Library-Based Microbial Source Tracking via Strain Identification

Jeffrey D. McGovern
jmcgover@calpoly.edu

Eric Johnson
ejohns32@calpoly.edu

Alex Dekhtyar
dekhtyar@calpoly.edu

Computer Science Department
California Polytechnic State University
San Luis Obispo, CA

Michael Black
mblack@calpoly.edu

Christopher Kitts
ckitts@calpoly.edu

Jennifer VanderKelen
jvanderk@calpoly.edu

Biological Sciences Department
California Polytechnic State University
San Luis Obispo, CA

ABSTRACT

Microbial Source Tracking (MST) aims to classify the source host-species of biological matter, typically fecal matter, using strains of fecal indicator bacteria, often *E. coli*. This paper continues addressing the MST problem using analysis of a library of bacterial fingerprints started in [9]. The Cal Poly Library of Pyroprints (CPLOP) is a collection of fingerprints of over 6,000 *E. coli* isolates collected from the fecal matter of a variety of host-species. In prior work [9] we studied the accuracy of the MST process based on k-Nearest Neighbors discovery in CPLOP. This process, while sufficiently accurate, does not scale well with the size of the database. In this paper, we study the accuracy of a clustering-based MST approach which scales significantly better: the bacterial isolate information stored in CPLOP is clustered using an efficient density-based clustering technique. We present our analysis of the accuracy and efficiency of the clustering-based MST methodology for CPLOP.

CCS Concepts

•**Computing methodologies** → **Cluster analysis;**
•**Applied computing** → **Bioinformatics;**

Keywords

microbial source tracking, pyroprinting, clustering

1. INTRODUCTION

In order to combat the contamination of publicly accessible water supplies, health and environmental protection agencies have been interested in tracking the source host-species of fecal contamination [2, 18, 20]. Contact with such contamination can be harmful to humans, pets, and the local ecosystem, and restricting public access until contamination levels reach an acceptable level is often the only course of action. Thus, it suits natural resource managers to identify the source of the contamination and take necessary steps to prevent further contamination.

Microbial Source Tracking (MST) is the field of research that aims to discover the source host-species of microbial lifeforms. In the case of fecal contamination, researchers use what are called fecal indicator bacteria (FIB) under the tentative hypothesis that this FIB stay relatively unique to the host-species over an extended period of time.

In 2009, the California Polytechnic State University San Luis Obispo (Cal Poly) Biology and Computer Science Departments built The Cal Poly Library of Pyroprints (CPLOP) [21][1], a database of bacterial isolate fingerprints (called *pyroprints*). CPLOP is designed for the so-called library-based MST: MST methods that work by comparing the information about an environmental bacterial sample of unknown origin against the digital representations of the bacteria of known origins stored in the database (library). In library-based MST work such digital representations may employ either genotypic or phenotypic characterizations. CPLOP uses a genotypic method called pyrosequencing on a mixture of DNA to create a genetic fingerprint called a *pyroprint* [7]. Previous work on CPLOP focused on exploring and tracking the similarity of *E. coli* pyroprints [13, 14, 15, 16], but it was not until [9] when we started addressing the microbial source tracking problem.

Our first approach to library-based MST with CPLOP used a modification of the k-Nearest Neighbors (k-NN) classification algorithm as the source tracking method. A fingerprint of a bacterial isolate of unknown origin was compared to the fingerprints of bacterial isolates stored in CPLOP, the k closest bacterial isolates were determined, and the host-species was identified as the original species of the plurality of the selected nearest [9]. We showed that the accuracy of identifying the correct host-species of a given bacterial

[1] `http://cplop.org`

BCB '16, October 02-05, 2016, Seattle, WA, USA

DOI: http://dx.doi.org/10.1145/2975167.2975205

isolate was reasonably high despite the CPLOP collection having grown organically.

The k-NN-based method we considered in [9] has a number of drawbacks. First, in its initial form it does not scale well[2] with the size of CPLOP. More importantly, though, the k-NN method ignores one of the core interests of biologists: grouping the bacterial isolates stored in the database into *strains*.

A *strain* of a species of microbe is a subtype of that species where the microbes in that strain are closely related to each other in some meaningful way. Generally, how one defines a strain differs between research groups, as each definition of a strain depends on the characterization of the microbe in question and the methods used to derive such characterizations. Typically, the goal is to discover which microbes of a bacterial isolate came from the same parent microbe [8]. A strain, then, can be thought of as consisting of individuals descending from the same individual to generate a "group" or "family."

It is this notion that drives us to define strains in CPLOP. *E. coli* in the gut of a given animal thrive and proliferate, creating more similar *E. coli*. Some end up in the fecal matter of that host-animal. If we can delineate which strains of FIB tend to reside in the host-animals of a given host-species, then we can use this information to track which host-species is causing the fecal contamination.

In this paper we use the scalable, density-based clustering method based off of DBSCAN, described in depth in [6] to build strains in CPLOP data. We then study the properties of the constructed strains as they pertain to the MST process. In particular, the contributions of this paper are the following:

- A library strain-based MST procedure for bacterial isolates.
- An efficient density-based clustering algorithm based off of DBSCAN that is scalable and meaningfully encodes the comparison-metric used in CPLOP.
- A set of validation measures for strain-based Microbial Source Tracking.
- An evaluation of our strain-based MST procedure based on the defined set of measures.

Our strain-based MST procedure places over 60% of the CPLOP isolates in strains and allows us to identify host-species of those isolates with over 80% accuracy.

The rest of this paper is organized as follows: §2 explains CPLOP and the relevant terms and functions, §3 defines our strain-based MST procedure, §4 summarizes the density-based clustering algorithm used, §5 presents our validation process, and §6 discusses the results of evaluation.

2. The Cal Poly Library of Pyroprints

CPLOP stores information about multiple collected bacterial isolates of *E. coli*. The information stored in CPLOP is called *pyroprints* [1]: the peak heights of pyrosequences of specially constructed DNA products extracted from the *E. coli* DNA. Each isolate in CPLOP has any number of pyroprints[3], but when we refer to *E. coli*, we are typically referring to an isolate. In what follows we provide a brief description of the pyroprinting process and the CPLOP data.

2.1 Pyroprints

Pyrosequencing is a DNA sequencing technique appropriate for sequencing short DNA fragments (up to around 150-200 base pairs) [17]. Pyrosequencing is less powerful than some of the other modern DNA sequencing techniques, however, it is also significantly less expensive with a single pyrosequencing run costing on the order of tens (rather than thousands) of dollars.

