BLAST Your Way through Malware
Malware Analysis Assisted by Bioinformatics Tools

Jay Pedersen, Dhundy Bastola, Ken Dick, Robin Gandhi, William Mahoney
School of Interdisciplinary Informatics
College of Information Science and Technology
University of Nebraska at Omaha
Omaha, Nebraska
{jaypedersen, dkbastola, kdick, rgandhi, wmahoney} @unomaha.edu

Abstract—*As a new strain of computer malware is discovered, it triggers a meticulous process of analyzing its behavior and developing appropriate defenses. A systematic process which identifies regions of commonality and variability with known samples can ease the burden of malware analysis. We address this challenge using an interdisciplinary approach which applies biological sequence analysis methods to computer malware. Specifically, we have developed a method which has the goal of classifying a digital artifact (possibly malware) based on its similarity to known digital artifacts (or known malware samples) using methods and tools of bioinformatics. Our approach is analogous to classifications of biological sequences, which are routinely performed using online databases of known biological sequences.*

Keywords: clustering, classification, malware, plagiarism

1. Introduction

Consider the evolution of a biological pathogen, which can be tracked using DNA markers. This operation often involves examining nucleotide sequences in DNA using powerful bioinformatics tools to identify regions of local or global similarity, or interactions with specific enzymes, which may be a consequence of functional, structural, or evolutionary characteristics of the pathogen's genetic makeup. This is biological stylometry at work!

In field of computer security, there is significant interest in understanding malware behavior to develop effective detection, prevention and recovery mechanisms [1] [2]. Unfortunately, malware analysis is still much of an acquired tradecraft, and the results depend heavily on the quality of personnel involved. Malware analysis typically involves reverse engineering compiled digital artifacts, configuration files, metadata or foraging through other information. The results from such analysis provide clues for malware origin, behavior, locating other variants, signature patterns, and proper malware classification (e.g. Trojan, worm, virus, zombie, fork bomb, bot, etc.).

In the field of bioinformatics, the Basic Local Alignment Search Tool (BLAST) tool discovers areas of local similarity between DNA or protein sequences [5] [8] [10] [14] [15] [16]. Local similarity comparisons have advantages over global similarity in studying the functional and evolutionary relationships among specimens. For example, ape and human DNA shares several areas of local similarity. When compared globally the DNA similarity is harder to discover due to years of evolutionary changes. In particular, BLAST compares a given DNA nucleotide sequences to a database of known sequences which can be from a wide variety of organisms. Based on the results of discovered local similarities, a given specimen can be inferred to have functional and evolutionary relationships with a known sample. Such local alignments help identify related members of gene families.

Computer malware analysis has interesting parallels with the study of biological pathogens. The similarity does not just stop with bio inspired names of computer malware and their high-level behaviors, but also extends to how we analyze and study them. The objective of our work is thus to utilize an interdisciplinary approach to determine the pedigree of a digital artifact of unknown origin. In this paper, we implement bioinformatics inspired methods in the study of three application areas: (a) document clustering using similarity detection (b) rapid malware classification, (c) plagiarism detection.

In the present research we apply BLAST to study synthetic DNA sequences that represent digital artifacts. The ability to use bioinformatics tools to study digital artifacts opens up several avenues of interesting studies ranging from literary stylometry; digital forensics; sources code clone detection; malware functional characteristics and evolutionary relationships; and most importantly attribution of digital artifacts to compilers, platforms, chipsets, versions, and possibly the author!

We assume that the reader is not familiar with bioinformatics tools, and in section two include an overview of BLAST and the technology involved. Our methods for the analysis of digital artifacts are in section three and include the description of how the data is manipulated to appear as DNA to the

bioinformatics tools. The results of several experiments are contained in section four. These results include the three areas described above: malware, file type, and plagiarism. The final section includes our ideas for additional research as well as our conclusions for our work thus far.

