.[6]

In this project, we focused on mycobacteriophages: bacteriophages that infect bacterial hosts in the genus *Mycobacterium*. Within the phylum Actinobacteria, *Mycobacterium* includes saprophytic species, which get energy from decaying organisms, and pathogenic species, which cause disease. While some *Mycobacterium* cause tuberculosis and leprosy, the large majority of bacteriophages in our sample infect *Mycobacterium smegmatis*, an advantageous research organism due to its quick growth. In particular, scientists who studied the shared genes between *M. Smegmatis* and other pathogenic strains of *Mycobacterium* were able to discover some of the protein functions.[7] Mycobacteriophages have been shown to have highly mosaic genomes, horizontal genome transfers, and a large GC content range, from 57.3 to 69 percent.[8]

2. Digitization of Bacteriophage Genomes

In 1960, Robert S. Ledley founded the National Biomedical Research Foundation with the motivation of introducing computers to biomedical research. This prompted chemist Dr. Margaret Oakley Dayhoff to organize the few protein sequences available at the time onto a computer-readable punch card. After compiling about 70 sequenced proteins, Dayhoff published the first comprehensive, computer-based collection of protein sequences: *Atlas of Protein Sequence and Structure*. This collection pioneered the use of computers in biology, consolidating the scattered printed literature into a single, digitized collection of data. Additionally, the collection was adopted as an open-ended tool, which invited researchers to submit their own sequences and corrections. As sequencing methods continued to improve, and DNA sequences became increasingly available, Dayhoff eventually attained the largest collection in the world. Her collection ultimately evolved into GenBank, which is the most widely-used public database of nucleic acid sequences today.[9]

In the 1960s, managing Actinobacteriophage genomes with solely GenBank and local spreadsheets was relatively simple because there were only thirteen mycobacteriophage genomes. However this became more difficult with the increase in the number of sequenced genomes. In 2003, the Phage Hunters Integrating Research and Education (PHIRE) program was developed by Graham Hatfull at the University of Pittsburgh to get undergraduate students involved with research. Students isolate, purify, and sequence the genome of bacteriophages before using computational tools to annotate the sequenced genome. As the program grew, it became Science Educational Alliance - Phages Hunters Advancing Genomics and Evolutionary Science (SEA-PHAGES) with the support of the Howard Hughes Medical Institute (HHMI). In the last decade, thousands of bacteriophage genomes have been annotated.[10]

As the field of bacteriophage genome analysis developed, several issues involving software accessibility arose.[11] Phamerator, a genome exploration tool created by Dr. Steve Cresawn, allowed scientists to publish databases that compared various genomes; however, these databases required a series of scripts, making it very difficult for non-technical users to utilize them. In 2010, PhagesDB was created as a way to organize and store information on the hundreds of novel phages isolated annually in undergraduate teaching labs from SEA-PHAGES, PHIRE and satellite programs. In 2015, it was further developed into the Actinobacteriophage Database, which included bacteriophages that infect hosts of the Phylum Actinobacteria.[12]

3. Previous Research on Environmental Factors Affecting Bacteriophages

In a research study aimed at improving the biocontrol of *Listeria monocytogenes,* researchers investigated the stability of bacteriophage-bacteria interactions with respect to the environment in which the lytic cycle of bacteriophages occurs. *L. monocytogenes* and bacteriophages alike vary significantly with respect to their stability in non-native environments because of factors such as UV light, pH, and ionic strength of the surrounding environment. Since bacteriophages have a high mutation rate and simple genetic data, they adapt quickly to new environments. This has led to many distinct species of bacteriophages, all inhabiting different climate conditions.[13]

Genetic mutations result from changes in the nucleotide base sequences or chemical additions to the bases that may prevent DNA replication or transcription. One major mutagen that damages genetic material is solar ultraviolet, or UV, radiation. When UV light hits the DNA, it causes a structural change at any position in the genetic sequence where there are two consecutive thymine bases. The energy from the UV light mutates the chemical bond between the thymine bases, forming a thymine dimer. This causes the two bases to stick together and disfigure the normal structure of the DNA. As a result, this prevents the cell from properly reading the genetic sequence during processes such as DNA replication or transcription. Up to one hundred dimers can form every second a cell is exposed to UV light. Cells that accumulate too many dimers will eventually fail to function properly.[14] This can affect bacteriophages since once host cell organisms are damaged, they are no longer able to provide a mechanism for phages to replicate and transcribe their own DNA.

Another distinct environmental factor that can affect DNA is temperature. The susceptibility of a cell to temperature-induced mutations can vary depending on its GC content, which is the frequency of guanine and cytosine bases in its genetic sequence. While adenine and thymine are connected by two hydrogen bonds, guanine and cytosine pairs have three. The extra bond results in a higher denaturation temperature, so GC-rich genetic sequences are more resistant to hotter temperatures.[15] For example, in Polymerase Chain Reactions, sequences with higher GC frequencies are chosen to help perform the process of amplifying coding segments because they are less likely to denature during the temperature oscillations.

Another item to note is that bacteriophages adopt certain traits under extreme conditions. For example, in a study by Ackermann et al., long and flexible tails in bacteriophages were suggested to be linked to thermal stability. Under tests with extreme conditions, such as large temperature fluctuations, bacteriophages under the *Myoviridae* group were found to survive through staying in the biofilm that was formed by bacterial hosts. Therefore, it was determined that temperature, particularly at the extremities, has some effect on the morphology of a bacteriophage and would hence affect the genomic makeup.

While there is limited data on the relationship between bacteriophages and the environment, bacteriophage survival has been correlated with weather. The ability for bacteriophages to survive under unfavorable conditions is dependent on the ability of their host to adapt to environmental variables such as temperature, acidity, ions, and bacteriophage persistence. Temperature is a key factor in bacteriophage survivability, as it determines occurrence, viability, and storage of bacteriophages. At lower temperatures, less bacteriophage genetic material penetrates into bacterial host cells, and hence fewer bacteriophages can replicate. At higher temperatures, the length of the latent stage can be prolonged. Moreover, as a result of the adaptations of their host, some bacteriophages can be resistant to unfavorable physical and chemical factors, such as low and high temperatures, pH, salinity, and ions and can settle in extreme environments.[16]

B. Purpose

The diversity of bacteriophages is astounding; there are currently estimated to be more than 10^{31} bacteriophages on the planet,[17] which mutate at a rate of around 0.003 mutations in each round of cell division.[18] While bacteriophage genomes are becoming more thoroughly documented, there is still little research on the relationships between bacteriophage genomes and environmental factors. Discovering trends using known genomic and climate information could make it easier to predict the genome of bacteriophages based on their environment rather than individually collecting and sequencing their genes.

Eventual prediction of particular traits in an unknown phage could streamline the process of applying the phage to a problem in industry such as phage therapy for antibiotic resistance[19], plant disease control, and agricultural disease control. More information on their genomes and climate could help create more effective vaccines, detect pathogenic bacterial strains, and advance biotechnology.[20]

C. Tools and Third Party Modules

1. Scikit-Learn: Clustering Algorithms

Scikit-Learn provided us with clustering algorithms, which organized our data points into similar groups based on specified variables. We used this module for k-means clustering analysis as well as the affinity propagation clustering technique. Built with a foundation of NumPy, SciPy, and matplotlib, these algorithms utilize unsupervised learning to cluster data.

2. Scipy: Statistical Analysis

The Scipy module provided us with the proper documentation to use tests of homogeneity to derive statistical analyses of our data. This module, along with Python's NumPy, utilized functions that returned the test statistic, p-value, and expected contingency table of each dataset.

3. Statsmodels: Statistical Analysis

This module provided us with documentation which allowed us to perform linear regression t-tests to statistically evaluate our data. With the help of matplotlib, numpy, and __future__, statsmodels enabled us to determine the least-squares regression line for two specified variables. It also returned the R-squared value, p-value, and test statistic for each dataset.

4. Biopython: Extracting Protein Sequences Using SeqIO[21]

The Biopython Python API is composed of a plethora of computational biology tools, which aid in the manipulation and comparison of biological objects and mechanisms. We specifically used Biopython's SeqIO to parse through an assorted file from GenBank's database.