A pyrosequence of a DNA fragment is represented as a vector of real values — one value per dispensed nucleotide, indicating the intensity of light emission that occurred during the sequencing reaction (light is emitted in response to a specific nucleotide reagent used in the sequencing process; the amount of emitted light is proportional to the amount of the nucleotide material used in the reaction).

Pyroprinting is a fingerprinting technique for bacteria that uses pyrosequencing on a "mixed" DNA product consisting of amplified copies of *multiple DNA fragments*. The resulting pyrosequence may not be used to reproduce the underlying DNA sequence (as there is often more than one single underlying DNA sequence), but instead becomes a "fingerprint" of the mixed product which we call a *pyroprint* [1]. CPLOP uses pyroprints of the two *E. coli ITS* regions whose nature is briefly described below.

2.2 Internal Transcribed Spacers

The internal transcribed spacers (*ITS*) are regions of DNA that reside between genes in a microbe. *ITS* regions are frequently used for strain delineation, due to their highly unconserved nature. Depicted in Figure 1 is an abstracted segment of *E. coli* DNA used in CPLOP. *The illustrated pattern repeats around the ring of E. coli DNA seven times.* The two *ITS* regions between the *16S* and *23S*, and between the *23S* and *5S* genes of the ribosomal DNA are respectively called *ITS-1* and *ITS-2* and contain mostly non-coding DNA.

These segments of DNA offer keen insight into strains of *E. coli*. Since they do not code for functional products, random variations occur in *ITS* regions that do not affect the survivability or reproducibility of the microbe. The *ITS-1* and *ITS-2* regions are inherited in any offspring, allowing one to use them to differentiate strains [22].

We pyroprint each *ITS* region separately. The DNA product used for pyroprinting *is a PCR-amplified mix of the DNA from the seven loci of the ITS region in the E. coli DNA*. Each locus has a DNA sequence that may be different from the sequences in the other six loci, but all seven loci are amplified jointly by selecting appropriate primers [1].

Studies performed in [19] determined the optimal number[4] of dispensations to consider for each *ITS*. Too few dispensations may not encode enough information, while too many may degrade the quality of the data [11]. As a result, the number of dimensions a pyroprint vector has differs depending on which *ITS* the pyroprint represents. For *ITS-1*, we use 95 dispensations of nucleotides and for *ITS-2*, we use 93.

[2]Although the methodology discussed in this paper can be used to speed up the pure k-NN procedure as well.

[3]at least one in each *ITS* region

[4]The actual sequence of dispensed nucleotides was determined via simulation using tools from [11].

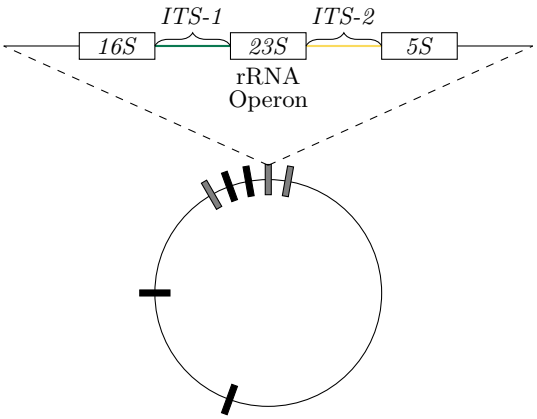


Figure 1: A diagram of a simplified segment of *E. coli* DNA, outlining the *ITS-1* and *ITS-2 ITS* regions, which repeat 7 times around the *E. coli* genome.

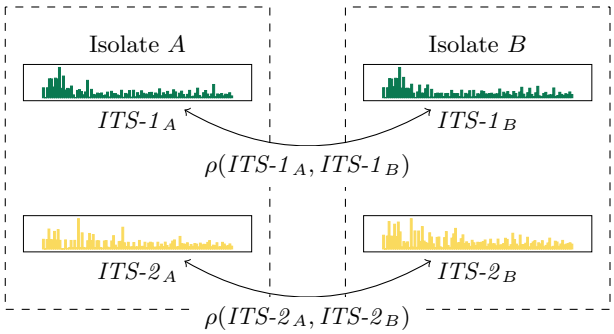


Figure 2: Comparing isolates involves comparing the pyroprints of each isolate using ρ, the Pearson correlation coefficient, with the stipulation that one can only compare pyroprints from the same *ITS* in their respective isolate. The green bar plots represent a pyroprint of *ITS-1*, while gold bar plots represent a pyroprint of *ITS-2*.

2.3 Isolates

The data stored in CPLOP is obtained as follows. Biologists collect fecal samples from a host subject of a known species. They extract *E. coli* and grow cultures of individual bacterial cells from the bacterial material contained in each sample. An individual bacterial culture grown from a fecal sample is called a bacterial *isolate*. Each isolate undergoes PCR procedures that amplify the DNA in the two *ITS* regions, after which the pyrosequencing of each region produces pyroprints that are stored in CPLOP.

Comparing isolates in CPLOP involves using the Pearson correlation coefficient[5] to calculate a similarity value between the pyroprints of two isolates. Since we use *ITS* regions to differentiate between strains of *E. coli*, it is only meaningful to compare pyroprints that come from the same *ITS* region in their respective isolate. That is, given isolate A and isolate B, one can only compare the pyroprint of $ITS\text{-}1_A$ to the pyroprint of $ITS\text{-}1_B$ and the pyroprint of $ITS\text{-}2_A$ to the pyroprint of $ITS\text{-}2_B$[6]. Figure 2 illustrates this concept.

3. STRAIN IDENTIFICATION FOR MICROBIAL SOURCE TRACKING

In [9] we looked at a k-Nearest Neighbors (k-NN)-based microbial source tracking method that essentially boiled down to the following process:

1. Given the pyroprints of an isolate of unknown origin, find k isolates from CPLOP most similar to it.
2. Return the plurality species of origin for the k-nearest neighbors of the query isolate as the method's guess.

This method showed reasonably high accuracy on the CPLOP data. However, it has two main disadvantages that require improvement. First, the performance of a naive implementation of k-NN over CPLOP is not scalable with the increase in our pyroprint collection. Second, one of the key initial goals of CPLOP is to help biologists study *E. coli* strains that are characteristic of different host-species and those that are *transient* (i.e., the strains that are found in multiple host-species). The k-NN-based approach of [9] does not contribute to our understanding of bacterial strains present in CPLOP.