2. Bioinformatics Background

DNA is the biological blueprint used for building proteins and other cellular components of living organisms. It is comprised of a long stretch of adenine (A), guanine (G), cytosine (C), and thymine (T) molecules, commonly referred to as "bases" due to their chemical nature. They are also referred to as nucleotides. DNA is represented computationally by character strings containing only the characters A, G, C and T. The seminal paper of Watson and Crick in April 25 of 1953 [3] described the molecular structure of DNA as a double helix. This discovery revolutionized the study of Biology. In double stranded DNA each strand runs anti-parallel to the other and each strand can be used as a template to construct the other strand using Chargaff's base pairing rule [3] [4] which states that Adenine (A) will only pair with Thymine (T), and Guanine (G) will only pair with Cytosine (C).

Each strand has an associated direction, which is indicated by its 5' (5 prime) and 3' positions; the direction is from 5' to 3'. The positions of the 5' and 3' ends of the strands are opposite, and thus they are anti-parallel. The following shows a representation of a small piece of double-stranded DNA. It shows the complementary base pairing and the anti-parallel nature of the strands.

```
5′ GAATTCGGCC 3′
   ||||||||||
3′ CTTAAGCCGG 5′
```

The computational representation of DNA only includes one of the strands; and the other strand is implied and can be computed as necessary using the base-pairing rules. The strand direction is also implied -- the 5' position is at the beginning of the string and the 3' position is at the end of the string.

DNA is the key information source in many bioinformatics research projects, including those trying to determine the relatedness of two organisms by a comparative study. GenBank is an international nucleotide sequence database and currently holds sequences from about 407,000 organisms.

BLAST is a widely used bioinformatics tool [5] [8] [10] [14] [15] [16] that compares a given DNA sequence with other known DNA sequences (e.g. GenBank sequences) that reside in a BLAST database and determines similarities between them. A BLAST database is collection of known biological sequences, optimized for similarity querying by the BLAST tool. The result set returned in response to a given query sequence includes local alignments between the query sequence and "subject" sequences in a BLAST database; an example alignment is shown in Figure 1. A local alignment indicates a region of similarity between two sequences. The regions involved can be in any part of either sequence. Within these regions, every base is aligned to exactly one base in the other sequence or to a gap position inserted between bases in the other sequence. Gaps are introduced to represent deletions or insertions of bases, which may have occurred over time. A local alignment is distinguished from a global alignment, which is an alignment of two entire sequences (rather than alignments of arbitrary regions within two sequences).

For each determined alignment, BLAST returns the name of the query and subject sequence and the positions within the sequences that were aligned. BLAST also returns a statistical measure of the likelihood that the identified alignment is a randomly expected occurrence; this is called the expect value (E-value). An E-value near zero indicates a nearly zero probability that the alignment represents a random occurrence [6]. A BLAST parameter allows you to specify a threshold E-value. Specifying an E-value near zero asks BLAST to return only highly similar alignments.

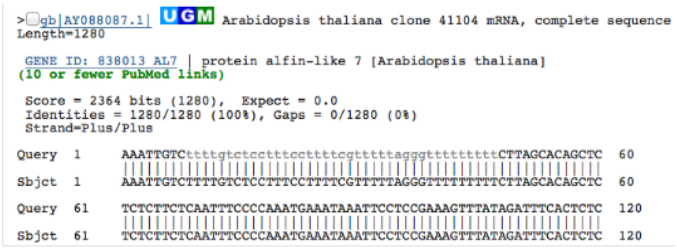

Fig. 1. A BLAST alignment between two highly similar DNA sequences, which has an expect value (E-value) of zero.