5. NOAAHist: Extracting Temperature Data

The NOAAHist Python API includes several processes to parse through NOAA's categorical weather data and to find weather data stations, including its main data fetching algorithm, noaahist.py. Using NOAA's main program, one can fetch hourly data given a coordinate location or zip code, as well as a time domain. Primarily, the coordinate locations were found to be too specific for the locations of the weather statements; they had to be first translated to zipcodes using a separate API. Still, if data could not be obtained for some bacteriophages, we used a separate NOAAHist program called "Find Stations" to gather weather stations close to a given location. If adjacent stations were not found, we would not include that bacteriophage in the overall data set.

6. Zipcode: Converting Coordinate Locations to ZIP Codes

The Zipcode Python package allows users to identify specific coordinates based on a standard US ZIP code, or vice-versa. This module was instrumental to the extraction of weather data on specific bacteriophages as we needed to interface between the bacteriophages whose location information was in latitude and longitude format—and the NOAA database—which required ZIP-based input.

7. Matplotlib: Data Visualization

Matplotlib was utilized to visualize the k-means clusters through a scatterplot and histogram.

8. Difflib: String Comparison

Difflib was used to calculate the difference between strings. Specifically, it was used for the comparison of the string representations of protein sequences.

D. Databases

1. Actinobacteriophage Database (PhagesDB)

In 2010, Drs. Graham Hatfull and Roger Hendrix developed the Actinobacteriophage Database in the Pittsburgh Bacteriophage Institute. Their intention was to create an accessible way to centralize and store bacteriophage information. The data on the Actinobacteriophage Database comes largely from satellite programs, mycobacterial genetics courses, the Phage Hunters Integrating Science & Education Program (PHISE), and the Science Education Alliance (SEA) at Howard Hughes Medical Institute. It works with GenBank to annotate gene sequences for every bacteriophage. The MySQL database began by simply grouping bacteriophages into three domains: Phages, Clusters, and Subclusters. In time, its grouping set has grown to include Institutions, Hosts, Publications, Phams, Protocols, Genes, and Documents as well.[22]

Each of the over 8,000 bacteriophages is well-documented with its discovery, sequencing, and characterization details. This information specifies the bacteriophage's GPS coordinates, discovery year, isolation temperature, host bacterium, genome length, GC content, cluster, and morphotype. Many of the bacteriophages in the database also contain photographs taken from the institutions that discovered them. Photographs include plaque pictures and restriction enzyme digest images from gel electrophoresis (Figure 3).

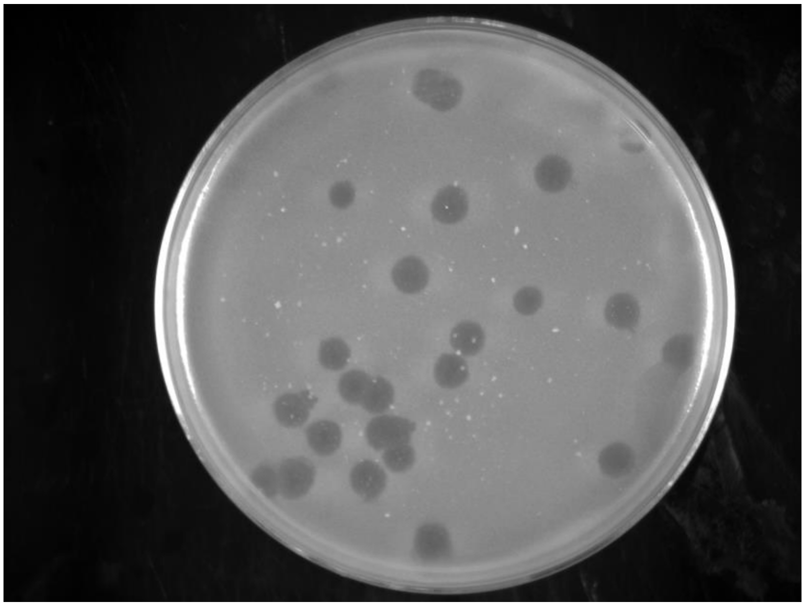

Figure 3: Plaque Picture (from PhagesDB entry for bacteriophage Moose)[23]

The extensive documentation of phages on PhagesDB allows bioinformatics researchers all over the world to access information on numerous aspects of the phages' genomics, from visual information like the image above to numerical data such as the files submitted by the researchers to GenBank, which were particularly pertinent to our purposes.

2. National Oceanic Atmospheric Administration (NOAA)[24]

The National Oceanic and Atmospheric Administration (NOAA) is a scientific organization created in 1970 under the United States Department of Commerce. It documents the conditions of the country's major waterways, oceans, atmosphere, and abiotic systems. Moreover, NOAA collaborates with the National Weather Service to conduct weather forecasts. The organization prides itself on its five "fundamental activities": engaging the public, managing resources, and observing, interpreting, and assessing Earth's systems. Its documentation is stored in an accessible online database, composed of current and historic data. In our project, we used the quantitative and qualitative attributes stored across the database to assess the climate in which our various bacteriophages were discovered. In particular, we focused on obtaining information about maximum, minimum, and average temperatures, as well as average and hourly precipitation and average air pressure for a particular location and time.

3. GenBank[25]

The GenBank database, maintained by the National Center for Biotechnology Information, provides access to information on the sequenced genomes of a wide range of organisms and viruses, including bacteriophages. Sequences of specific genes, their amino acid translations, and annotations describing gene functions are available on the GenBank database. The bacteriophages used in the project were limited to those that were annotated, found in the United States, and known to infect the *Mycobacterium* host because most of NOAA's stations are in the United States and we limited this project to one host. Digital visualization of the GenBank files allowed the gene's function and identity to be projected into the context of the complete bacteriophage genome. The SnapGene Viewer software is used to visualize the contents of the GenBank file as pure nucleotides, amino acids, and genes to be transcribed, as seen in Figure 4 below.

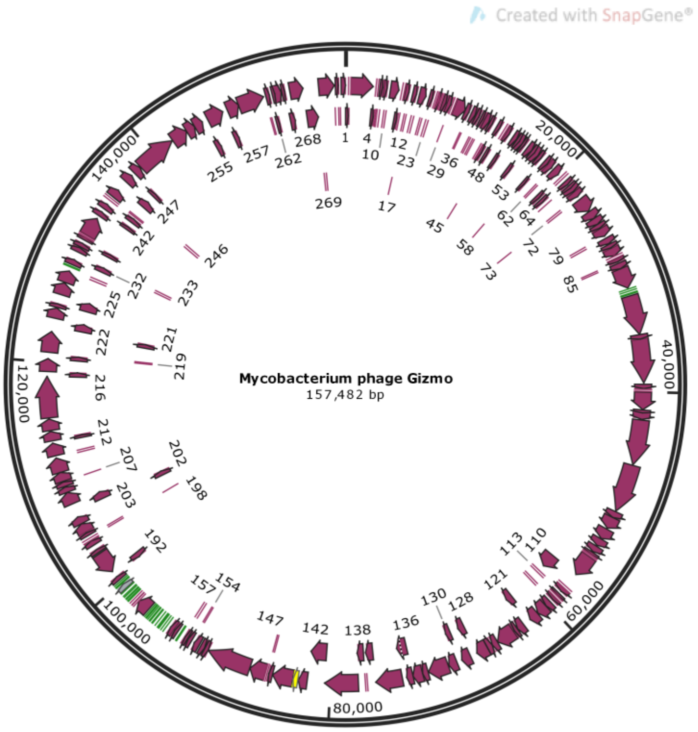

Figure 4: GenBank File seen from SnapGene Viewer as Genome, Sequence, and Features[26]

Digitization of bacteriophage genomes allows us to visualize each gene at every level, from as small as nucleotides to as large as the entire genome. Software like SnapGene allows us to see the bacteriophage in the context of a larger viewpoint than that of the data recorded in the PhagesDB database. Because bacteriophage genomes are difficult to visualize in their pure sequence form, above is a graphical representation of the file we parsed from PhagesDB for one particular bacteriophage called Gizmo. One gene can be seen from a yellow arrow in the circular genotype, expanded out to the nucleotide and amino acid level on the top right, and recorded in the “feature” format of the file at the bottom right. This particularly chosen bacteriophage has its own unique GeneID, and its function to the bacteriophage is to form a tail lysozyme, which breaks down the cell walls of the bacteriophage host.