In this paper, we consider an MST method that is based on strain discovery in CPLOP. For the purposes of this work, we define an *E. coli* strain as a *group of E. coli isolates that share exactly the same ITS-1 and ITS-2 DNA sequences.*

From the computer science point of view, a bacterial strain is essentially a cluster of *E. coli* isolate representations stored in CPLOP. Our MST method, thus, works as follows:

1. **Strain Identification.** Identify bacterial strains in CPLOP by clustering all CPLOP isolates.
2. **MST.** Given an isolate of unknown origin, find the cluster it belongs to. Return the host-species of the plurality of isolates in the cluster.

For our clustering algorithm, we use a density-based clustering algorithm developed by Johnson [6]. This algorithm extends DBSCAN for the case of two similarity metrics between data points (our isolates are compared based on two ITS regions) and implements an efficient spatial data structure to manage the storage and retrieval of the data points.

In this paper we look at the results of clustering CPLOP data using this algorithm from the perspective of *cluster purity*. We call a cluster (strain) *100% pure* if all isolates that belong to it come from the same host-species.

Of interest to us is the following information:

1. The number of 100% pure clusters and the percentage of bacterial isolates from CPLOP clustered into pure clusters.
2. The structure of impure clusters: specifically, whether a dominant host-species can be clearly identified in each cluster.
3. Coverage: the total number of CPLOP isolates found to belong to a strain.
4. MST Accuracy: the percentage of isolates for which the strain-based MST procedure produces the correct response.

[5]See (1) in §4.3.

[6]Algorithms to resolve these two regions for MST are discussed in [9].

In the next section we provide a brief discussion of the density-based clustering algorithm of Johnson [6] and its use to build CPLOP isolate clusters.

4. DENSITY-BASED CLUSTERING

Density-based clustering algorithms build clusters based on two parameters: the minimum number of neighbors, `MinPts`, a point must have to be a core point of a cluster, and ε, the radius that those neighbors must be within. These algorithms define clusters with respect to core points and border points — points within ε of a core point — labeling everything else (singletons) as noise. In the following section, we describe density-based clustering and summarize our modifications and optimizations of it for CPLOP. In this section we provide a brief description of the density-based clustering algorithm we use to cluster CPLOP isolates [6].

4.1 DBSCAN

DBSCAN[4] provides the framework for our clustering algorithm. In a nutshell, DBSCAN uses a distance metric, a minimum neighbors value `MinPts`, and an ε range to categorize data points as one of three types: (a) core point (b) border point, or (c) noise. A *core point* is a point that has at least `MinPts` data points within ε of it. A *border point* is a point that is within ε of a core point, but that does not have `MinPts` points within ε of it. Every other point is *noise*. A *cluster* is a group of neighboring core points with their associated border points. According to this definition of a cluster, all clusters must have at least `MinPts` points in them.

For this work, we chose to use DBSCAN as the clustering technique for grouping isolates because dense groupings of similar isolates fits our intuition of bacterial isolate strains. Closely related "families" of isolates will appear in the same cluster and we want these clusters to have sufficient purity to aid us in MST. While the work in [9] using k-NN was reasonably accurate for MST, it failed to provide the kind of insight into bacterial strains that DBSCAN's density-based groupings can offer.

4.2 Spatial Indexes

Density-based clustering techniques require a distance metric, often times the euclidean distance, between data points in order to cluster. Performing fast range queries greatly improves the speed of clustering. If the range query can finish in $O(\log n)$ time, then DBSCAN can run in $O(n \log n)$ time. Organizing the data into a spatial index can optimize these spatial queries.

Spatial indexes structure the data into a search tree, similar to a binary search tree, organizing the points by distance. When querying for nearby points, the algorithm can traverse this search tree, ignoring certain points along the way. While this can speed up the range query to a $O(\log n)$, many spatial indexes degenerate into a $O(n)$ operation. In the former case, this makes DBSCAN run in $O(n \log n)$ time.

In DBSCAN, the `RangeQuery` function handles range queries by taking as parameters the data point and a distance and returning all data points within range of the query point. One can imagine the range query as a hypersphere centered at the query point with a radius of the query range. In order to make `RangeQuery` fast, we had to make some optimizations.

4.3 Clustering Isolates

The CPLOP data we cluster are each isolate's pyroprint[7]. Each pyroprint is a D dimension long vector of positive, real values, where D is 95 and 93 for *ITS-1* and *ITS-2*, respectively. Since there are two regions of pyroprints for any given isolate, we consider both regions at the time of clustering, providing two ε values to query for. The comparison function used in CPLOP to compare pyroprints has been the Pearson correlation coefficient [21, 22]:

$$\rho(\vec{x}, \vec{y}) = \frac{1}{D} \sum_{i=1}^{D} \frac{(x_i - \mu_{\vec{x}})(y_i - \mu_{\vec{y}})}{\sigma_{\vec{x}} \cdot \sigma_{\vec{y}}} = \frac{cov(\vec{x}, \vec{y})}{\sigma_{\vec{x}} \cdot \sigma_{\vec{y}}} \quad (1)$$

Here, $\vec{x} = (x_1, \ldots, x_D)$ and $\vec{y} = (y_1, \ldots y_D)$ are two D-dimensional vectors, $\mu_{\vec{x}}, \mu_{\vec{y}}$ are the means of x_is and y_is respectively, and $\sigma_{\vec{x}}$, $\sigma_{\vec{y}}$ are their standard deviations.

Unfortunately, Pearson correlation coefficient does not encode a metric space, because it fails the triangle inequality[8]. This complicates range queries, discussed in §4.2, because spatial indexes tend to rely on the triangle inequality, usually with euclidean distance, to argue that certain points can be ignored during a spatial index tree traversal.

To allow us the usage of spatial data structure to store data points during the clustering procedure we use instead the euclidean distance pyroprint z-score normalizations, which we derive by recognizing in Equation 1 that Pearson correlation coefficient is made up of z-score normalizations of $\vec{x}$ and $\vec{y}$. The z-score normalization of $\vec{x}$ is:

$$z(x_i) = \frac{x_i - \mu_{\vec{x}}}{\sigma_{\vec{x}}}$$

where $\mu_{\vec{x}}$ and $\sigma_{\vec{x}}$ are the mean and standard deviation of the values in a single pyroprint respectively. Thus, for clustering, we compare pyroprints using the Euclidean distance d of z-scores.

$$d(\vec{z}_{\vec{x}}, \vec{z}_{\vec{y}}) = \sqrt{\sum_{i=1}^{D} (z(x_i) - z(y_i))^2} \quad (2)$$

where D is the number of dimensions. This allows us to use spatial indexes and $O(\log n)$ lookup in DBSCAN.