3. METHODS

The premise of this project is that a digital artifact may be represented by a "synthetic" DNA sequence and that BLAST should be able to find similarities between that sequence and a set of sequences representing other digital artifacts, which are stored in a BLAST database. (Note that BLAST has previously been used to examine sequences, which do not represent actual biological sequences. A previously reported use was examining journal papers [7])

A. DNA representation of an arbitrary digital artifact

BLAST supports both nucleotide (DNA) and protein sequences. However, BLAST attaches biological significance to the amino acids in a protein sequence (BLOSUM and PAM scoring matrices [12] have this logic encoded in them). On the other hand, when BLAST is analyzing DNA sequences there is minimal scoring logic related to chemical properties of the nucleotides. For this study the DNA format was chosen, so that chemical properties would not be a significant factor in the BLAST analysis. As a result, all digital artifacts are transformed into a corresponding DNA representation for processing by BLAST. This transformation is obtained by

following a mapping between digital bits and characters representing nucleotides.

A digital artifact is considered to be a sequence of byte values. The initial step is thus to convert the sequence of bytes from an arbitrary digital artifact into a DNA representation. The conversion is completely reversible. Four DNA characters are created for each byte in the digital artifact. Each DNA character represents two bits of the byte. The four characters represent bits six and seven, four and five, two and three and zero and one, respectively. The following mapping is used:

```
00 ←→ T
01 ←→ G
10 ←→ C
11 ←→ A
```

There are twenty-four possible ways to perform such a mapping. Any of those mappings could be used to provide a consistent and comparable DNA representation of a digital artifact. This mapping has the property that the values for G and C; and T and A are complementary, considering bit values of zero and one to be complements.

A method was developed which uses this mapping to allow for the comparison and clustering of arbitrary digital artifacts. Steps in the method are guided by the alignments discovered by BLAST among the DNA representations of those artifacts. The method includes three steps: 1) preprocessing, 2) sequence analysis, and 3) visualization. The steps can be applied for several use cases.

Digital artifact clustering: A set of digital artifacts to be clustered is converted into DNA representations. A pairwise comparison by BLAST produces alignments between artifacts with similar structures. The alignments are used to build a clustered graph representation of the similarities between the artifacts. The visualization is performed using Cytoscape (a popular bioinformatics graph visualization tool) [9] [13].

Digital artifact identification: Consider an artifact of unknown origin and a BLAST database of sequences of known digital pedigree. The unknown artifact is converted to a DNA format, and compared to the sequences in the database using BLAST. The resulting alignments are used to identify the most likely type of that artifact. Again, the results are visualized for foraging through the various reported alignments.

A. *Preprocessing*

The input to this step is an arbitrary set of digital artifacts to be examined. The artifacts are converted to a DNA format and a BLAST database is created using the steps shown in Figure 2.

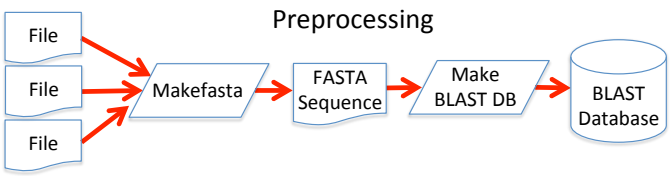

Fig. 2. Preprocessing Step for Digital Artifact Analysis. The various digital files are converted to DNA sequences and are merged into a FASTA sequence file. This is converted into the database used by BLAST.

The specific type of file that is created by this step is a FASTA format file [11]. This is a flat text file, which can contain multiple DNA sequences. Each sequence is introduced by an identification line which has the ">" character as the first character on the line and then has information which identifies the sequence. Each identification line is followed by one or more lines containing the DNA characters which define the sequence. The following shows the beginning of a FASTA file representing the digital artifact "zeus_005_f04.exe":

```
>lcl|/home/jayp/bigtest/zeus_005_f04.exe
GTAGGGCCCGTTTTTTTTTATTTTTTTTTTTTTTGTTTTTTT
TTTTTTAAA
```

Given a FASTA file containing one or more DNA sequences -- the "makeblastdb" tool from NCBI's BLAST tools [15] [16] can be used to create a BLAST database containing those sequences.