II. Designing the Analytics Pipeline

The objective of our research project was to look for relationships between the environment of a bacteriophage and its genetic content. To accomplish this, we first needed to gather both environmental and genetic information for each phage. The environmental data, gathered from the NOAAHist module, included information such as the temperature, sky coverage, precipitation, and barometric pressure for each phage’s location of discovery. The genetic data included information for each phage such as GC content and the particular proteins found in each of them, which was obtained from GenBank and PhagesDB. Once both areas of data were collected, they were organized within a single comma-separated values (CSV) file.

Some of the statistical tests that we planned to implement required the grouping of bacteriophages by similar climate conditions or similar genetic contents. In order to make associations between phages with corresponding data points, we implemented two clustering algorithms: k-means clustering and affinity propagation clustering. By clustering the bacteriophages into groups with similar properties, we were able to investigate possible correlations by performing statistical tests, including chi-squared tests of homogeneity and linear regression t-tests.

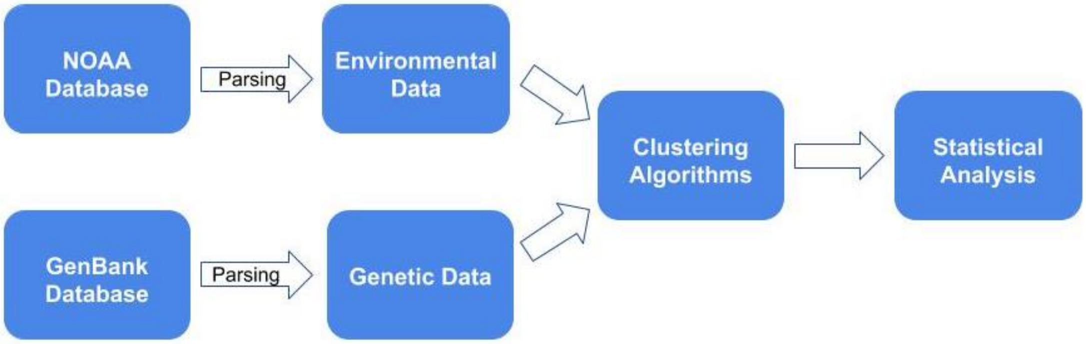

Figure 5: The Pipeline

A. Parsing Input Data

Before we could cluster the data for analysis, we had to parse through a variety of files obtained from the aforementioned databases. Parsing the files allowed us to store the data in an easily accessible format that was compatible with all parts of our pipeline. This approach allowed for greater procedural efficiency by enabling us to access the databases only once, rather than every time that we needed information about a particular bacteriophage. We were able to access file types including .fasta, .txt, .csv, and .gbff, utilizing a combination of Python's basic "open" function and parsing methods built into various third-party modules.

1. Parsing Environmental Data

We limited our PhagesDB search to only bacteriophages found within the United States, because NOAA has the most number of weather stations in that country. Our sample also only included Mycobacteriophages with annotated sequences in GenBank. These criteria decreased the number of phages in our sample from the 2574 from PhagesDB to 804 phages.

After obtaining our sample, in order to parse the NOAA database for each bacteriophage's environmental data, we needed to access the time and location of each bacteriophage. This information came from a tab-delimited CSV file on the PhagesDB database. PhagesDB provided us with each bacteriophage's longitude, latitude, and year founded. This information was then passed into noaahist.py's input file. The file required the following values: the bacteriophage name, year found, location found- in longitude and latitude-, and the data parameters. The table below displays an example of the input file format for three bacteriophages.

Table 1: Sample of noaahist.py Input File

244|20040101,20041231|40.444259,-79.953208|MAX,MIN,SKC,STP,ALT,PCP01
ABCat|20110101,20111231|41.33,-72.948|MAX,MIN,SKC,STP,ALT,PCP01
Abdiel|20110101,20111231|38.643888,-90.311944|MAX,MIN,SKC,STP,ALT,PCP01

A list of all weather information for the bacteriophages was printed to an output file through the following linux code, where infile.txt is the input and all.csv is the output.

```
python2 noaahist.py -i infile.txt -p -o weather/all.csv
```

Table 2 displays the output of this code with all available fields.

Table 2: Sample of infile.txt

NAME,HR_TIME,LAT,LON,MIN,MAX,PCP01,SKC,STP,ALT
Adephagia,2009010111,33.21,-97.15,24,34,0.00,CLR,995.2,30.08
Adephagia,2009010117,33.21,-97.15,32,54,0.00,CLR,990.9,29.95
Adephagia,2009010123,33.21,-97.15,55,69,0.00,CLR,986.2,29.81

The output of this file, after it was organized into an accessible format, was placed into a CSV file, containing PhagesDB data.

There were many obstacles in obtaining environmental information from NOAA. One such obstacle was the fact that the NOAA historical weather program was not compatible with the current Python update; it was only accessible using Python 2.7, whereas Python is currently updated to the version Python 3.6. We needed to code using Python 2.7 in order to circumvent this issue. Moreover, we needed to streamline the data fetching process so that thousands of data points could be gathered in a single sitting; this was mandatory, as our time budget for the research paper was limited. We thus needed to remove keyboard input, which introduced human reaction time and human error, from this process. When the NOAA program was unable to locate a weather station for a given location and year, the user was prompted with a notification that stopped the program from running; we bypassed this notification by passing in a boolean “True” to it which allowed the NOAA program to continue. Furthermore, problematic location inputs were added to a list which enabled our python program to throw an exception when its contents were accessed later. This exception indicated to the NOAA program to not fetch information for those specific locations.

Another problem involved a particular function within the NOAA historical data program that searches for nearby weather locations when one cannot be found given the ZIP code or coordinates. We first attempted to gather weather data using ZIP codes. PhagesDB provided us with the city and state of each phage which we then converted into a ZIP code using Zipcode, a Python API. However, the NOAA program that accessed weather data stations did not check for nearby weather stations if the inputted ZIP code did not exactly one. It did, however, check for these stations if we inputted latitude and longitude coordinates in place of a ZIP code. We thus did not make use of our ZIP code location data.

Beyond these implementation problems, there were human programming errors that stalled the collection of data. One example is the lack of consistent formatting in PhagesDB. Some fields, like city, state, country, annotation status, cluster, and coordinates, created complexities when extracting and comparing the data. For example, the longitudes and latitudes were stored in three different forms, but were only accepted by NOAA as one. This was solved by implementing code that converted each of the wrong forms to the correct one.

2. Parsing Genetic Data

In order to obtain the bacteriophages’ genetic information, we needed to access both the GenBank and the PhagesDB databases. The PhagesDB database provided us with the GC content of each bacteriophage, whereas the GenBank database gave us the information regarding bacteriophage proteins. We parsed files from each of these databases to organize our own .csv files, which mapped bacteriophage names to their corresponding genetic information.

In order to extract information from GenBank, we first needed to gather the relevant files that the database provided to us. We concatenated five pertinent files, which stored bacteriophage protein sequences, along with the protein names and descriptions of their functions. We then parsed these files, which were in a GenBank Flat File format, using the SeqIO method provided by the Biopython API. We were able to store this data in a CSV file, created using a csvWriter. This file included the Protein IDs for their specific proteins, as well as the protein sequences. The code responsible for the creation of this file is in the sample below.

```
layout = ['GeneID', 'Description', 'Protein Sequence']
writer = csv.DictWriter(r, fieldnames=layout)
writer.writeheader()
```

```
…
proteinSeq = seq_record.features[x].qualifiers['translation'][0]
proteinID = seq_record.features[x].qualifiers['protein_id'][0]
note = ""
if str(seq_record.features[x]).find("note") != -1:
     note = seq_record.features[x].qualifiers['note'][0]
writer.writerow({‘ProteinID’: proteinID, 'Description': note, 'Protein
Sequence': proteinSeq})
```

A sample of this file is depicted in Table 3.