4.4 Clustering ITS Regions

Each isolate is represented in CPLOP by a pair of pyroprints: one each from each *ITS-1* and *ITS-2* region, complicating the use of DBSCAN and the meaning of α threshold. We chose to handle this in DBSCAN by performing two range queries, one each for *ITS-1* and *ITS-2*, and taking the intersection of the two results. We must, however, pick a suitable ε for each *ITS* region.

CPLOP uses a threshold value of $\alpha = 0.995$ to compare two pyroprints. Pyroprints with Pearson correlation coefficient above α are considered to represent the same DNA material, while pyroprints with a Pearson correlation coefficient below α are considered to represent different DNA material [19, 21, 22].

The number of dispensations D used to build a pyroprint differ for the *ITS-1* and *ITS-2* regions. Because $D_{ITS\text{-}1} \neq D_{ITS\text{-}2}$, the original α under the space defined by (2) no

[7] See §2.3.

[8] $d(x, z) \leq d(x, y) + d(y, z)$

Table 1: Converted α threshold to fit the new metric space defined by (2).

***ITS* Region**	*ITS-1* α	*ITS-2* α
$\rho(\vec{x}, \vec{y})$	0.995	0.995
D	95	93
$d(\vec{z}_{\vec{x}}, \vec{z}_{\vec{y}})$	0.9747	0.9644

longer applies in the same way to both regions. An alternative formulation of (2), with respect to the Pearson correlation coefficient ρ, is:

$$d(\vec{z}_{\vec{x}}, \vec{z}_{\vec{y}}) = \sqrt{2 \cdot D \cdot \rho(\vec{x}, \vec{y})} \tag{3}$$

where D is the number of dimensions and $D_{\vec{x}} = D_{\vec{y}} = D$. Using (3), we can convert α to the values in Table 1. We use these converted α values as the ε for each *ITS* region's `RangeQuery`.

4.5 Optimizations

There are two optimizations we added to DBSCAN. The first we drew from an observation of how DBSCAN queries points, while the second is due to a characteristic of our data. Both of these combined give us a $O(n \log n)$ clustering time complexity.

The first optimization comes from the observation that DBSCAN queries each point only once and nearby points are processed quickly after. Thus, once DBSCAN processes a point and its neighbors, it will never need to return those points to another range query. These points are removed from the spatial index, so that future queries have fewer points to search through.

The second optimization comes from the nature of our pyroprint data and a strong need to have a manageable spatial index. Similar to quadtrees[5] and octrees[10], we considered splitting at multiple dimensions independently. Unfortunately, these approaches increase the number of children nodes exponentially (2^D for D dimensional vectors) and we would eventually need to add 2^{90} children to split on the nearly 100 dimensions each pyroprint has.

Instead, we split multiple dimensions dependently — that is, split once, but have that split cover multiple dimensions. As a result, the split plane will not be axis aligned. The idea is that by changing the angle of the split plane, the average distance from points on either side to the split plane will increase, dependent on a correlation between dimensions. That dependence, for example, may be that points with low values (relative to other points) in one dimension also tend to have low values (relative to other points) in another dimension. Inverse correlations work too, as long as most points with low values in one dimension have high values in another. Fortunately, the pyroprints in CPLOP do exhibit correlations between dimensions.

4.6 Scalability

The scalability of this clustering algorithm is better than anything we have used previously, namely *OhClust!*[12]. In order to make *OhClust!*, a hierarchical clustering method, fast, the Pearson correlation coefficients between all of the pyroprints were precomputed and held in memory. This required the computer it ran on to need over 4GB of memory just to run the clustering. The computers that run the C-PLOP servers only have 4GB total, making the algorithm unable to run locally to CPLOP.

Alternatively, this density-based algorithm can run on a basic laptop in a matter of minutes. Most of the time taken by this algorithm is in querying the database.

5. VALIDATION

The MST procedure described above for an isolate of unknown origin finds the CPLOP cluster it belongs to, and assigns that cluster's host-species of plurality as the query isolate's host-species.

To understand how accurate such a procedure is in determining the host-species of a bacterial isolate, we study the structure of the bacterial strains discovered by our clustering algorithm in CPLOP. After clustering is completed, a CPLOP isolate can be found in one of four different situations:

1. In a *100% pure* cluster.
2. In a cluster where its host-species forms a plurality.
3. In a cluster where its host-species is in minority.
4. As a noise/outlier point.

In this section we describe the measures we use to study the structure of the CPLOP strains, and discuss the details of our experiment.

5.1 Cluster Purity

Our core measure is *cluster purity*. In [9] we described the MST process as essentially a classification task, where the class labels are the names of the host-species. A CPLOP bacterial strain (or cluster) may contain isolates obtained from a single host-species, or from multiple host-species. A *100% pure cluster* is a cluster which only contains data points (isolates) with the same class label (same species of origin).

Consider a cluster $C = \{c_1, \ldots, c_K\}$. Let $s(c)$ refer to the species of isolate c. Let m be the plurality species label for data points in C, and let the total number of points in C with $s(c) = m$ be s_m. Then the *individual cluster purity* ν of cluster C is:

$$\nu(C) = \frac{s_m}{K}$$

In addition to computing the purity of individual clusters we want to have an understanding of the overall purity on the entire dataset. Given a *clustering* $\mathcal{C} = \{C_1, \ldots, C_n\}$ on a dataset, we define the size $\mathcal{M}$ of the set of clusters:

$$\mathcal{M} = \sum_{i=0}^{n} |C_i| \tag{4}$$

The *overall clustering purity* is:

$$\sum_{i=1}^{n} \frac{|C_i|}{\mathcal{M}} \cdot \nu(C_i) \tag{5}$$

One can think of (5) as a form of weighted arithmetic mean of the purities, where the size of the cluster adds more weight to the value.

5.2 Coverage

Coverage of the dataset is important to an effective MST method. Density-based clustering method we use has one key disadvantage: a clustering run with the parameter `MinPts`, treats all points that do not fit into a cluster of size of at least `MinPts` as noise. This means that as the value of `MinPts` grows, so will the number of isolates without a strain.

Given the parameter `MinPts` of the clustering algorithm, we collect the following four measures, that collectively represent the breakdown of all data points (isolates) in CPLOP:

1. *Noise.* Number/percentage of isolates clustered as noise points.