B. *Sequence Analysis*

Once the database has been created, the artifacts can be analyzed for pairwise similarity using BLAST. We have used NCBI's BLAST version 2.25 [16] for our analyses. Bioinformatics practitioners call this type of search an "all versus all" BLAST comparison, used in Biology to look for orthologs (similar genes) across multiple species [14]. In our case, we are looking for similarities among a set of digital artifacts. The result of this step is a BLAST report of the determined alignments between the artifacts.

The same FASTA sequence file, which was used to create the BLAST database, is now used to query against the database. Thus, BLAST will determine the similarities between each sequence and every sequence in the database; which is the same set of sequences. Thus, the "all versus all" comparison. The overall flow is depicted in Figure 3.

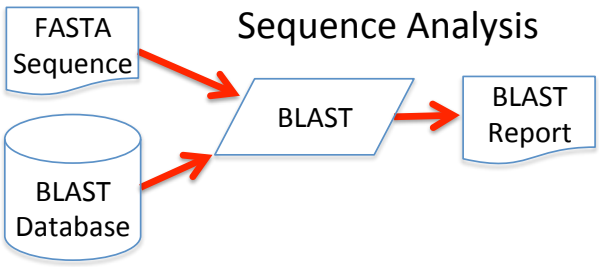

Fig. 3. Sequence Analysis Step. In this middle step the FASTA file and the BLAST database are examined and the sequence alignments between the original artifacts are reported.

As described in Section 2, BLAST has an "E-value" parameter. Specifying an E-value close to zero asks BLAST to return only highly similar alignments. We examined E-values ranging from 10^{-6} to 10^{-300} with "all versus all" BLAST comparisons. As the negative exponent decreased towards -300, the number of small alignments reduced dramatically, while the larger alignments remained stable . This indicates that BLAST considers longer alignments to be less likely to be random occurrences.

A BLAST report includes information concerning each alignment. This includes:

- Names of the sequences involved in the alignment
- Starting and ending positions of the alignment.
- Measures of the statistical significance including E-value

C. *Visualization*

The final step is the visualization of the results of the BLAST alignment. The visualization step consists of examining the BLAST report and creating a graph that represents those files where the sequences aligned with each other. The BLAST report is parsed and the alignment information is saved in a Simple Interaction Format (SIF) file. This graph format is used as input by the Cytoscape visualization tool [9] [13]. The overall flow of the visualization step is depicted in Figure 4.

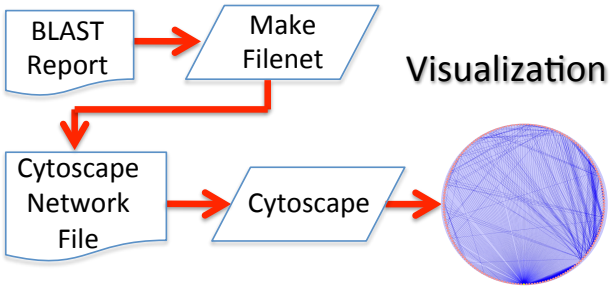

Fig. 4. Visualization of Similar Files. The output from the BLAST step is used to create input for Cytoscape. This provides various visualizations for the alignments of the DNA sequence

The visualization graph is constructed by considering every digital artifact to be a node in the graph, and edges representing the BLAST alignments between the artifacts. Such a graph will contain components that can be considered as clusters. The size of the clusters and the density of relationships among nodes in a cluster will vary depending on the E-value, which was used when performing sequence analysis.

This step optionally creates files, which show the alignments in "original format" (instead of DNA characters). This can be useful to examine what BLAST is determining aligns in its original form (rather than as DNA characters). Such inspection is particularly useful for text-based digital artifacts such as documents and source code.

