Cal Poly Library of Pyroprints:

Quality Control Analysis and Web Development

A Senior Project

presented to

the Faculty of the Computer Science department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

by

Chase Ricketts

March, 2014

1 Introduction

1.1 Background

Microbial source tracking has many different applications within the field of biology. Cal Poly biologists have implemented an effective tracking method by accumulating a database of known bacteria and matching unknown bacteria against them. This is possible when bacteria have particular digital fingerprints. For example, different strains of E. coli vary significantly in two particular intergenic regions in their DNA: regions 23-5 and 16-23. The biologists use the Pyroprinting process to sequence these regions of DNA in order to cheaply and quickly develop Pyroprints which represent bacteria's digital fingerprints.

The Cal Poly Library of Pyroprints (CPLOP) is a web application designed for the Cal Poly biology department that enables its users to manage a database of E. coli Pyroprints and metadata. CPLOP provides basic operations such as inserting, deleting, and searching data and more complex analysis operations such as matching, clustering, and displaying graphic representations of data. This enables the biologists to utilize CPLOP as an E. coli strain tracker[1] and come to conclusions about the geographic spread of E. coli. A significant portion of CPLOP was developed by master student Jan Soliman and is documented in his thesis [1]. The clustering algorithms were developed by another master student Aldrin Montana and is documented in his thesis [2].

Figure 1: E. coli is everywhere, including the great state of California

[1] Often reworded to "poop tracker."

1.2 Scope

The contents of this document primarily relate to the portions of CPLOP that I worked on directly, but section 5 contains some observations that were unrelated to my work. My most significant contributions related to developing quality control functionality, integrating Aldrin's clustering algorithm, and migrating the server to Git.

2 Quality Control: Pyroprint Integrity

Ensuring the integrity of Pyroprint data in CPLOP is necessary for Cal Poly biologists to make conclusions that are reliable. I have helped this happen by implementing visualizations of data and developing quality control checks that automatically validate data whenever new Pyroprints are uploaded to CPLOP.

2.1 Visualizing the Data

2.1.1 Pearson Correlation Shortfall

CPLOP compares Pyroprints against each other using a Pearson Correlation[2] in order to determine if they are significantly similar. The Pyroprints are similar if their Pearson Correlation value exceeds or equals 0.995, possibly similar if the value exceeds or equals 0.990, and dissimilar otherwise. This is illustrated in Table 1.

Pearson Correlation Value	Significantly Similar Pyroprints?
[0.995, 1.000]	Yes
[0.990, 0.995)	Possibly
[0.000, 0.990)	No

Table 1: Higher Pearson Correlation values imply more similar Pyroprints

The Pearson Correlation function calculates the similarity between two Pyroprints by applying a formula on the first `N` Pyroprint dispensation peak heights where `N = 93` for region 23-5 and `N = 95` for region 16-23. However, these values

[2] The Pearson Correlation function is used to counteract typical experimental variance that occurs during the Pyrosequencing process, such as the change in ambient light. This is explained in detail in Jan's thesis [1].

alone cannot explain *where* two Pyroprints differ. I have implemented an incremental depiction of the Pearson Correlation function so that the user can get a more detailed understanding of a Pyroprint comparison.

2.1.2 Cumulative Pearson Correlation Graph

This graph calculates Pearson Correlation values with varying dispensation lengths within a single Pyroprint comparison instead of just one dispensation length. Specifically, the value is calculated for dispensation lengths `2, 3, 4...N` where `N` is 93 or 95 depending on the region. This enables the user to see the change in similarity between two Pyroprints as the Pearson Correlation value is calculated over an increasing number of dispensation lengths.

Example: A Typical Match Figure 2 shows the cumulative Pearson Correlation of Pyroprint 7615 and Pyroprint 7505. The value at dispensation length 95 is about 0.9986, which means that the two Pyroprints are significantly similar. The graph has more variation at smaller dispensation lengths because a smaller number of peak heights are being analyzed and therefore each has a higher influence on the overall Pearson Correlation value.