2. *Misses.* Number/percentage of isolates from minority species in impure clusters.

3. *Hits.* Number/percentage of isolates from plurality species in impure clusters.

4. *Pure points.* Number/percentage of isolates in 100% pure clusters.

5.3 Experimental Design

The current state of CPLOP is the result of the organic growth of this database over the course of the past four years. CPLOP is used as the funnel of all the data collected by Cal Poly biology students researching *E. coli* as part of their coursework, senior projects and MS theses. Additionally, CPLOP stores the data collected as a result of a number of larger studies of *E. coli*[1, 3, 15]. In our study, we cluster the part of CPLOP that contains the isolates of known origin[9]

Figure 3 shows the distribution of CPLOP isolates considered in this study among its 53 different host-species.

There are a total of 4,610 isolates in our dataset[10]. As seen from Figure 3, the organic growth of CPLOP yielded disproportionately many *E. coli* isolates originating from humans and cows (however, as shall be seen below, these isolates belong to a large number of strains). Each isolate is represented in CPLOP with two pyroprints — one each for *ITS-1* and *ITS-2* region.

When clustering CPLOP isolates using our density-based clustering algorithm, we need to set up the two parameters at our disposal: `MinPts` and ε. For ε we choose the two values shown in Table 1 converted from the 0.995 Pearson correlation coefficient threshold of pyroprint similarity. Essentially, we only want to consider the ε-neighborhood of a pyroprint that contains the other pyroprints that are considered to represent the same DNA material.

We use *grid search* for the `MinPts` parameter, running our clustering with `MinPts` set to $1, 2, 3, 4, 5, 6,$ and 7.

The `MinPts` value adjusts how strict our definition of a cluster is. That is, the higher the value of `MinPts`, the more neighbors a core point must have with ε of it and its neighbors to become a cluster. Balancing this value with the coverage of our algorithm is crucial to its success, because for too low of a value, we may not have a clear plurality in a cluster, while too high of a value may miss some smaller clusters that might classify our unknown isolate into something other than noise.

[9]As a result of some of the studies, a number of environmental samples of unknown origin is also available in CPLOP. Because we have no ground truth for such isolates, we do not use them in this work.

[10]A simplified version of CPLOP containing isolate IDs, host-species, and z-score normalizations can be found at `https://github.com/jmcgover/cplop-acm-bcb-2016`.

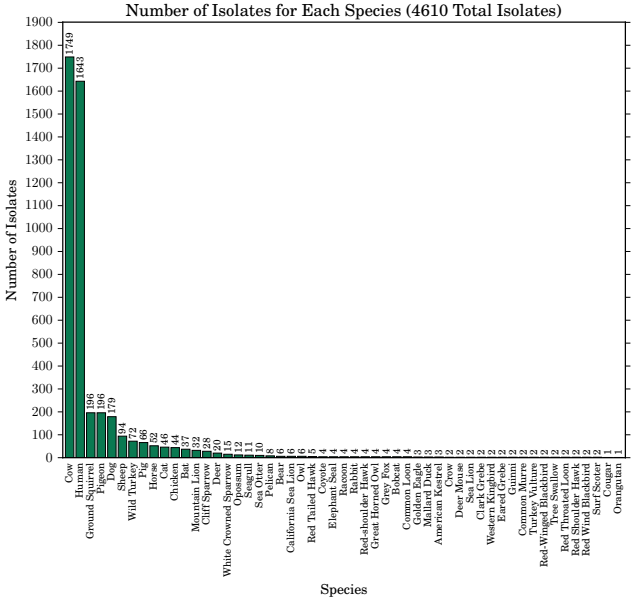


Figure 3: A histogram of the number of isolates of each species in our study, taken from CPLOP. There are 4,610 total isolates from 53 different host-species.

6. RESULTS

In gauging how effective our clustering method is against CPLOP, we looked at the distribution of cluster sizes, the number of isolates that fell into high purity clusters, the number of unique species in each cluster and how that affected the size and purity, and overall coverage and accuracy metrics. From these data, we gained some insight into the clustering algorithm and were able to visualize some predictions we had about the biological aspects of strains.

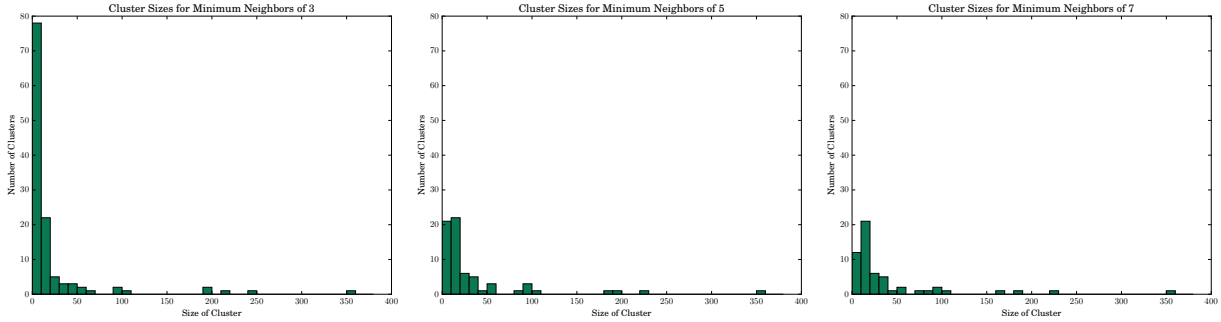
6.1 Cluster Size Distribution

Figure 4 shows the distribution of cluster sizes as `MinPts` increase from 3, to 5, to 7. We see that at all three `MinPts` values, the number of small clusters (fewer than 10) dominates the overall makeup of clusters. Figure 4 shows a propensity towards small clusters at low `MinPts` values. This creates a high number of 100% or almost 100% pure clusters. Most clusters are tiny, with a few larger clusters for small `MinPts` values.

As we approach higher `MinPts` values, the smaller clusters disappear. As `MinPts` increases from 3 to 5, we lose over half of clusters of size smaller than 10; while as `MinPts` increases to 7, we lose only a few more. Furthermore, while the number of clusters with 10-20 isolates stays relatively stable across `MinPts` values, the number of clusters with 50-100 isolates increase for `MinPts` of 5 and 7.