Table 3: Sample of GenBank Information by Bacteriophage

ProteinID	Description	Protein Sequence
AAQ06663.1	Peptidoglycan hydrolase	MSNSIALKRSYGVDVASYQSTTVNYAGAKFAFVKL TEGTEYTNPKAEAQIKSAKAHGL…VTGNRAYIVL
APD20970.1	gp1	MVVAAPDVVNIGCLPEATGTLNLTLVARCFEHGLP VVLVPVRREPAGAVGGAPAPPRI
APD20971.1	gp2	MGERGPIGKRSDQRVRRNKTDNPVTKLPARGPV KQPQIGIPDAHPVVTQLWDSLAH…QLGAQQRSG

Originally, our intention was to count the frequency of each gene across all bacteriophages utilizing the gene IDs from PhagesDB; however, because the gene IDs were taxon-specific, we instead used amino acid sequences, which could easily be compared across multiple taxa.

In general, our approach to obtain the counts for the occurrences of each protein was to read a sequence of amino acids from the GenBank files and map them to their corresponding annotated function and name. Afterwards, the sequences were compared to those already existing in the dictionary; if they had not previously been stored, they were added; and, if they had been previously stored, their frequency was incremented. Additionally, the name of the bacteriophage in which the protein was found was recorded in a dictionary mapped to the sequence. However, this approach presented an important limitation of the algorithm’s functionality: protein sequences were simply compared and selected as similar sequences if they were exactly alike. This prevented the program from taking into account missense mutations that cause negligible changes to the function of proteins and still resulted in similar phenotypes for the bacteriophages. For this reason, instead of using GenBank amino acid sequences to map the bacteriophages from the database to their genetic data, we tried utilizing BLASTp to compare the nucleotide sequences themselves and quantify the degree of correlation between sequences found in the results.

After the CSV file was created from the GenBank files, it was passed to a program that both compares proteins and gathers the counts of similar proteins for the statistical analysis. Using protein BLAST (BLASTp), the comparison of two protein sequences yield one e-value, which describes the number of matches expected to appear by chance. These e-values are weighted depending on the length of both proteins and their similarity. The e-value similarity threshold is 10^{-28}, which means any e-value less than that will be added to a list of similar protein sequences. The following code runs the Python BLASTp protocol.

```
 blast_records = NCBIXML.parse(result_handle)
```

Once a record of BLASTp data from allProteins.fasta had been stored in blast_records, the following code determined whether the matches are significant based on the e-values. Each protein sequence was fetched as queryNum and its e-value was compared to 10^{-28}.

```
for blast_record in blast_records:
     queryNum = blast_record.query_id[6:]
          for blast_alignment in blast_record.alignments:
               subjectNum = blast_alignment.accession[8:]
               if (queryNum != subjectNum and
blast_alignment.hsps[0].expect < pow(10, -28) and int(subjectNum) >
int(queryNum)): #if query is unique and phages are well enough correlated
                     next_similar[queryNum] = subjectNum
```

Because every protein had to run comparisons across every other protein, this process would have taken too long given the time we were allotted.

Another technique that was used to identify proteins across phages was fuzzy string comparison, a technique for identifying similarities between strings. In order to expedite our research, we initiated this process by identifying gene sequences that were already perfect matches. This established a baseline set of protein sequences that was known to be expressed across multiple phages. Afterwards, the baseline sequences were compared to all of the other sequences using DiffLib's SequenceMatcher, which scores the similarity of the strings from zero to one. After some experimentation, it became clear that a similarity score greater than 95% would be more than sufficient to consider two protein sequences to be a match.

Utilizing clustering data, we created lists that included the bacteriophage contents of each cluster. From these, each bacteriophage and its cluster could be mapped to the protein sequences it contained, and correlations could be drawn from the numeric contents of the cluster mappings. All of the protein information was placed into a CSV file, containing GenBank data.

3. Combining Parsed Data into a Single File

At this point in our environmental genetic information parsing process, we had both a PhagesDB .csv file and a GenBank .csv file. It was essential to manually create a third .csv file that contained the information of the bacteriophages common to both the GenBank .csv file and the PhagesDB .csv file; this would ensure that we would have a single set of bacteriophages to which we could apply our clustering algorithms. This process reduced the number of bacteriophages that we could analyze, as many bacteriophages in one .csv file were not contained in the other.

In order to collect all of our data points for every bacteriophage, we needed to parse through our genetic information CSV file as well as our environmental information CSV file. We organized that information into a final, comprehensive file from which we were able to cluster bacteriophages. Table 4 depicts this complete CSV file.

Table 4: Sample of Simple Bacteriophage Information

Phage Name	Max High	Min Low	Average High	Average Low	Average	GC Content
244	87	1	56.16131	46.83754	51.49943	63.4
Abrogate	88	12	58.97619	37.14286	48.05952	63.8
ABU	85	-10	51.60425	42.1403	46.87228	66.5

B. Clustering

In order to identify correlations between environmental factors and bacteriophage genomic information, the bacteriophages needed to be arranged into groups, or “clusters,” based on the climate in which they were found. The environmental data for the year and location of the acquisition of each individual bacteriophage had been obtained and organized in the database. Similar environmental attributes could then be grouped together, resulting in clusters of bacteriophages that shared climate traits. Later, the bacteriophages were clustered in accordance to genetic data pulled from the GenBank and PhagesDB.

In order to be confident that our clusters were feasible, and not just dependent on an algorithm-specific criteria, we used two clustering techniques to group the bacteriophages.

1. K-Means Clustering

The first clustering method employed was k-means clustering, an algorithm in which a data set is separated into a user-specified number of clusters, *k*, with as little variance as possible within each cluster. This mechanism was selected because it works well for large amounts of data[27] and produces tighter clusters than other similar methods.[28]

The overall goal of the k-means algorithm is to reduce the inertia for each cluster. The inertia, also known as the within-cluster sum of squared criterion, measures how coherent the values within a cluster are. Each cluster is described by its mean μ_j, so the inertia of each cluster is determined by the square of the differences between each data point in the cluster and μ_j, in the equation:

$$\sum_{i=0}^{n} \min_{\mu_i \in C} \left\| x_j - \mu_i \right\|^2$$

In determining these clusters, the algorithm first identifies *k* samples from the data set to use as initial centroids. Then, each datapoint is assigned to its nearest centroid, and new centroids are calculated by finding the means of the data points associated with each initial centroid. This centroid re-identification is repeated until the differences between the previous and new centroids are less than a threshold, or when the centroids do not shift significantly. The values closest to each centroid then form clusters with minimum inertia values.

To use this clustering method with the climate data, we established two arrays: one containing the bacteriophage names (see Figure 6) and the other containing the environmental data. For instance, in one experiment, the second array included the average temperatures for the years and locations where each bacteriophage was found (see Figure 7). We then inputted the environmental array and the chosen number of clusters into the k-means function. The output was a list with the cluster number for each datapoint (see Figure 8), which was organized into a dictionary with the cluster number as the key and an array of all bacteriophage names in that cluster as the value (see Figure 9).

```
['Nyxis', 'Oosterbaan', 'Toto', 'Oline', 'BAKA', 'RonRayGun', 'KLucky39',
'Marvin', 'McFly', 'OkiRoe']
```

Figure 6: Sample Name Array

```
['66', '64', '55', '66', '62', '64', '39', '53', '57', '69']
```

Figure 7: Sample Environmental Data Array for Average Temperature

```
[0, 0, 1, 0, 0, 0, 2, 1, 1, 0]
```

Figure 8: Sample K-Means Output for Average Temperature and k=3

```
{0: ['Nyxis', 'Oosterbaan', 'Oline', 'BAKA', 'RonRayGun', 'OkiRoe'], 1:
['Toto', 'Marvin', 'McFly'], 2: ['KLucky39']}
```

Figure 9: Sample Dictionary for Average Temperature and k=3

The k-means algorithm clusters the data into different groups according to similar characteristics. To visualize the size of each cluster, we plotted each cluster against the number of datapoints it contained (see Figure 10). The most ideal clustering has few clusters with a small number of bacteriophages.