4. Results

Several experiments were designed to evaluate the usefulness of this approach including document clustering, malware classification and plagiarism identification

A. *Document Clustering*

A set of 1,202 digital artifacts of fifteen different types were collected to test the document clustering process. The artifacts included text and binary files and both benign and malicious executables and benign and malicious JavaScript files. The types and counts of artifacts were as follows:

14 executable files	33 JavaScript	319 Java
229 Java ".class"	203 natural language	192 C
45 Scala	46 Perl	9 CGI (Perl)
15 Python	31 C#	24 HTML
18 PNG (image)	17 MP3 (audio)	7 ZIP

Among the 14 executable files, were 7 benign files and 7 malware files (4 Zeus Trojans and 3 Zeus Version Two Trojans as identified by MalwareDomainList.com). Also included were 33 JavaScript source files, of which 25 were malicious including 13 obfuscated and 12 de-obfuscated files. The malware JavaScript examples were obtained from http://redleg-redleg.blogspot.com/p/examples-of-malicious-javascript.html

The preprocessing, sequence analysis and visualization steps defined in the method section were followed. The E-value parameter of BLAST was set to 10^{-300}. This resulted in the creation of a graph, which contained 9,932 edges between the artifacts. The visualization of much of the graph, including its largest clusters is in Figure 5.

The graph contained clusters with very similar files, some of the clustering highlights included:

- A cluster with 23 HTML files (of the 24 in the data set)
- A cluster with 18 C# files (of the 31 in the data set)
- A cluster with 16 MP3 files (of the 17 in the data set)
- A cluster with 127 Java class files from a single project; another cluster with 16 Java class files from a different project; another cluster with 22 Java class files from three highly related projects (all implementing the same class assignment)
- A cluster with 10 Windows executable files including 4 Zeus Trojan executable files (but none of the Zeus version two executable files).
- A cluster with all 3 Zeus version two executable files was generated
- A cluster with 45 Java files from the same project, another with 19 Java files from a different project, one cluster with 14 Java files from one project.
- One cluster with a mixture of 50 Java and Scala source files (29 Java, 21 Scala); the Scala files had been converted directly from the project the Java files belonged to and were thus highly similar.
- One cluster contained 61 C files and header files from the same assignment

One cluster with 3 deobfuscated JavaScript files and another with 2 deobfuscated JavaScript files. (of the 12 deobfuscated JavaScript files)