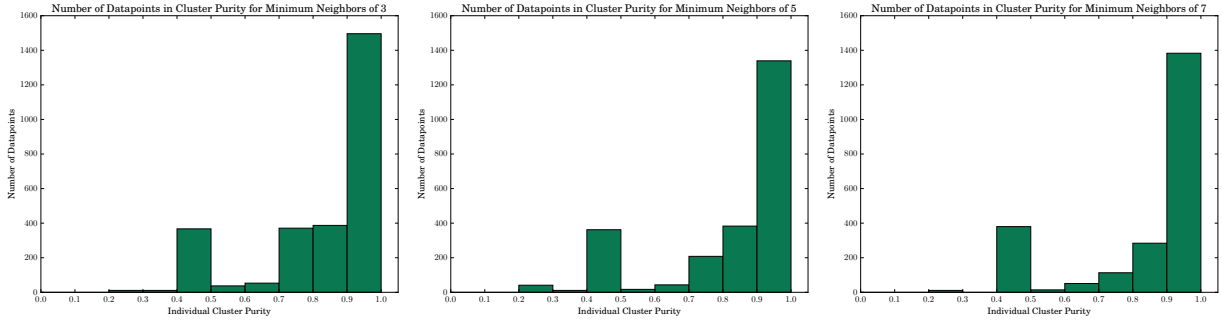
6.2 Cluster Purity Distribution

Of interest to our investigation is the number of isolates that fall within clusters of a particular purity. Figure 5



(a) Cluster Size Distribution for `MinPts` of 3 (b) Cluster Size Distribution for `MinPts` of 5 (c) Cluster Size Distribution for `MinPts` of 7

Figure 4: The size distribution of clusters skews heavily towards smaller clusters.



(a) Cluster Purity Distribution for `MinPts` of 3 (b) Cluster Purity Distribution for `MinPts` of 5 (c) Cluster Purity Distribution for `MinPts` of 7

Figure 5: The number of isolates that fall into a cluster of a given purity. We notice that that the number of isolates that fall into the 0.90 to 1.00 cluster purity range decreases as we increase `MinPts` from 3 to 5, but increases from 5 to 7.

shows the number of isolates that fall within a cluster of a particular purity as a histogram. We notice that as `MinPts` increases, the purity skews towards purer clusters and that a portion of isolates remain in an impure cluster regardless of the `MinPts` value.

From `MinPts` of 3 all the way to 7, there are about 400 isolates that land in a cluster of purity between 0.4 and 0.5. This group of isolates remains largely unchanged as we restrict the `MinPts` value. We suspect (and discuss in §6.6) that certain *E. coli* strains find themselves in many host-species fecal matter.

6.3 Unique Species in Each Cluster

Knowing the number of unique host-species in our clusters is key to understanding how our strain-based MST algorithm performs. Figure 6 plots the number of unique species in each cluster (vertical axis) against individual cluster purity (horizontal axis) representing each cluster as a circle of diameter proportional to cluster size[11]. The points at the lower right represent many clusters of various size of 100% (or near so) purity and are stacked from largest behind cluster to smallest in front.

As `MinPts` values increase we see one cluster at the top right (`MinPts`=3) with 14 unique species disappear as `MinPts` becomes 5. One very low purity cluster at a `MinPts` value of 5 disappears when we increase `MinPts` to 7 in Figure 6c.

A particularly large cluster at around 0.45 purity with 11 unique host-species, remains relatively intact (and is clearly recognizable) as `MinPts` changes from 3, to 5, to 7. This can account for the large amount of isolates clustered into impure clusters in Figure 5.

As we restrict the cluster size with `MinPts`, we see that this appears to break up some clusters and cause others to become bigger. It is difficult to track exactly how a cluster changes without making some simplifying assumptions or without tracking all 4,610 isolates as they move from cluster to cluster.

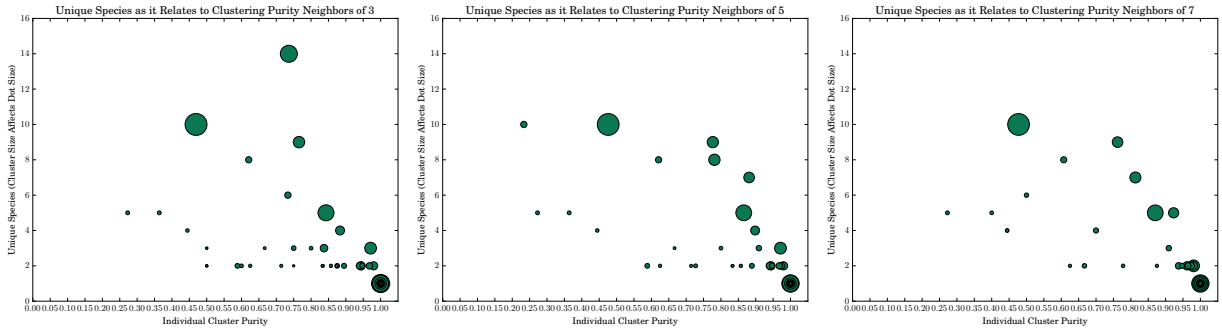
6.4 Clustering Coverage

Clustering coverage is important to consider, since we want our clustering algorithm to apply to as many isolates as possible. Towards this end we investigated the four metrics introduced in §5.2 — noise, misses, hits, and pure points — for each `MinPts` clustering investigated. We hope to find the `MinPts` value that gives us the most pure points, but will also settle for the fewest misses, shown in Figure 7.

The cyan area is noise — isolates that were not clustered. The dark green area is the proportion of pure points. Light green is the number of hits. Gold is the number of misses.

It is good to note that the number of misses are low and flatten out as we increase `MinPts` from a value of 3, giving us good reason not to investigate clustering where `MinPts`

[11] A linear scaling of the diameter with respect to *the largest cluster amongst all the clusterings* defines the diameters of the dots.



(a) Cluster Purity for `MinPts` of 3 (b) Cluster Purity for `MinPts` of 5 (c) Cluster Purity for `MinPts` of 7

Figure 6: These three dimensional graphs show the individual cluster purity in the horizontal axis, the number of unique species in the vertical axis, and the relative size of the cluster in the diameter of the dots. Each individual dot is its own cluster. We find that as we restrict cluster to needing more neighbors (increasing the `MinPts` value), we lose some clusters and gain more pure clusters.