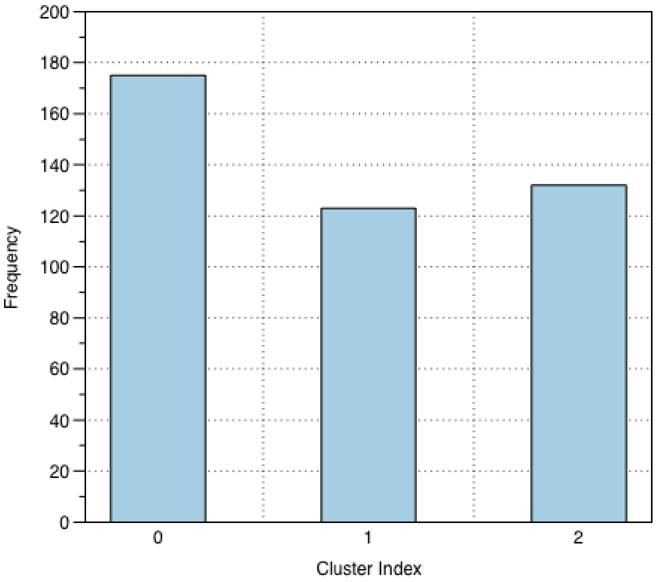

Figure 10: Sample Histogram for Overall Average Temperatures

The k-means function can also be applied to environmental data sets of more than one dimension, such as the global maximum and global minimum temperatures for the location and year where the bacteriophage was found (see Figure 11).

```
[[  95.   23.]
 [  99.   30.]
 [  95.    7.]
 [  98.   18.]
 [ 102.   13.]
 [ 110.  -74.]
 [  82.   -2.]
 [  93.    2.]
 [ 103.   13.]
[  92.   29.]]
```

Figure 11: Sample 2-D Environmental Data Array for Overall Max and Min Temperatures

This array can then be clustered by the k-means function, whose output can be translated to a dictionary of clusters and bacteriophage names. The data points can be graphed and color-coded based on their cluster in order to visualize the results of the k-means clustering, as demonstrated in Figure 12.

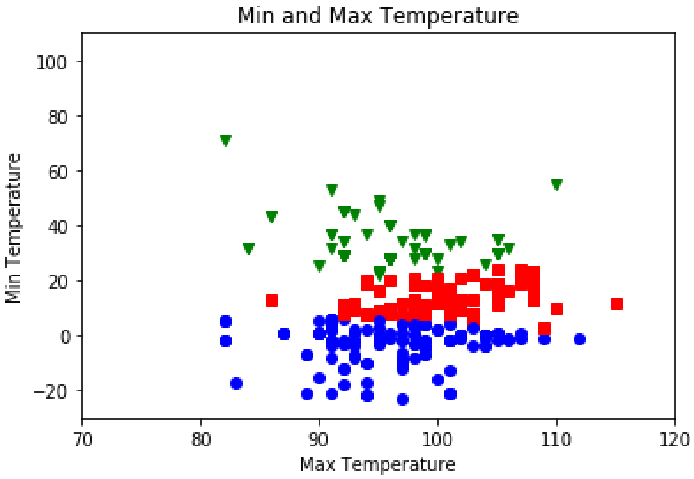

Figure 12: Scatterplot of Sample K-Means Output for Max/Min Temperatures, 3 Clusters

Although 2-dimensional clustering was effective, the input of absolute maximum and minimum temperatures resulted in the same clusters as when we used a 1-dimensional input of either of the variables individually. Thus, we decided to only use 1-dimensional inputs for the environmental factors.

Finally, to confirm that the clusters were non-overlapping and represented similar climates, we created a chart with the highest and lowest points for the given environmental factor within each cluster. For example, Table 5 demonstrates the highest and lowest overall average temperatures for the locations at which each phage was found. Note that the clustering numbering begins at 0.

Table 5: Data Range within Each of Three Clusters for Overall Average Temperatures

	Min Temp in Cluster	Max Temp in Cluster
Cluster 0	36°F	53°F
Cluster 1	54°F	62°F
Cluster 2	63°F	81°F

In addition to clustering temperature and other environmental factors such as precipitation and pressure, we clustered the GC content. Once the clusters had been assembled and the appropriateness of their contents confirmed, the clusters are ready for statistical analysis. The following code is a component of our main clustering function. After we parsed through our .csv file to obtain an array of a specified environmental variable- in this case, it was the average temperature-, we clustered the array elements. Then, we linked those clusterings back to the bacteriophage names so that the clusterings could be used in a later function that creates the contingency table for our statistical analysis.

```
X = np.array(AveTempArray)
     kmeans = KMeans(n_clusters=numcluster, random_state=0).fit(X)
     clusters=list(kmeans.labels_)
     clusternumbers=[]
     for num in range(0,numcluster):
          clusternumbers.append(clusters.count(num))
     PhageClusters={}
     for d in range(len(clusters)):
          if clusters[d] in PhageClusters.keys():
               PhageClusters[clusters[d]].append(NameArray[d])
          else:
               PhageClusters[clusters[d]]=[NameArray[d]]
     return(PhageClusters)
```

2. Affinity Propagation Clustering

Affinity Propagation provides another method to cluster bacteriophages based off climate and genetic data. While k-means requires the user to input the number of clusters, affinity propagation is able to determine the number of clusters itself. A subset of representative points, known as exemplars, are determined; each exemplar corresponds to a cluster. The previous clustering algorithm, k-means, randomly chose an initial set of exemplars and recursively redefined the set to decrease the sum of squared errors. Affinity propagation, however, considers all data points as potential exemplars. In affinity propagation, messages are transmitted along each pair of points until a high-quality set of exemplars is found.

The basis of affinity propagation relies on quantities known as similarities. A similarity represents how well a data point, *k*, is suited to be the exemplar for another data point, *i.* Generally, the similarity, *s*, is set to be the negative squared error, or negative Euclidean distance, between two values. The method of computing similarities can vary depending on the application of the clustering, and is sometimes set as a log-likelihood, or even set by hand. For our project, the similarity was set to the negative Euclidean distance.

$$s(i,k) = -||x_i - x_k||^2$$

To determine the most effective number of clusters, affinity propagation uses a measurement called the preference. A preference represents the similarity between one point to itself, or *s*(*i*,*i*). In this way, the preference represents the likelihood a point will become its own exemplar. A greater preference implies an increased likelihood that the point would be chosen as an exemplar. If all data points have the same weight or have the same exemplar likeliness, preferences are set to a common, shared value. Because no bacteriophage is weighted more than another bacteriophage and all data points have the same weight in our project, the preference is a shared value. In our project, the shared value was set to the median of all input similarities, which is the standard practice and default value of affinity propagations.[29] In addition to a preference, a damping value is inputted between 0.5 and 1. The damping value acts as a numerical stabilization and indicates the weight of incoming values compared to past values.

Two kinds of messages are passed in affinity propagation. Responsibilities, *r*(*i*, *k*), are first sent from a data point, *i*, to a candidate exemplar, *k*, and indicate how “responsible” point *k* is to serve as the exemplar of point *i* over other exemplars. A responsibility is the similarity between points *i* and *k* minus the largest of the similarities and the availability sum between point *i* and every other possible exemplar. Thus, the responsibility also accounts for other potential exemplars for *i*.

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i,k') + s(i,k')\}$$

The other kind of message, availabilities, *a(i,k)*, are sent from a candidate exemplar to a data point to indicate how effective it would be for point *i* to pick point *k* as its exemplar, considering how well point *k* serves as the exemplar of other points.

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i' \text{ s.t. } i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\}$$

$$a(k,k) \leftarrow \sum_{i' \text{ s.t. } i' \neq k} \max\{0, r(i',k)\}$$

This message-passing continues until chosen exemplars stay constant for a certain number of iterations. Exemplars are the points with a positive self-responsibility and self-availability sum.[30]

Affinity Propagation, like k-means clustering, takes an array of environmental or genetic data as its input. Our code outputs a dictionary, where cluster numbers correspond to bacteriophage names. The following function implements the clustering algorithm:

```
from sklearn.cluster import AffinityPropagation
def affinitytest(array):
     af = AffinityPropagation(damping=0.7,preference=median).fit(array)
     clusters = list(af.labels_)
     ...
     for d in range(len(clusters)):
          if clusters[d] in PhageClusters.keys():
               PhageClusters[clusters[d]].append(Name[d])
          else:
               PhageClusters[clusters[d]]=[Name[d]]
     print(PhageClusters)
```

For our project, as with the k-means clustering algorithm, we established two arrays: bacteriophage names and environmental data. In one trial, the environmental data array included the average temperatures for the years and locations where each bacteriophage was found. The output was a dictionary with the cluster number as the key and the bacteriophage names in that cluster as the value. The following histogram displays an example Affinity Propagations trial with the number of clusters and number of phages in each cluster.

C. Statistical Analysis

The data gathering and clustering techniques used previously to this point have yielded two things: information about the genomes of each bacteriophage, and groupings of bacteriophages by environmental factors. In order to analyze the correlations between these two domains, we applied various statistical analyses. Our objective was to discover associations between various genetic attributes and the environment in which the bacteriophages were found.

1. Tests of Homogeneity

a) Purpose

The objective of a test of homogeneity is to determine whether two or more groups are likely to have come from the same population. More specifically, the test investigates if a specified categorical variable has the same distribution across the groups. The null hypothesis assumes that the categorical variable is distributed in the same way across each group. The alternate hypothesis assumes that the categorical variable is distributed differently across each group. Our test outputs a p-value, which represents the chance that a random sample will yield a test statistic as extreme or more extreme than the test statistic of our data, assuming that the null hypothesis is true. We have predetermined a critical value of $\alpha = 0.05$ for our experiments. We will only reject our null hypothesis if the p-value is lower than or equal to α.

b) Usage

In the context of our study, the groups used in the tests for homogeneity were those obtained by the various clustering algorithms. On a high level, we wanted to determine if certain genetic attributes were distributed in the same way across different environmental clusters. For example, after grouping a random sample of annotated bacteriophages by the average temperature in their environments, we calculated the count of a particular protein for each group. Then, we applied the test of homogeneity to determine if the difference in protein count across each cluster was significant.

c) Method

In order to conduct these experiments, we first clustered the bacteriophages by a specific environmental factor, such as average temperature, using the k-means clustering technique. We did so by parsing through our cohesive .csv file to gather relevant data for our particular experiment. We then needed to determine categories across which to distribute the specified variable counts for each cluster. For example, we grouped the GC content into low, medium, and high categories so that we were able to find their distribution across a specific cluster. In order to do so, we used the k-means clustering technique again. After compiling our findings into multiple dictionaries, we utilized those dictionaries to create a contingency table. Its rows indicated each of the clusters and the columns represented the distributed variable. The expected contingency table assumed that the distribution of bacteriophages in each category was uniform. The test of homogeneity compared our observed contingency table to our expected one. The following code implements a test of homogeneity on a contingency table, and prints the statistically significant results for various cluster sizes of both the environmental and genetic factors. We made use of another function, better_contingency, that creates a contingency table after parsing through the .csv files and eliminating clusters that did not include enough phages for statistical analysis.

```
def statstest(numcluster1, namecolumn, envcolumn, numcluster2, gencolumn):
      contingency = better_contingency(numcluster1, namecolumn, envcolumn,
      numcluster2, gencolumn)
      obs = np.array(contingency)
      return(scipy.stats.chi2_contingency(obs)[1])

def significance(namecolumn, envcolumn, gencolumn):
     for i in range(2, 7):
          for j in range(2, 7):
               result = statstest(i, namecolumn, envcolumn, j, gencolumn)
                    if result <= 0.05:
                         print ("Significant", result, (i, j))
```

2. Linear Regression T-Tests

a) Purpose

The objective of a linear regression t-test is to determine whether or not there is a linear correlation across a bivariate data set. As with all such tests, it is necessary to state our hypotheses. Our null hypothesis is that the two sets of data have no correlation, or that the least-squares regression line has a slope of zero. Our alternate hypothesis is that the two sets of data are correlated, or that the least-squares regression line has a slope significantly different from zero. Similar to before, if the test outputted a p-value less than the predetermined α value of 0.05, we concluded that the result was statistically significant. A statistically significant result gives us enough evidence to reject the null hypothesis.

b) Usage

At this point in the pipeline, we had a .csv file that contains both the environmental and genetic data for each bacteriophage. In order to look for relationships between both sets of data, the most natural thing to do is plot them together on a scatter plot. The x-axis would represent some quantitative environmental attribute, like average temperature. The y-axis would represent some quantitative genetic attribute, like GC content. We applied a linear regression t-test on this plotted data.

When we applied the statistical tests on each environmental attribute against each genetic attribute, all of the results were statistically significant. This raised a red flag for us, because of how unlikely obtaining a statistically significant result is in the first place. We revisited the assumptions necessary for applying linear regression t-tests to discover the problem. Some of the assumptions had failed. In particular, the assumption that our data was relatively linear to begin with was not true; most of the coefficients of determination for our datasets were numbers under 0.1. In addition, we had failed to fulfill the assumption of multivariate normality. Thus, it was necessary to ignore any of the results obtained by these tests. The following code implements this statistical test, given two lists which we created from our parsed data.

```
y = np.array([genetic_attribute_list])
x = np.array([environmental_attribute_list])
x = sm.add_constant(X)
model = sm.OLS(y, X)
results = model.fit()
print(results.summary())
```

III. Results

A. Tests of Homogeneity Results

At this point in the pipeline, we had two clustering algorithms and all of the genetic and environmental data for each phage parsed into a single file. With those factors, it was possible to implement the chi-squared test of homogeneity. As described earlier, we used our clustering algorithms on the environmental factors to determine the rows of our contingency table, and clustering algorithms on the genetic factors to determine the columns of our contingency table.

In running the tests of homogeneity, we considered environmental factors such as average temperature, average high temperature, average low temperature, record high temperature, record low temperature, sky conditions, precipitation, and barometric pressure. We considered genetic factors such as GC content and protein occurrence in each bacteriophage.

It is important to note that, for each test of homogeneity, it was important to fulfill certain assumptions. The first necessary assumption is that the group of bacteriophages examined was chosen without bias. Since we ran the tests against the entire population of sequenced and annotated bacteriophages that attack the Mycobacterium host, it was not necessary to randomly sample our data. The second necessary assumption is that each cell of the contingency table has a count less than or equal to five. To avoid this problem, we removed “outlier clusters”, or clusters that had less than fifty bacteriophages in them, from our contingency table. This was acceptable because the excluded clusters contained outlier data. Having a statistically significant result would be more trustworthy if only reliable data points were used.

1. Grouping Environmental Data against GC Content

When using the k-means clustering algorithm, it was necessary to specify the number of buckets into which we would group the bacteriophages. We wanted to avoid the case in which an obscure number of buckets for the environmental and genetic clusters yields a statistically significant result, but other clusterings with the same data does not. To avoid this situation, we tried many bucket combinations for each pair of clusters, looking for significant results amongst more than one of them. Although the tests should be significant even with moderate changes in clusters, it is also important to make sure the cluster ranges make sense. For example, grouping bacteriophages that come from regions that are often below the freezing point should not be grouped with those that come from regions that are often above the freezing point. This verification will help eliminate the effect of confounding variables on our results.

When grouping GC content against either overall high temperature or average low temperature in the location of bacteriophage discovery, the results were statistically significant independent of the number of clusters used. This means that there is a relationship between these temperature attributes and the GC content of bacteriophages. Across all of the different combinations of number of buckets, the average p-value for tests against overall high temperature was 0.00508. Against average low temperature, the average p-value was .0361. This means that in these cases, we have enough evidence to reject the null hypothesis, which is that there is an equal distribution of GC content across different environmental clusters. We hypothesize that this has to do with the fact that guanine and cytosine are bonded by three hydrogen bonds instead of two. This increases the melting temperature of the bacteriophage’s DNA sequence, therefore giving protection to bacteriophages that reside in regions of high temperatures. Because the genome of a bacteriophage is easily mutated and often mutates with accordance to its host, it may also be the case that different temperatures are affecting the genomes of the host bacteria, which are in turn affecting the genomes of the bacteriophages.

We also applied tests of homogeneity comparing GC content against the other temperature attributes: average temperature, average high temperature, and record low temperature. These tests also outputted low p-values, indicating statistically significant results. However, because these attributes did not output statistically significant results for more than a few bucket combinations, these attributes were not as indicative of GC content as the two listed before. In other words, our data is enough to suggest a correlation amongst these attributes, but not enough to confidently state one.

When we applied tests of homogeneity comparing GC content against the average precipitation, barometric pressure, and weather conditions of a bacteriophage’s location of discovery, none of the results were statistically significant. This means that we did not have enough evidence to reject the null hypothesis, which was that there is an equal distribution of GC content across different environmental clusters. The results of running Tests of Homogeneity on environmental attributes with GC content can be summarized in Table 6.

Table 6: Tests of Homogeneity Comparing GC Content with Environmental Attributes

Test #	Grouped By	Compared Across	Average P-value	Statistically Significant?
1	Avg Temp	GC	Over 0.05	Somewhat
2	Avg High Temp	GC	0.0465	Yes
3	Overall High	GC	0.00508	Yes
4	Avg Low Temp	GC	Over 0.05	Somewhat
5	Overall Low	GC	Over 0.05	Somewhat
6	Sky Conditions	GC	Over 0.05	No
7	Barometric Pressure	GC	Over 0.05	No
6	Precipitation	GC	Over 0.05	No

2. Grouping Environmental Data against Protein Counts

In order to investigate whether or not occurrences of certain proteins within a bacteriophage were correlated to the environment in which that bacteriophage was discovered, we ran a second series of statistical tests. We created our contingency tables differently for these tests. Like before, the rows represented bacteriophages grouped with accordance to an environmental attribute, using the aforementioned clustering algorithms. The columns represented “Yes” or “No” occurrences of a particular protein. If a bacteriophage in a particular environmental clustering contained the protein of interest, the cell intersecting the cluster row and the “Yes” column was increased by one. Likewise, if a bacteriophage in a particular environmental clustering did not contain the protein of interest, the cell intersecting that cluster row and the “No” column was increased by one. The significance tests were run against these contingency tables. We ran the tests against the proportion of “Yes” and “No” in each cluster, as the absolute number of samples will vary based on the cluster size.

It would have been difficult to run the statistical tests on the hundreds of proteins prevalent in our bacteriophage population with the time restraint for this project. To avoid this problem, we had to determine which proteins were of interest to us. In order to do so, we ignored proteins that were common to the strong majority of all of the bacteriophages, as it was likely that they had some structural or metabolic function common to all bacteriophages. Although it would be interesting to apply tests on these common proteins, the time restriction on the project forced us to prioritize. We also ignored proteins that were so rare that they only occurred in a few bacteriophages, because the statistical tests that we implemented do not work well as well with such small numbers in each cell of the contingency table. In the end, we decided to only look at proteins that were common to more than fifty but less than the strong majority of bacteriophages. There were thirty such proteins of interest.

We ran tests of homogeneity comparing various environmental attributes with “Yes” and “No” instances of those thirty proteins of interest. The environmental attributes that we examined were the same as before: average temperature, average high temperature, average low temperature, record high temperature, record low temperature, sky conditions, precipitation, and barometric pressure. Again, if the test outputted p-values under α = 0.05, we deemed the result statistically significant. This means that with 95% confidence, we were able to reject that null hypothesis that the protein occurrences were distributed in the same way across each environmental clustering.

As expected, only three of the thirty examined proteins of interest outputted statistically significant results. The three proteins were labeled by our indexes as Protein 335, Protein 346, and Protein 355 because they are unnamed in the scientific community. Because these proteins were identified with fuzzy string comparison, we were unable to determine their exact amino acid sequences. For all three, the results were significant in relation to average temperature, average high temperature, average low temperature, and record low temperature. They were not significant in relation to record high temperature, precipitation, and barometric pressure. This means that for the significant environmental attributes, we could reject the null hypothesis that the protein occurrences were distributed in the same way across each environmental clustering. In other words, the occurrences of these proteins are highly correlated to the environment in which bacteriophages are found.

An interesting comparison of the three aforementioned proteins is that all of them outputted exactly the same results from the statistical tests; for each protein, the p-value, observed contingency table, and test statistic were identical. After further research, we concluded that each of these proteins occurred in exactly the same bacteriophages. This is particularly interesting because they did not appear sequentially within a genome. This supports a hypothesis that these three genes are related to environment and that they both co-occur. It may be possible that they co-occur because of the environmental dependency. It may also be the case that they coevolve together. Information regarding the known function of these particular proteins has not been documented by the scientific community.

IV. Conclusion

A. Summary of Results

We were able to successfully gather and analyze bacteriophage data using our pipeline. After we collected bacteriophage data from NOAA, GenBank, and PhagesDB, we implemented our clustering algorithms on bacteriophages that were documented across each of the databases. Our statistical tests determined that the average low temperature and record low temperature of the location in which a bacteriophage was discovered were strongly correlated to that bacteriophage's GC content. We also were able to recognize that there was somewhat of a correlation between a bacteriophage's GC content and the overall low temperature, average temperature, and average low temperature of their discovery environment. Moreover, we recognized three proteins whose appearances in a bacteriophage correlated strongly to the environment in which the bacteriophages were found. Although the modelling of our pipeline in the form of Python code took us weeks to effectuate, the actual collection, clustering, and statistical analysis of the data only took us hours. This suggests that, given additional bacteriophage information, we would be able to swiftly determine correlations between other factors; our pipeline is easily extendable to perform on further data. Finding relationships between a bacteriophage's genome and the environment can help us further understand and classify the viruses. Certain bacteriophage attributes may be more beneficial in future and real-world applications.

B. Future Work

1. Future Explorations

As a result of the diversity among different bacteriophages in the Actinobacteriophage database, the data collected could be interpreted as a reasonable assumption based on possible observations. However, the vast majority of bacteriophages are largely unknown to researchers, so correlations cannot be decisively determined to be factual. Although there is a large enough and broad enough quantity of data to assume a reasonable degree of randomness, there a slight bias in location. Since the project was first developed in Pittsburgh, there are far more bacteriophages collected there. Additionally, another roadblock is that startup costs exist for institutions to contribute to the database, so most bacteriophage collection occurs at research institutions, providing slightly biased climate data toward large cities.

If we had more time during the experiment, we would have tested more variables from the NOAA database and analyzed their individual effects on genomes. Also, although we found three protein sequences that co-occur, we do not know exactly the sequence or name of them because of our local alignment method used to find similar proteins. All we do know is these three proteins are genetically different and considered hypothetical proteins in GenBank, meaning their function is relatively unknown. Using BLASTp, we would be able to better identify alike proteins and accurately determine their name and sequence, but this would require a lot more time because of the run-time of this approach.

In the future, the algorithm we used to cluster the bacteriophages based on the NOAA climate data could be updated, both as new bacteriophages are added to the PhagesDB database and as average weather data for the locations used shifts over time. Other interesting correlations could be found by including data from bacteriophages sequenced outside of the United States, which could possibly display similar correlations to our data or could display completely different correlations.

Another extension of the algorithms used for this project is to investigate using supervised machine learning. One type of supervised machine learning we could utilize is decision trees where parameters are used to split the data at the decision nodes, which then output outcomes, or leaves. In this specific case, we could use a regression tree which is good for continuous data types, which is in our case, the GC Content. The decision tree would then be built off of multiple variables such as minimum temperature, maximum temperature, cloud covering, average precipitation, and average pressure to then predict the GC content of the bacteriophage.[31]

2. Real-World Applications

Current advances in our understanding of bacteriophages have allowed for the expansion of bacteriophage applications. Such examples include pathogen detection and biocontrol, which uses biological entities to depress a pest, especially for the agricultural industry. Another example is phage therapy, in which bacteriophages are utilized to treat pathogenic bacterial infections, especially helpful in an increasingly antibiotic resistant world. Bacteriophages have been used longer than antibiotics to treat bacterial infections, and recent progress in biotechnology can aid bacteriophage selection and production. Eventually researchers could selectively breed for the traits they want and genetically engineer the bacteriophages to better perform their desired functions. Specific functions of these bacteriophages in industry depend on the types of proteins they produce and the way they behave. Consequently, understanding the correlations between geographic factors and the genomes of their bacteriophages could help predict which bacteriophages could produce these specific desired proteins or behavior, making bacteriophage applications easier, more efficient, and more financially accessible.[32]

C. Acknowledgements

We would first like to thank our advisor, Andrew McGuier, for his continuous and invaluable leadership, support, and motivation throughout the project. The project would not be possible without our Teaching Assistant, Matthew Weidner, who supported us throughout the project. We also greatly appreciate Dr. Natalie McGuier at Carnegie Mellon University for her valuable biology knowledge and mentorship and Dr. Javier López and Mr. Graham Hatfull. We also like to thank, our program advisor, Dr. Barry Luokkala. In addition, we would not have been able to access and manipulate our data as easily if it were not for the open-source projects that we used. We would also like to thank Carnegie Mellon University for providing the computing facilities and other resources. Above all, we would like to give thanks to the PGSS Alumni Association and Corporate Sponsors that made the program possible.

V. References

[1] Alexander Sulakvelidze, Zemphira Alavidze, and J. Glenn Morris, Jr., "Bacteriophage therapy," *Antimicrobial Agents and Chemotherapy* **45**, (3), 649-659, (March 2001).

[2] Eric C. Keen, "A century of phage research: bacteriophages and the shaping of modern biology," *Bioessays* **37**, (1), 6-9, (Jan 2015).

[3] The Editors of Encyclopaedia Britannica, "Bacteriophage," Chicago, Illinois, USA, 19 May 2017, www.britannica.com/science/bacteriophage.

[4] Genetics Education Networking for Innovation and Excellence, "Bacteriophage," Leicester, England, www2.le.ac.uk/projects/vgec/highereducation/topics/microbial-genetics-1/bacteriophage.

[5] Cronodon, "Bacteriophage lambda," May 2018, cronodon.com/BioTech/Virus_Tech_2.html.

[6] Graham Hatfull and Roger Hendrix. "Bacteriophages and their genomes". *Current Opinion in Virology* b1b (4), n.p., (2011 October 1).

[7] Jean-Marc Reyrat and Daniel Kahn, "*Mycobacterium smegmatis*: an absurd model for tuberculosis?," *Trends in Microbiology* **9**, (10), 472-473, (1 October 2001).

[8] Marisa L. Pedulla, Michael E. Ford, Jennifer M. Houtz, Tharun Karthikeyan, Curtis Wadsworth, John A. Lewis, Debbie Jacobs-Sera, Jacob Falbo, Joseph Gross, Nicholas R. Pannunzio, William Brucker, Vanaja Kumar, Jayasankar Kandasamy, Lauren Keenan, Svetsoslav Bardarov, Jordan Kriakov, Jeffrey G. Lawrence, William R. Jacobs, Roger W. Hendrix, and Graham F. Hatfull, "Origins of highly mosaic Mycobacteriophage genomes," *Cell* **113**, (2), 171-182, (18 April 2003).

[9] Bruno J. Strasser, "Collecting, comparing, and computing sequences: the making of Margaret O. Dayhoff's *Atlas of Protein Sequence and Structure*, 1954–1965," *Journal of the History of Biology* **43**, (4), 623-660, (December 2010).

[10] Collaboration with Dr. Natalie McGuier

[11] Bryan D. Merrill, Andy T. Ward, Julianne H. Grose, and Sandra Hope, "Software-based analysis of bacteriophage genomes, physical ends, and packaging strategies," *BMC Genomics* **17**, (1), 679, (26 August 2016).

[12] James G. Lamine, Randall J. DeJong, and Serita M. Nelesen, "PhamDB: a web-based application for building Phamerator databases," *Bioinformatics* **32**, (13), 2026-2028, (1 July 2016).

[13] Susanne Fister, Christian Robben, Anna K. Witte, Dagmar Schoder, Martin Wagner, and Peter Rossmanith, "Influence of environmental factors on phage–bacteria interaction and on the efficacy and infectivity of Phage P100," *Frontiers in Microbiology* **7**, 1152, (28 July 2016).

[14] Carolyn Csanyi, "How does UV light damage the DNA strand?" 25 April 2017, sciencing.com/uv-light-damage-dna-strand-12687.html.

[15] Rodolphe Suspène, Myrtille Renard, Michel Henry, Denise Guétard, David Puyraimond-Zemmour, Agnès Billecocq, Michèle Bouloy, Frederic Tangy, Jean-Pierre Vartanian, and Simon Wain-Hobson, "Inversing the natural hydrogen bonding rule to selectively amplify GC-rich ADAR-edited RNAs," *Nucleic Acids Research* **36**, (12), e72, (1 July 2008).

[16] E. Jończyk, M. Kłak, R. Międzybrodzki, and A. Górski, "The influence of external factors on bacteriophages—review," *Folia Microbiologica* **56**, (3), 191-200, (May 2011).

[17] Scott LaFee and Heather Buschman, "Novel phage therapy saves patient with multidrug-resistant bacterial infection," San Diego, California, USA, 25 April 2017, health.ucsd.edu/news/releases/Pages/2017-04-25-novel-phage-therapy-saves-patient-with-multidrug-resistant-bacterial-infection.aspx.

[18] José M. Cuevas, Siobain Duffy, and Rafael Sanjuán, "Point mutation rate of bacteriophage ΦX174," *Genetics* **183**, (2), 747-749, (October 2009).

[19] Ben Burrowes, David R. Harper, Joseph Anderson, Malcolm McConville, and Mark C. Enright, "Bacteriophage therapy: potential uses in the control of antibiotic-resistant pathogens," *Expert Review of Anti-infective Therapy* **9**, (9), 775-785, (10 January 2014).

[20] Irshad Ul Haq, Waqas Nasir Chaudhry, Maha Nadeem Akhtar, Saadia Andleeb, and Ishtiaq Qadri, "Bacteriophages and their implications on future biotechnology: a review," *Virology Journal* **9**, (9), 1-8, (10 January 2012).

[21] Biopython Contributors, "Introduction to SeqIO," 2018, biopython.org/wiki/SeqIO.

[22] Roger Hendrix and Graham Hatfull, "About PhagesDB," 2018, phagesdb.org/about/.

[23] Rebecca Foreman, "Moose [Photograph]," Washington University, St. Louis, Missouri, 2012, phagesdb.org/phages/Moose/.

[24] National Oceanic and Atmospheric Administration, "NOAA - our history," 2018, www.noaa.gov/our-history.

[25] National Center for Biotechnology Information, "GenBank overview," National Library of Medicine, Bethesda, Maryland, 2017, www.ncbi.nlm.nih.gov/genbank/.

[26] Elyse Borchik, "Mycobacterium phage Gizmo," Illinois Wesleyan University, Bloomington, Illinois, 2011, phagesdb.org/phages/Gizmo/.

[27] Marina Santini, "Advantages & disadvantages of k-means and hierarchical clustering (unsupervised learning)," Uppsala, Sweden, 12 December 2016, stp.lingfil.uu.se/~santinim/ml/2016/Lect_10/10c_UnsupervisedMethods.pdf.

[28] F. Pedregosa et al., "Clustering: k-means," Scikit Learn, 2017, scikit-learn.org/stable/modules/clustering.html#k-means.

[29] Probabilistic and Statistical Inference Group, "Affinity Propagation," http://genes.toronto.edu/affinitypropagation/faq.html#clusters

[30] Brendan J. Frey and Delbert Dueck, “Clustering by passing messages between data points,” *Science* **315**, (5814), 972-976, (February 2007).

[31] Mayur Kulkarni, “Decision trees for classification: a machine learning algorithm,” Sunnyvale, California, 2018, www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html.

[32] Diana P. Pires, et al., “Genetically engineered phages: a review of advances over the last decade,” *Microbiology and Molecular Biology Reviews* **80,** (3), 523-543, (September 2016).