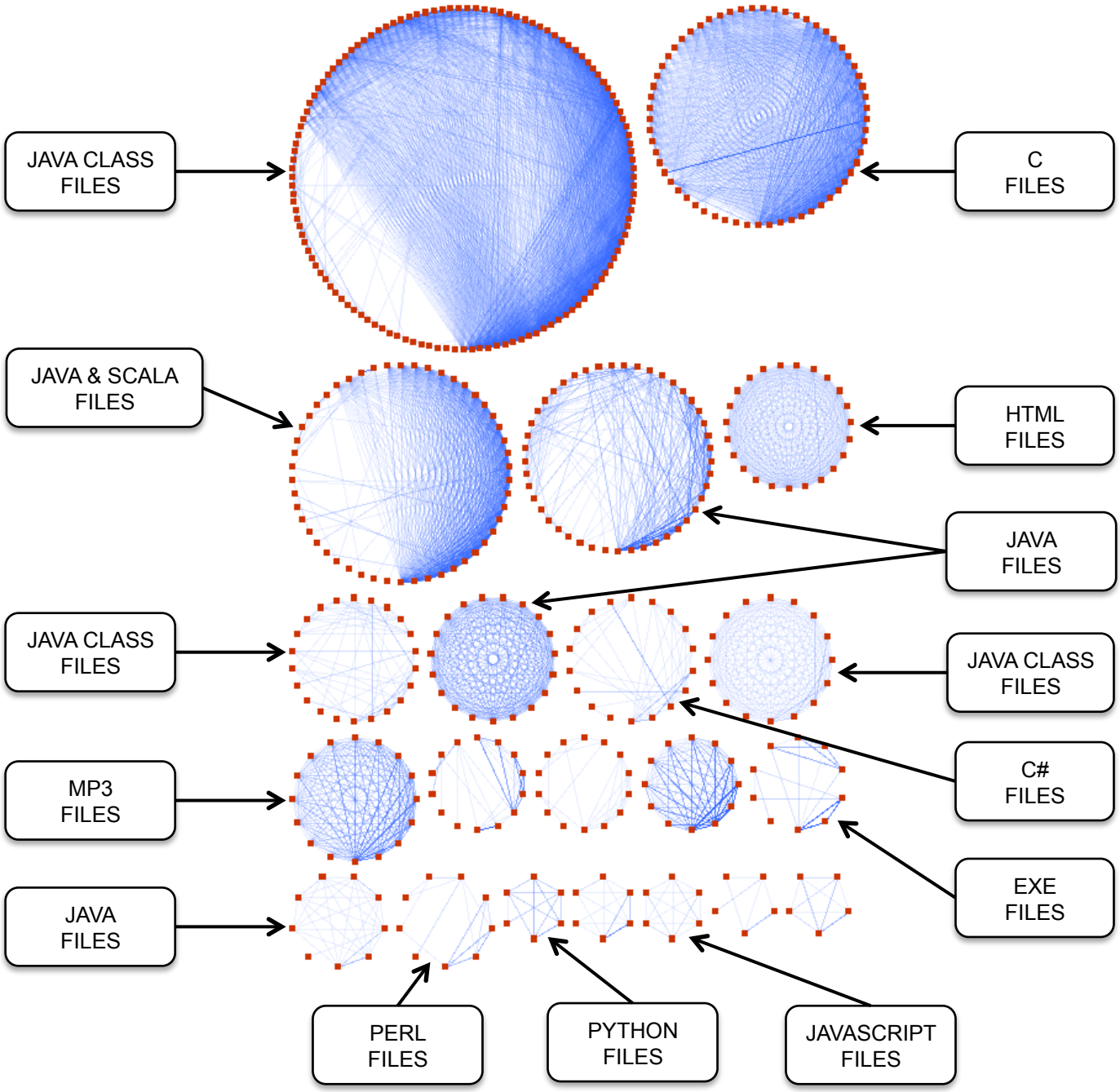

Fig. 5. Document Clustering at E-value 10^{-300}. Each almost exclusively consisted of files of the same type. For example, Java source files from different projects cluster together, while MP3 files and HTML cluster into separate groups.

The following was also observed:

- Of the 1202 files, there were 453 which had no alignments with any other files (approximately 38%). This is not unexpected due to the very high local similarity requirement enforced by using E-value 10^{-300}. Of the non-clustered files, there were 182 natural language files. If natural language files are excluded from consideration, there were 27% of the remaining files, which remained un-clustered.
- Obfuscated JavaScript had few alignments to other files.
- The following file types had almost no alignments to any other files: ZIP files, PNG files and natural language files.

In general, the clusters that formed had strong similarity of file type and frequently consisted of files from the same programming project. There were cases where a single Java programming project resulted in multiple clusters, but the clusters always exclusively consisted of Java source files. There were a significant number of files, which did not align and join with other files in a cluster. This appeared to be especially prevalent with natural language files.

The four Zeus Trojan executable files clustered with benign executable files, but Zeus Version Two Trojans executable files did not.

Additional testing was performed to see how the results would differ when examining the same 1202 files but using "looser" E-values of $10^{-250,}$ $10^{-200,}$ and 10^{-150}, which reduced the amount of local similarity required.

The following were some of the differences observed when clustering at E-value 10^{-200}:

- There were 30,680 edges in the network (compared to 9,932 previously).
- A cluster of all 14 executable files was created; including the Zeus and Zeus version two Trojan executable files. The Zeus version two executable files were no longer distinguished from other executable files).
- A large cluster of 259 files was generated consisting of 189 Java class files but also contains 35 related Java files and 35 Scala files. All files were related to implementing the same programming project.
- A cluster of 143 Java files from the same project was created.
- A cluster containing 91 C files from multiple programming assignments was created (of the 192 total C files).
- A cluster of 6 of the 8 benign JavaScript files was created (compared to two separate clusters which contained 5 benign JavaScript files previously).
- There were several clusters that were nearly identical to those of the 10^{-300} E-value case:
  - A cluster containing 23 C# source files (of the 31 total C# files)
  - A cluster containing 23 HTML files (of the 24 total HTML files)
  - A cluster with 19 Java files from the same project.