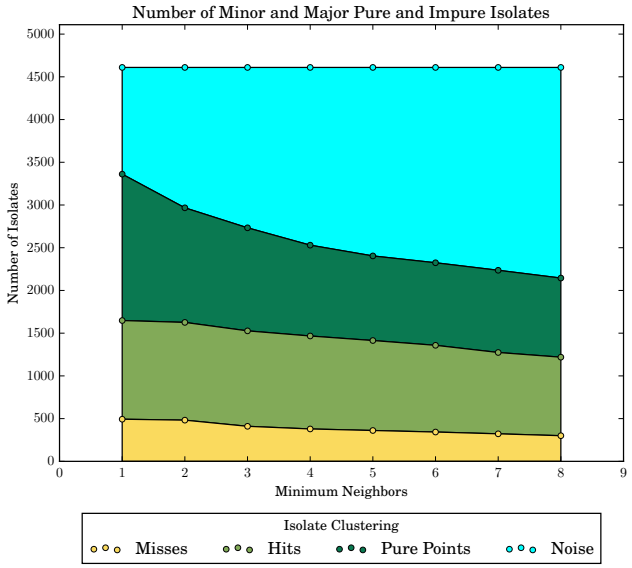


Figure 7: As `MinPts` increases, we see that we cluster fewer isolates. throughout, the number of major pure isolates stays relatively equal to the number of major impure isolates.

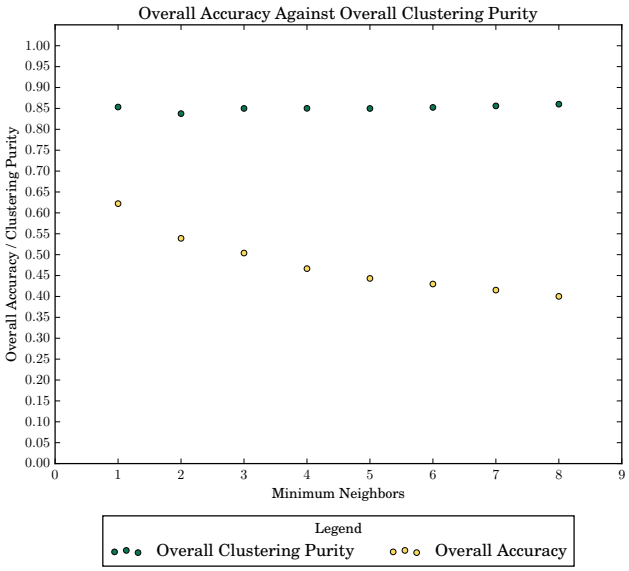


Figure 8: The overall accuracy decreases as we restrict `MinPts`. The overall clustering purity stays relatively the same as we increase the value for `MinPts`. That is, for clustered isolates, the classification algorithm stays relatively the same relative to the number of isolates accurately clustered.

is greater than we have already investigated. The number of pure points stays relatively equal to the number of hits. Important in Figure 7 is the amount of isolates that the algorithm does cluster. The combination of the gold and two green areas show the total number clustered, while the cyan shows the number of isolates that were *not* clustered. It is unfortunate that the number of noise isolates is high, but we plan to mitigate that in future work.

6.5 Overall Clustering Purity

Overall clustering purity, defined in (5), is the number of isolates clustered that end up in a cluster where their host-species is the most plural host-species. The overall accuracy is the proportion of correctly classified isolates out of all the isolates under consideration. We want to maximize both values, but would prefer the former over the latter. Coverage is an issue we are concerned about, but we plan to mitigate this issue by leveraging [9] against clusters of isolates.

Figure 8 shows the overall accuracy compared to the overall clustering purity. A `MinPts` value equal to 3 is the last `MinPts` value where the overall accuracy stays above 0.50. It is not for a lack of correctness, as Figure 7 shows, but more that isolates simply are not being clustered as we restrict the `MinPts` value. In fact, the overall clustering purity in Figure 8 stays relatively constant. This means that if an isolate is clustered by our algorithm, it will likely be clustered with other isolates of the same host-species.

6.6 Discussion

In general, we observe two trends in our data. For the isolates that get clustered into strains, our approach correctly

identifies the host-species with over 80-85% accuracy. This accuracy is sufficient to conduct sophisticated MST studies. Most of the strains discovered in the CPLOP data show high degree of purity, and even considering the presence of a few large impure clusters, most of the clustered isolates fall into strains of high purity.

At the same time, the pure strain-based approach suffers from a drop in the coverage as the size of a cluster grows. This means that in general CPLOP isolates tend to be very diverse and come from strains for which not enough DNA material has been collected and pyrosequenced. Identifying the host-species for isolates that do not fall into strains/clusters using the pure strain-based method is impossible. In future work, our goal is to combine the k-NN-based MST method of [9] with the strain-based approach discussed in this paper to increase coverage while preserving the high MST accuracy.

One factor explaining the large impure clusters is the possibility that these clusters represent what the biologists call "transient" strains, i.e., strains that persist in more than one host-species. Such a characteristic can compound MST by making certain strains of *E. coli* less reliable as FIB for identifying host-species. In Figure 5, we see evidence of that and it is revealed in Figure 6. One mitigation strategy may be to reduce the presence of these strains in library holding the FIB. Another may be to fall back to an alternative MST technique that works with CPLOP when an unknown isolate falls into an impure cluster. Finally, if a true transient strain is indeed discovered, and an isolate is mapped to it, our MST procedure can simply acknowledge the that the query isolate belongs to a transient strain and provide information about the host-species that show high frequency of *E. coli* incidence from this strain.

7. CONCLUSION

In order to combat the issue of contamination of publicly accessible water supplies, namely fecal contamination, the Cal Poly Biological Sciences Department teamed up with the Cal Poly Computer Science Department to build a library-based MST method, called CPLOP. Using the *E. coli* isolated from fecal samples as fecal indicator bacteria, Cal Poly students pyrosequence the PCR-amplified internal transcribed spacer regions of the *E. coli* and store the resulting vector, called a pyroprint, in the CPLOP database for later retrieval, analysis, and comparison.

In this paper, we study the accuracy of a clustering-based MST approach which scales significantly better: the bacterial isolate information stored in CPLOP is clustered using an efficient density-based clustering technique. It clusters data by taking two parameters — the minimum number of neighbors and an ε range that those neighbors must be within to form a cluster — and performs range queries in a spatial index that performs $O(\log n)$ look-up on neighbors using a comparison metric. Compared to previous work [9, 12], it requires fewer comparisons to other isolates and computational resources, by being able to perform reasonably fast clusterings on a consumer laptop in minutes.

To ascertain how well this technique classifies, we build a notion of cluster purity. By calculating the proportion of the entire cluster that the most-plural host-species makes up, we hope to understand how a density-based clustering algorithm clusters the CPLOP data. Furthermore, we inspect coverage and overall accuracy.