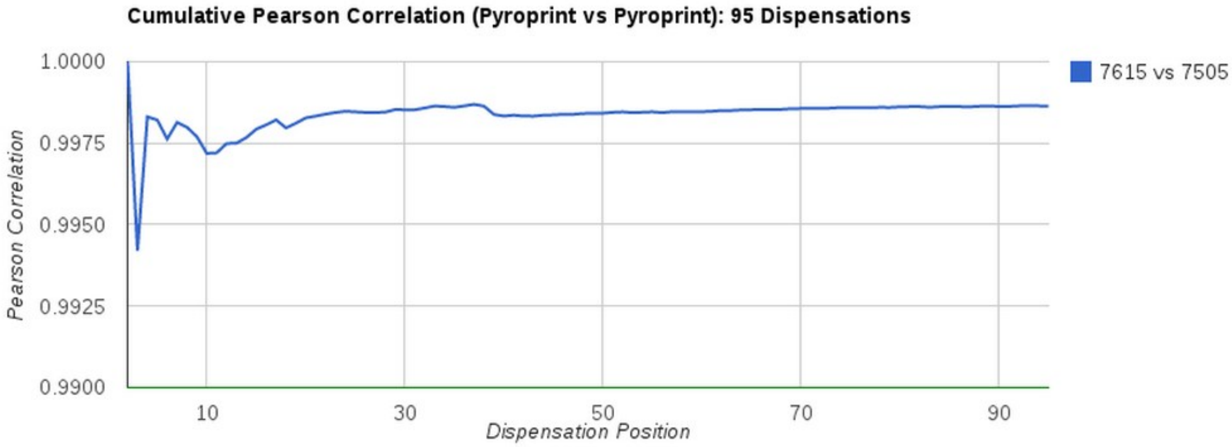

Figure 2: A typical cumulative Pearson Correlation graph with matching Pyroprints

Example: A Typical Mismatch Figure 3 shows the cumulative Pearson Correlation of Pyroprint 12634 and Pyroprint 12403. The result at dispensation length 95 is about 0.9856, which means that the two Pyroprints are significantly dissimilar. The graph reveals that the Pyroprints vary dramatically around dispensation 73.

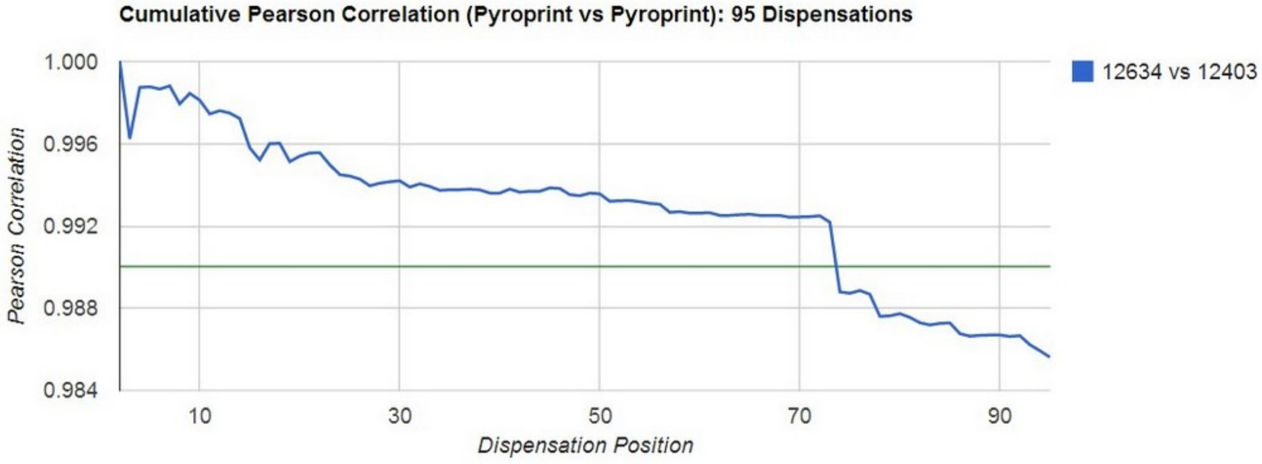

Figure 3: A typical cumulative Pearson Correlation graph with mismatching Pyroprints