When further increasing the E-value to 10^{-150}, the following was observed:

- There were 177,011 edges in the network
- There were 235 files, which had no alignments to any other files; of these, 169 were natural language files. If we exclude natural language files, then 6.6% of the remaining files were un-clustered.
- One cluster contained 453 files which included Java class files but also some C source code, Java source code and Scala source code and C# source code.

The observations indicate that the specificity of the clustering based on file type starts to break down at a higher E-value setting. In summary, as the E-value increased from 10^{-300} to 10^{-150}, the number of alignments returned by BLAST increased substantially, and the level of similarity between files in clusters appeared to be reduced. For example, Java files from different projects, which were separated into different clusters at E-Value 10^{-300} were being clustered together at E-Value 10^{-150}. Similar results were seen for C source files and Java class files (byte code files).

B. Rapid Malware Classification

This experiment repurposed the BLAST database created by the document clustering test. It relies on the fact that there are Zeus and Zeus version two malware executable files within the database. The premise is that Zeus and Zeus version two malware executable files found "in the wild" should align closely with their counterparts in the database.

Recall that the 1,202 digital artifacts in the BLAST database were of the following types:

14 executable files	33 JavaScript	319 Java
229 Java ".class"	203 natural language	192 C
45 Scala	46 Perl	9 CGI (Perl)
15 Python	31 C#	24 HTML
18 PNG (image)	17 MP3 (audio)	7 ZIP

The experiment was to find another Trojan executable and see if it could be identified as such by examining its BLAST alignments with that BLAST database.

A malware executable was obtained on March 7, 2012, from a reference at MalwareDomain.com, which identified it as a Zeus Trojan. Its size and content differed from the four Zeus Trojan executable files in the BLAST database.

A biological representation of this executable was generated and BLAST was used to determine its alignments with the 1,202 files in the BLAST database.

At E-value level 10^{-300}, BLAST generated 31 alignments which were all to the four Zeus Trojan executable files in the database. This was viewed as a positive result.

At E-value level 10^{-200}, BLAST generated 2,364 alignments, of which the highest scoring 2355 (99.6%) were to the 4 Zeus executable files in the database. Of the remaining 9 lowest scoring alignments -- 7 were to benign executable files and 2 were to Zeus version two executable files. This was also viewed as a positive result.

C. Plagiarism Detection

A separate investigation was undertaken to investigate possible plagiarism in student program submissions in a C

programming class taught by one of the authors. The examination was done using E-value 10^{-300}. Several sets of programs files were examined. The topic of the assignment, and whether or not any "boilerplate" code (i.e. assignment bootstrapping code) was given to the class influenced the results significantly.
In cases where there was boilerplate code given to the class as part of the programming assignment, the student submitted programs all tended to cluster together. The alignments were observed primarily due to the boilerplate code that was common to all of the student files. Even in this circumstance, there were seen clusters with more inter-file alignments. There was not a clean separation of into components, but the number of alignments between files was a strong indicator of those files, which were suspiciously similar in regions other than the boilerplate.

When no boilerplate code was given, there tended to be fewer alignments. In this case, the alignments that were found showed suspiciously similar code between different student submissions.

5. Further Research and Conclusions

The specific uses of bioinformatics tools in this project gives only a taste of what the tools could be used for in the future. There are tools for classification of biologic objects, such as Restriction Enzymes, which may be of use in classifying computer artifacts – given a biological representation of those artifacts. One caveat is that there may be strong Biological assumptions made by those tools that would not be satisfied by biological representations of arbitrary digital artifacts

An intriguing possibility is to create a BLAST database containing sequences representing known malware including strains of malware executable files and JavaScript malware. This could be used as a possible rapid malware identification mechanism. A digital artifact whose biological representation aligns closely with any sequences in that database could be considered to be a likely malware executable. The "Rapid Malware Detection" test that was done indicates that this could be an effective identification mechanism.