Results are that we are able to cluster most of the isolates in CPLOP with high accuracy. Most clusters have high purity and low number of unique species, which is promising for using this for MST. Transient strains are also visible in the clustering technique, which will further aid the biologists working on CPLOP in researching transient strains and improving MST techniques. Future work will leverage other MST techniques designed for CPLOP against this to make up for the lack of coverage and transient strains.

8. REFERENCES

[1] Michael W. Black, Jennifer VanderKelen, Aldrin Montana, Alexander Dekhtyar, Emily Neal, Anya Goodman, and Christopher L. Kitts. Pyroprinting: A rapid and flexible genotypic fingerprinting method for typing bacterial strains. *Journal of Microbiological Methods*, 105:121 – 129, 2014.

[2] Timothy R Desmarais, Helena M Solo-Gabriele, and Carol J Palmer. Influence of soil on fecal indicator organisms in a tidally influenced subtropical environment. *Applied and environmental microbiology*, 68(3):1165–1172, 2002.

[3] Joshua Ryan Dillard. Demographics and transfer of escherichia coli within bos taurus populations. Master's thesis, California Polytechnic State University, San Luis Obispo, 2015.

[4] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[5] Raphael A. Finkel and Jon Louis Bentley. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9, 1974.

[6] Eric Johnson. Density-based clustering of high-dimensional dna fingerprints for library-dependent microbial source tracking. Master's thesis, California Polytechnic State University, San Luis Obispo, 2015.

[7] Jason Kent, Maria Alvarado, Jennifer VanderKelen, Aldrin Montana, Jan Soliman, Alex Dekhtyar, Anya Goodman, Christopher Kitts, and Michael Black. Pyroprinting: novel pyrosequencing-based method for studying e. coli diversity and microbial source tracking (779.8). *The FASEB Journal*, 28(1 Supplement):779–8, 2014.

[8] Wenjun Li, Didier Raoult, and Pierre-Edouard Fournier. Bacterial strain typing in the genomic era. *FEMS Microbiology Reviews*, 33(5):892–916, 2009.

[9] Jeffrey D. McGovern, Alexander Dekhtyar, Chris Kitts, Michael Black, Jennifer Vanderkelen, and Anya Goodman. Leveraging the k-nearest neighbors classification algorithm for microbial source tracking using a bacterial DNA fingerprint library. In Jun Huan, Satoru Miyano, Amarda Shehu, Xiaohua Tony Hu, Bin Ma, Sanguthevar Rajasekaran, Vijay K. Gombar, Matthieu-P. Schapranow, Illhoi Yoo, Jiayu Zhou, Brian Chen, Vinay Pai, and Brian G. Pierce, editors, *2015 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2015, Washington, DC, USA, November 9-12, 2015*, pages 1694–1701. IEEE Computer Society, 2015.

[10] Donald JR Meagher. *Octree encoding: A new*

technique for the representation, manipulation and display of arbitrary 3-d objects by computer. Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory, 1980.

[11] Aldrin Montana. Algorithms for library-based microbial source tracking. Master's thesis, California Polytechnic State University San Luis Obispo, 2013.

[12] Aldrin Montana, Alex Dekhtyar, Michael Black, Chris Kitts, and Anya Goodman. Ontological hierarchical clustering for library-based microbial source tracking. In Wei Ding, Takashi Washio, Hui Xiong, George Karypis, Bhavani M. Thuraisingham, Diane J. Cook, and Xindong Wu, editors, *13th IEEE International Conference on Data Mining Workshops, ICDM Workshops, TX, USA, December 7-10, 2013*, pages 568–576. IEEE Computer Society, 2013.

[13] Aldrin Montana, Alex Dekhtyar, Emily Neal, Michael Black, and Chris Kitts. Investigating temporal strain diversity in human e. coli populations using pyroprinting: A novel strain identification method. Technical report, Technical report, California Polytechnic State University, San Luis Obispo, CA, 2012.

[14] Aldrin Montana, Alexander Dekhtyar, Emily Neal, Michael Black, and Chris Kitts. Chronology-sensitive hierarchical clustering of pyrosequenced DNA samples of e. coli: A case study. In Fang-Xiang Wu, Mohammed Javeed Zaki, Shinichi Morishita, Yi Pan, Stephen Wong, Anastasia Christianson, and Xiaohua Hu, editors, *IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2011, Atlanta, GA, USA, November 12-15, , 2011*, pages 155–159. IEEE Computer Society, 2011.

[15] Emily Neal, Collin Sabatini, Winnie Tang, Michael Black, and Chris Kitts. Demographics of E. coli strains in the human gut using pyroprints: A novel MST method. In *CSUPERB, Poster*. Jan, 2012.

[16] Emily R Neal. Escherichia coli strain diversity in humans: effects of sampling effort and methodology. Master's thesis, California Polytechnic State University, San Luis Obispo, 2013.

[17] Mostafa Ronaghi, Mathias Uhlén, and Pål Nyrén. A sequencing method based on real-time pyrophosphate. *Science*, 281(5375):363–365, 1998.

[18] Troy M Scott, Joan B Rose, Tracie M Jenkins, Samuel R Farrah, and Jerzy Lukasik. Microbial source tracking: Current methodology and future directions. *APPLIED AND ENVIRONMENTAL MICROBIOLOGY*, pages 5796–5803, 2002.

[19] Diana Shealy. Exploration of pyroprinting for environmental forensics. Technical report, California Polytechnic State University, San Luis Obispo, California, June 2012.

[20] Joyce M Simpson, Jorge W Santo Domingo, and Donald J Reasoner. Microbial source tracking: state of the science. *Environmental science & technology*, 36(24):5279–5288, 2002.

[21] Jan Lorenz Soliman. CPLOP: The Cal Poly Library of Pyroprints. Master's thesis, California Polytechnic State University San Luis Obispo, 2013.

[22] Jan Lorenz Soliman, Alex Dekhtyar, Jennifer Vanderkellen, Aldrin Montana, Michael Black, Emily Neal, Kevin Webb, Chris Kitts, and Anya Goodman. Microbial source tracking by molecular fingerprinting. In Sanjay Ranka, iTamer Kahveci, and Mona Singh, editors, *ACM International Conference on Bioinformatics, Computational Biology and Biomedicine, BCB' 12, Orlando, FL, USA - October 08 - 10, 2012*, pages 617–619. ACM, 2012.