Example: A Potential False Positive These graphs can help detect Pyroprint comparisons that fall within the "possibly significantly similar" range but should actually be dissimilar. For example, we compare Isolate ES-725 against Isolate Cw-776 in Figure 4. Although their Pearson Correlation value is just above 0.990 and therefore possibly significantly similar, the dramatic variance around dispensation length 10 suggests they may be erroneous or significantly different. Biologists must analyze these Pyroprints to draw their own conclusions of the validity of this match.

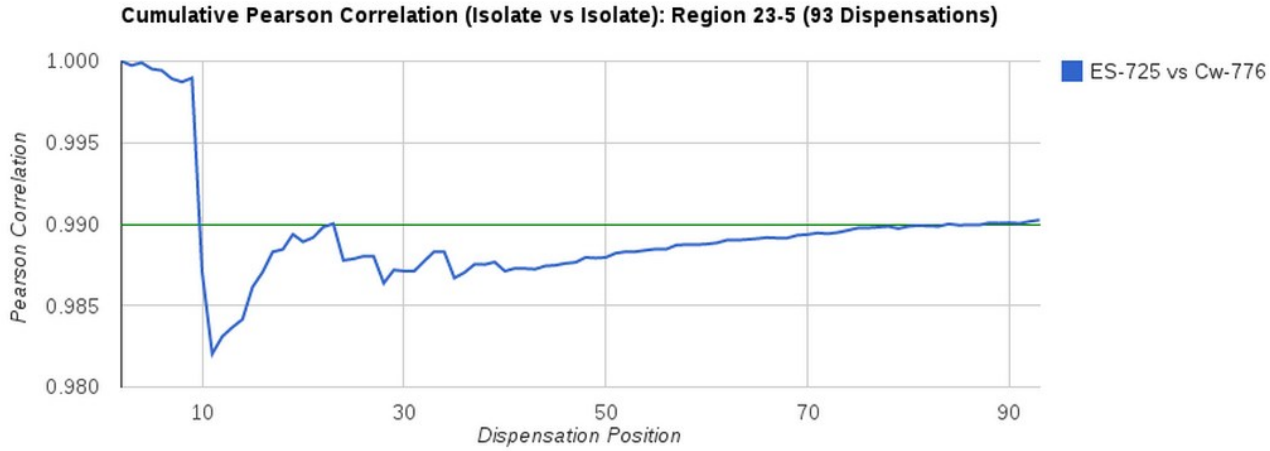

Figure 4: An atypical cumulative Pearson Correlation graph

2.2 Google Charts

Google Charts is a collection of JavaScript libraries that was released by Google Developers under a Creative Commons Attribution 3.0 License [3]. These libraries make it easy to generate many different types of customizable charts and Google provides full documentation online. I utilized these libraries to dynamically generate cumulative Pearson Correlation graphs for Pyroprint comparisons. The website and full documentation for all charts can be found at https://developers.google.com/chart/.

2.3 Detecting Erroneous Histograms

2.3.1 Pyroprint Histograms

The Pyroprinting process generates a lot of data. CPLOP only stores a small subset of this data, including the amount of light released during each dispensation cycle. This light value helps determine the type of nucleotide in the DNA sequence: `A`, `T`, `G` or `C`. For example, if the light value is high when the dispensation cycle releases `A`s, then the next nucleotide in the DNA sequence is likely to be a `T` or a string of `T`s because there is more chemical bonding occurring.

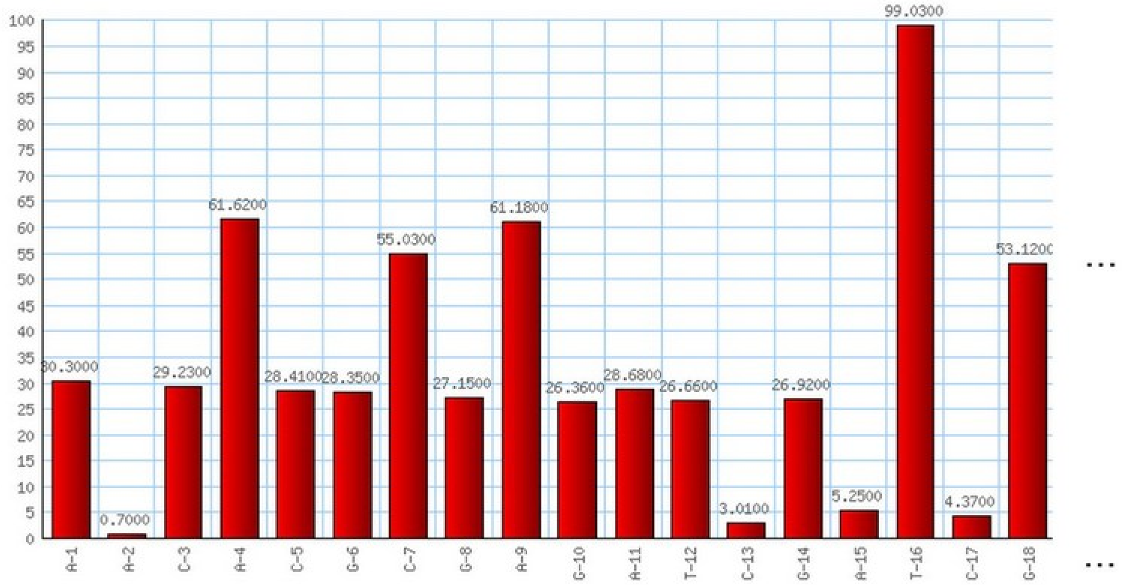

Figure 5: A typical Pyroprint histogram with only 18 dispensations shown

These histograms will vary from Pyroprint to Pyroprint due to E. coli DNA sequence differences, changes in ambient light, and other experimental variance. Pyroprints with similar histograms may be significantly similar whereas Pyroprints with completely different histograms will be significantly dissimilar.

2.3.2 Detecting Erroneous Pyroprint Histograms

The Pyroprint histograms for E. coli regions 23-5 and 16-23 are typically[3] consistent for the first 8 dispensations for each region. The absolute values may change in these first 8 dispensations because of changes in ambient light, but the ratios from dispensation to dispensation remain consistent. For example, the values {1, 2, 3} and {2, 4, 6} have perfectly consistent ratios.

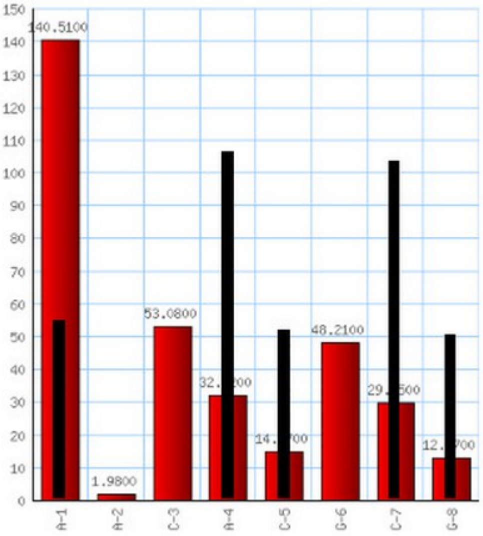

Figure 6: A histogram that does not follow the pattern (shown in black)

I used an Excel spreadsheet to calculate the standard deviation, standard error, and medians for the ratios of all pairs over the first 8 dispensations for both regions using the entire database (5 to 6 thousand Pyroprints per region). I determined that the standard error for dispensation ratios is lowest when all 8 dispensations are divided by the 3rd dispensation (for region 23-5) and 6th dispensation (for region 16-23). Now whenever a new Pyroprint is uploaded to CPLOP its histogram is analyzed and checked to be consistent with the expected pattern.

This kind of quality control check catches histograms that are erroneous and should not be analyzed as good data. Something may have gone wrong in the Pyroprinting process or it may have been uploaded incorrectly. However, this algorithm also catches valid histograms that are atypical cases and do not adhere to this typical pattern. The flag brings awareness to the Pyroprint, but the biologists must determine the validity of the flag.

The histogram in Figure 6 is an example of a Pyroprint of region 23-5 that is flagged by this quality control check. The expected pattern is shown in black. CPLOP will flag this Pyroprint with the warning message "outstanding histogram

[3] Some strains of E. coli are not consistent with this pattern, but they happen infrequently and it's worth getting a few false positives.

values at position[stddev]: 1[15.28] 4[-11.02] 5[-11.38] 7[-10.47] 8[-7.75]" to signify the unexpected low and high values at positions 1, 4, 5, 7, and 8. The biologists may then analyze the original Pyroprint data to determine its validity.

2.4 Detecting Dissimilar Pyroprint Replicates

The Pyroprinting process is repeatable; the same E. coli sample Pyroprinted twice should lead to two significantly similar Pyroprints. However, due to experimental error the Pyroprinting process will occasionally produce erroneous results. A human can sometimes detect this experimental error by analyzing the Pyroprint data but it requires too much time and effort to manually check every Pyroprint. As of this writing, master student Alex Bozarth is working on an automated way to detect these experimental errors.

In the meantime, I have implemented a simple quality control check that will calculate the Pearson Correlation for a newly uploaded Pyroprint to its existing replicates in CPLOP. In ideal circumstances, all Pearson Correlation values will be above 0.990. However, if a value below 0.990 is calculated then the Pyroprint is significantly dissimilar to its replicate. This suggests experimental error and so CPLOP flags the Pyroprint with a warning. This brings the Pyroprint to the attention of the biologists to manually check its validity.

3 Integration of Hierarchical Clustering

3.1 Background

Master student Aldrin Montana developed the Suite of Pyroprint Analysis Methods (SPAM) which includes algorithms to group Isolates using either hierarchical clustering or ontological clustering [2]. Because SPAM is an application external to CPLOP, Cal Poly biologists had to contact Aldrin through email whenever they wanted to cluster Isolates. This was inefficient due to time overhead and miscommunications.

As part of my development of CPLOP, I have integrated the hierarchical clustering portion of SPAM into CPLOP so that users can run the clustering algorithm on demand.

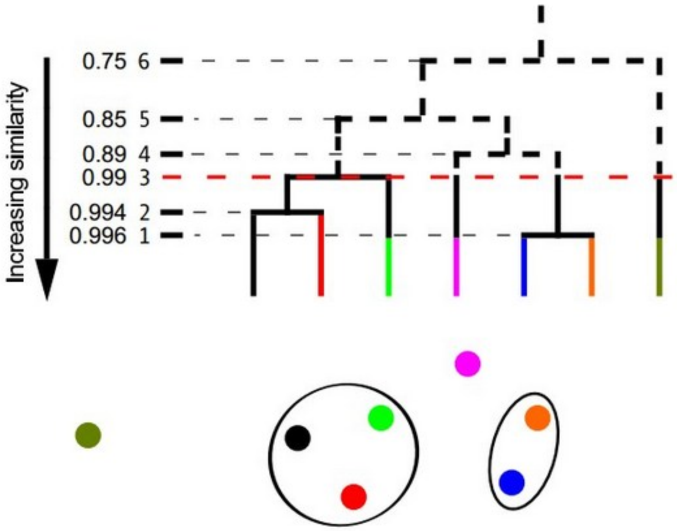

Figure 7: Aldrin's illustration of hierarchical clustering with 0.99 cutoff [2]

3.2 Clustering Isolates

My work enables a user to select a group of Isolate Datasets within CPLOP and run the clustering algorithm on them. CPLOP does this by executing a Java main class within the SPAM jar file using PHP's `exec` function. The arguments are alpha and beta thresholds (0.995 and 0.990) and a list of Isolate identifiers. SPAM utilizes the local database in order to access Pyroprint data. The output is parsed into a PHP object and I have provided additional functions to convert the object into HTML and CSV formats.

This direct integration of the hierarchical clustering algorithm has enabled CPLOP's users to run clustering analysis without having to communicate with Aldrin.

4 Migration to Git

4.1 Codebase Location

As of this writing, the CPLOP codebase is located on BitBucket.com which is a similar website to GitHub.com. Jan Soliman was the student to start using Git as the primary version control software and he now owns the CPLOP repository under the account name `jsoliman`.

4.2 Git on the Production Server

When I started working on CPLOP, the codebase was being versioned with Git but the production server had to be updated by manually copying files. This was inefficient and prone to human error. I have since installed Git on the server and created a local `release` branch that contains specific configurations for the production server. The server can now be updated by performing a `git pull` followed by a `git rebase`. This simplifies the update process and reduces the number of errors that can occur.

5 Future Work

Through my work with CPLOP I have understood the codebase well enough to identify its shortfalls. In summary, it needs a complete overhaul in order to be in a secure and maintainable state. As CPLOP grows and it becomes used by more users, security vulnerabilities will become more of a concern. As more software developers add features to CPLOP as it is, it will become increasingly difficult to maintain. This issues can be avoided if the codebase is restructured and partially rewritten according to PHP conventions.

5.1 SQL Injection Prevention

SQL queries riddle the codebase because CPLOP is so data centric. Most of these queries are prone to SQL injection due to the lack of input string sanitation. These queries should be rewritten to sanitize their inputs. This would also be a good time to centralize these queries in one location rather than having them spread throughout the codebase and integrated with front-end logic.

5.2 Password Hashing

The database currently stores user passwords as plain text. This unnecessarily exposes passwords to CPLOP developers and the passwords are vulnerable to theft if the database is hacked. Password hashing should be used to protect user passwords from both developers and hackers.

5.3 Add Error Checking

Most of the CPLOP codebase lacks error checking. When something fails it can be very difficult to debug because the “fail fast” convention is not followed. Error checking needs to be added to most of the codebase to provide helpful error messages.

5.4 Localize Configuration Settings

The configuration settings for CPLOP is split up among multiple PHP files. This makes it difficult to find and alter these settings as necessary to get a local application running or to update the production server configuration. These configuration settings should be moved to a single configuration file.

5.5 Restructure Codebase

PHP files are meant to be built in a hierarchical fashion. This creates more consistency among the entire website and simplifies maintenance. However, currently the codebase does not take much advantage of `includes` and not many PHP conventions are followed. Most of the PHP files are located in a single directory and there is a lot of code duplication. This makes it difficult to update the entire website when the same code is repeated in multiple places.

6 Final Thoughts

Working on CPLOP has been a great experience for me. The research behind CPLOP has made my work valuable and I am glad to have been a part of its creation.

I thank my professor and advisor Alexander Dekhtyar for getting me involved in this project and being incredibly helpful along the way. I also thank Michael Black, Jennifer VanderKelen, and Christopher Kitts for being helpful, considerate, and informative when discussing and specifying requirements for CPLOP. I’ve enjoyed my time with all of you. It’s been fun.

References

[1] Soliman, Jan. “CPLOP: The Cal Poly Library of Pyroprints.” MS thesis. California Polytechnic State University, 2013. Web. 16 Mar. 2014.

[2] Montana, Aldrin. “Algorithms for Library-Based Microbial Source Tracking.” MS thesis. California Polytechnic State University, 2013. Web. 16 Mar. 2014.

[3] Google Developers. *Google Inc.*, 3 Apr. 2012. Web. 16 Mar. 2014