It may be found that it is beneficial to have different BLAST databases for different types of artifacts. Just as there are different biologic databases for nucleotides and proteins, perhaps it may be useful to have databases that are specific to viruses as opposed to malware JavaScript source files.

The specific outputs produced by this project can be improved in various ways:

- The Cytoscape visualization can be improved to change the size of a node or the width of an edge based on the size of the alignment. Multiple edges between two artifacts might be able to be condensed into a single edge using a color scheme to indicate the number of alignments
- A visual map of the alignments of two digital artifacts could be produced. This would be analogous to a "homolog map" in Biological domain which shows the positions of related genes in the genomes of two species.
- A Cytoscape plugin could be created to allow an viewing the original format of any alignment and remove those not of concern or highlight those of concern.

This paper presented a novel method for clustering digital artifacts, identifying a digital artifact as similar to known malware, and detecting plagiarism by using a synthetic DNA representation of the digital artifacts and using the bioinformatics BLAST tool. It demonstrates that the classification power of bioinformatics tools that are used in biology problems can also be used in other domains.

The success of using BLAST to examine nucleotide representations of computer artifacts can be partly attributed to the fact that BLAST does not make strong assumptions about the chemical differences between nucleotides when processing a nucleotide database. This is not true when using BLAST with protein databases. It may be possible to use BLAST with synthetic protein definitions by providing specialized scoring matrices to BLAST which do not make biologic assumptions that would not hold.

6. References

[1] Distler, Dennis, and Charles Hornat. "Malware Analysis: An Introduction." *Sans Reading Room*. Sans, 14 Dec. 2007.

[2] Kendall, Kris. "Practical Malware Analysis." Blackhat, 2007. Web. 11 Jan. 2012. <http://www.blackhat.com/presentations/bh-dc-07/Kendall_McMillan/Paper/bh-dc-07-Kendall_McMillan-WP.pdf>.

[3] Watson JD, Crick FH, "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid", *Nature, 1953, Vol* **171**

[4] Elson D, Chargaff E, "On the deoxyribonucleic acid content of sea urchin gametes".Experienti, 1952, Vol 8

[5] Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. Basic local alignment search tool, 1990, J. Mol. Biol, Vol 215

[6] Pagni M, Jongeneel CV, Making sense of score statistics for sequence alignments, Briefings in Bioinformatics, 2001, Vol 2

[7] Krauthammer M, Rzhetsky A, Morozov P, Friedman C, Using BLAST for identifying gene and protein names in journal articles,

[8] Altschul, S. et al, Gapped BLAST and PSI-BLAST; Nucleic Acids Research, 1997, Vol. 25, No.17,

[9] Shannon, P. et al. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks; Genome Research, 2003, Vol. 13, Pgs. 2498-2504

[10] McGinnis S and Madden T, BLAST: at the core of a powerful and diverse set of sequence analysis tools, Nucleic Acids Research, 2004,

[11] Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison, 1988, Proc. Natl Acad. Sci. USA,

[12] Mount, D., Comparison of the PAM and BLOSUM Amino Acid Substitution Matrices, 2008, Cold Spring Harbor Protocols, doi:10.1101/pdb.ip59

[13] Cline, M. et al, Integration of Biological Networks and Gene Expression Data using Cytoscape, Nature Protocols, 2007, Vol 2, Pgs 2366-2382

[14] Moreno-Hagelsieb G and Latimer K, Choosing BLAST options for better detection of orthologs as reciprocal best hits, Bioinformatics, 2008, Vol 24

[15] HTTP://BLAST.NCBI.NLM.NIH.GOV/

[16] ftp://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST