Predicting Watershed Characteristics Using Bacterial DNA

by
Jessica Chadwick

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Ecological Engineering
(Honors Scholar)

Presented May 28, 2019
Commencement June 2019

AN ABSTRACT OF THE THESIS OF

Jessica Chadwick for the degree of Honors Baccalaureate of Science in Ecological Engineering presented on May 28, 2019. Title: Predicting Watershed Characteristics Using Bacterial DNA.

Abstract approved:____
Stephen Good

This study was conducted to determine if bacterial DNA present streams could be used to predict upstream watershed characteristics. Previous studies have found that bacterial composition in soil is influenced by land use. It was hypothesized that if the bacteria present in a stream is known that it can be used to predict upstream watershed characteristics. Collecting bacterial data involved sampling at 62 different sites in Oregon. The bacterial DNA from these samples were then extracted resulting in a spreadsheet of operational taxonomic units (OTUs). Land cover characteristics for each site were obtained by delineating each site's watershed in StreamStats. The OTU and StreamStats data were used as inputs to create a model using support vector regression (SVR) in python to predict land cover characteristics. The SVR inputs kernel and C value were manipulated to improve the model along with the prevalence of OTUs. The largest Nash-Sutcliffe efficiency (NSE) value obtained when manipulating the model for forest and shrub cover was 0.26 using an 'rbf' kernel, C value of 20433 and a prevalence greater than 91%. This indicates that the model produces a better prediction of land coverage than using the average of all the sites' land cover.

Key Words: land cover, bacterial DNA, support vector regression, Nash Sutcliffe efficiency

Corresponding e-mail address: chadwicj@oregonstate.edu

Predicting Watershed Characteristics Using Bacterial DNA

by
Jessica Chadwick

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Ecological Engineering
(Honors Scholar)

Presented May 28, 2019
Commencement June 2019

Honors Baccalaureate of Science in Ecological Engineering project of Jessica Chadwick presented on May 28, 2019.

APPROVED:

____

Stephen Good, Mentor, representing Biological & Ecological Engineering

____

Byron Crump, Committee Member, representing Earth, Ocean, and Atmospheric Sciences

____

Gerrad Jones, Committee Member, representing Biological & Ecological Engineering

____

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, Honors College. My signature below authorizes release of my project to any reader upon request.

____

Jessica Chadwick, Author

Acknowledgements:

I would like to thank Stephen Good from the department of Biological & Ecological Engineering at Oregon State University for being my advisor throughout the undergraduate thesis process. Stephen Good provided the idea behind this research and guidance along the way. I would like to thank Stephen Good for the opportunity to work in his lab and gain experience in field work throughout Oregon. I would also like to thank Stephen Good for providing me guidance in using support vector regression (SVR) in python.

I would like to thank Dawn Urycki, a PhD student in Water Resources Engineering, for letting me participate in her graduate research. Dawn allowed me to gain field work experience by conducting the sampling around Oregon for her research. I would also like to thank Dawn for allowing me to gain experience in DNA extraction through extracting samples from her study. Dawn also provided guidance in the SVR modeling and pictures of her sampling and testing that can be seen throughout this document.

I would like to thank Byron Crump from the College of Earth, Ocean, and Atmospheric Sciences at Oregon State University. Byron Crump dedicated a lot of time to train me in the DNA extraction process described later in this document. He allowed me to complete DNA extractions in his lab in Weniger Hall using resources in this lab. I would like to thank Byron Crump for also being on my undergraduate committee and providing me guidance throughout the thesis process.

I would like to thank Gerrad Jones from the department of Biological & Ecological Engineering at Oregon State University for being on my undergraduate thesis committee. Gerrad Jones talked through my results with me and provided guidance throughout the undergraduate thesis process.

Table of Contents:

1. Introduction

The goal of this research was to investigate the relationship between bacteria present in a stream and its upstream watershed land cover characteristics. Bacteria have great importance because they are one of the most abundant organisms on the planet. This is likely due to their ability to grow in so many conditions and habitats. Bacteria can be found deep in the ocean, frozen in ice in Antarctica, at the tops of mountains, and in the guts of animals. There are many types of bacteria, but there are general groupings based on their need for oxygen and how they obtain their energy. Ecosystems depend heavily on bacteria to cycle nutrients like carbon, sulfur and nitrogen. Bacteria play an important role in the decomposition of organic matter. Plants can obtain nitrogen from the atmosphere through nitrogen fixation which is carried out by bacteria that convert gaseous nitrogen into nitrites and nitrates through their metabolism. Bacteria also complete denitrification, turning nitrate into a gas which can cause depletion of nutrients in soil (UCMP, 2019). Bacteria carry out many important processes, so it is necessary to understand what influences bacterial composition.

A study in China investigated “the Effects of Land Use on Hydrochemistry and Soil Microbial Diversity”. This study assessed the impacts of land cover through biological and chemical data from four different land use groups. These land use groups included bare land, land growing peaches, land growing castanea, and land with pine growing on it. This study stated that land use is the key factor of human disturbance that is affecting ecosystems. The different land uses change the amount of gas, liquid, and solid in the soil which then affects microbial activity in the soil. The results of this study indicated that microbial diversity varied with different vegetation forms (Zhang et al., 2019). Another study in New Zealand investigated “Bacteria as Emerging Indicators of Soil Condition”. This study aimed to understand how bacterial communities are affected by anthropogenic activity and to use them as an indicator of environmental health. This study investigated 110 soil samples in natural and human-impacted areas that were up to 300 km apart. The results of this study showed that there was a relationship between the bacterial community and their environment. They found these relationships between bacteria and soil variables that are known to be influenced by human activity. This suggested that the bacterial community could indicate the condition of the soil (Hermans et al., 2016).

According to the review titled "Transport of Microorganisms Through Soil", many studies have been executed to understand how bacteria move through soil and enter water sources. One explanation is preferential flow of microorganisms through cracks, holes formed by plants or animals, macropores and fractures. These studies found that the factors that affect the movement and survival of microorganisms are related to the interactions between water, the surrounding environment, soil, and microorganisms. Soil bulk density, soil texture, size and morphology of microorganisms, and the presence of plants or other living organisms all have been shown to influence bacterial transport through soil. The transport mechanisms for bacteria in soil can be categorized into physical, geochemical, and biological processes. The physical processes that influence microbial movement are advection, convection, and hydrodynamic dispersion. Geochemical processes usually influence microbial movement through soil by delaying movement through adsorption, filtration, and sedimentation (Abu-Ashour et al., 1994).

This previous research suggests that the bacterial communities in soil vary with land use. Using this information, it was predicted that if the bacteria present in a stream is known, that it can be used to predict the upstream watershed characteristics. To test this prediction sampling was done around Oregon to collect bacterial composition using Sterivex filters. The bacterial DNA was then extracted and sequenced resulting in operational taxonomic units (OTUs). Watershed characteristics for the sampling sites were estimated using StreamStats. The OTUs and watershed characteristics were then used with support vector regression (SVR) in python to develop a model to predict a watershed characteristic for a certain site using the OTUs for that site, the OTUs for the other sites, and the land cover characteristic for other sites. This model could then be used as a tool to predict watershed characteristics based on the bacterial composition of the water.

2. Methods

2.1 Sampling Method

DNA samples were collected from water sources in 62 different locations in Oregon. These sites were chosen because of their proximity to USGS gauges so that hydraulic data could be used. Of these 62 sites, 5 were within the middle coast basin, 20 were within the Deschutes basin, and 37 were within the Willamette basin, with 10 of the 37 specifically in the HJ Andrews experimental forest. A map of the sampling site locations can be seen below in Figure 1.

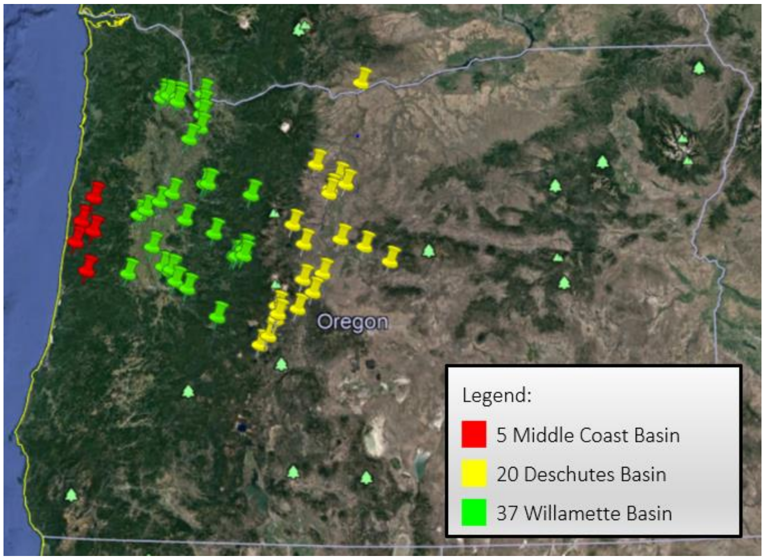


Figure 1: Map of sampling sites (Google Earth, 2019)

At each site an acid washed bucket was filled with water from the center of the river or stream. Then this water was emptied to perform a clean transfer and the bucket was filled again. With gloves on, the tip of a 25 mL disposable plastic pipette was broken off while still in the packaging. Then the pipette packaging was opened on the plugged end and the cotton was removed from the end with autoclaved tweezers. Then this end was inserted into autoclaved flex tubing while trying not to touch the ends and keeping as much of the pipette inside the packaging as possible to prevent contamination. Once the pipette and the tubing were connected, the tube was carefully inserted into the pump head of a Geopump^{TM} peristaltic pump (Geotech Environmental Equipment, Inc.). The bucket filled with sample was placed below the pump, the pipette was placed in the sample, and the pump was turned on. As the sample was being pumped out of the bucket the pipette was moved in a slow figure 8 motion to ensure appropriate mixing of the sample. Enough sample was pumped through the tube to coat all the surfaces to ensure a clean transfer and then the pump was turned off. After this, two MilliporeSigma™ Sterivex Sterile Pressure-Driven filters were labeled and then screwed on to the ends of the tubing and a graduated cylinder was placed under each filter. This sampling apparatus can be seen below in Figure 2.

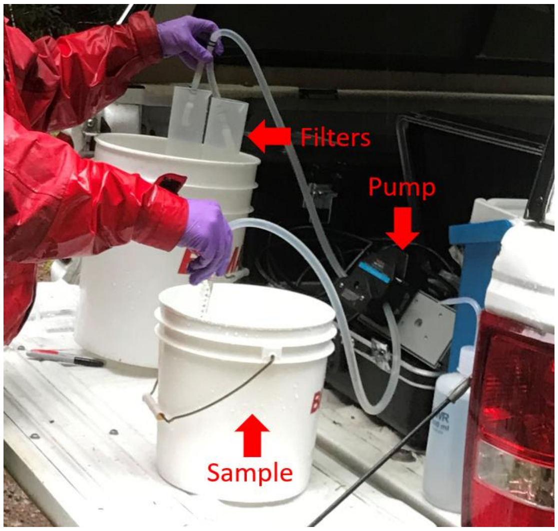


Figure 2: Picture of the sampling apparatus (Dawn Urycki)

Once the apparatus was set up, the pump was turned on and the volume pumped through each filter was recorded using the graduated cylinders until the filters clogged.

Then the pump was turned off and the filters were removed from the tubing. After the filters were removed, DNA extraction buffer was inserted into each filter, autoclaved lock plugs were screwed on to the wide end of each filter, and the thin end of each filter was sealed with putty. The filters were then placed in a labeled Ziploc bag and placed in a cooler on dry ice. Then the equipment was rinsed with DI water before moving on to the next site. The procedure was repeated for every site that was visited during the day in the field. After returning from a day of sampling the samples were all stored in a -80°C freezer until DNA extraction was performed.

2.2 DNA Extraction Method

The DNA extraction method used was developed by Byron Crump in 2007 and the samples were extracted in Byron Crump’s lab at Oregon State University. First the samples were pulled out of the freezer to thaw and a sterile scalpel was prepared. Then the scalpel and two pairs of forceps were placed in a small container of ethanol. Then, under a laminar flow hood, using pliers, the outport end of the filter was cracked open. Then the filter barrel was pulled out of the plastic casing and placed on a sterile disposable petri dish. After the filter barrel was removed the buffer was poured into a 2 mL microcentrifuge tube removing the lock plug to ensure all the buffer was emptied and the plastic casing was discarded. Then an ethanol flame was lit, and the scalpel and forceps were passed through the flame to burn off the ethanol and sterilize them. Then the scalpel was used to cut the white filter off the plastic barrel. Then the plastic barrel was thrown away and the filter was placed in the sterile petri dish. The forceps were then used to fold the filter in half with the side with organic matter on it on the inside. Then while folded in half the filter was sliced into thin strips and the forceps were used to place these strips in the 2 mL microcentrifuge tube submerged in the buffer. After this the 2 mL microcentrifuge tube was closed, the petri dish was thrown away, and the blade and forceps were placed back in ethanol (Crump, 2007). This process was repeated for however many samples were being processed that day. The filter cutting process can be seen below in Figure 3.

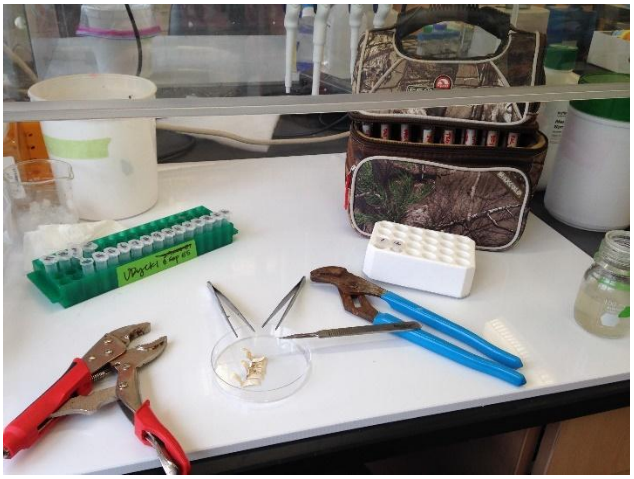


Figure 3: Photo of the filter cracking and cutting process (Dawn Urycki)

The next step in the extraction process was to add proteinase-K and lysozyme to the sample and freeze-thaw. First, 20 µL of 10 mg/ml proteinase-K and 20 µL of 100 mg/ml lysozyme were added to each microcentrifuge tube. The proteinase-K is used to digest any contaminating proteins that may be present, and the lysozyme is used to lyse the bacterial cells. Lysing a bacterial cell involves perforating the bacterial cell wall without denaturing the proteins (Genlantis, 2017). After the enzymes were added, the samples were placed in a -80 °C freezer for 15 minutes or in dry ice until they were frozen. Then the samples were placed in a 37 °C water bath for 5 minutes or until the samples had thawed. This process was repeated three times. The third time the samples were left in the 37 °C water bath for another 30 minutes to incubate (Crump, 2007). This freeze-thaw method is used to lyse the bacterial cells as well. The dry ice and water bath used for the freeze thaw process can be seen in Figure 4 below.

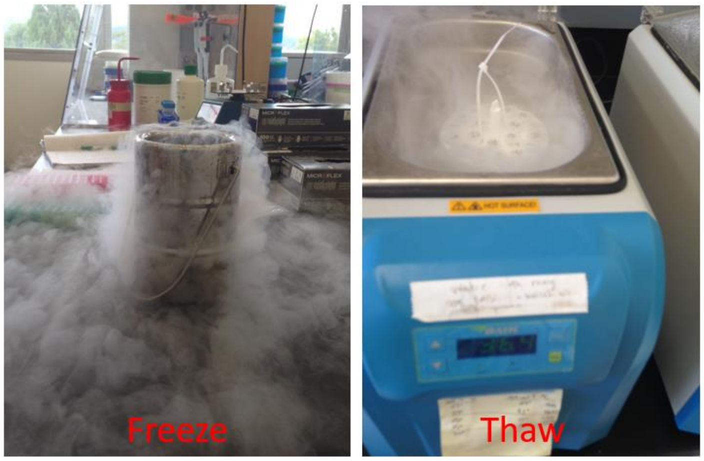


Figure 4: Dry ice freezing (left) and the 37 °C water bath thawing (right) (Dawn Urycki)

After the 37 °C water bath incubation, 50 µL of filter-sterilized sodium dodecyl sulfate (SDS) was added to each microcentrifuge tube. Then each tube was inverted several times to properly mix. These tubes were then placed in a 65 °C water bath to incubate for at least one hour (Crump, 2007). The SDS is used to disrupt the cell membrane during the lysis process.

The next step was phenol-chloroform extraction. Underneath a fume hood, all the microcentrifuge tubes were filled the rest of the way with phenol-chloroform-isoamyl alcohol (25:24:1, pH 8.0). The phenol-chloroform was added using a sterile glass pipette with a rubber pipette bulb. Then the samples were placed on a vortex until the filter paper had noticeably started to degrade. Then the microcentrifuge tubes were centrifuged at 3000 rpm for 5 minutes (Crump, 2007). The phenol-chloroform in this step helps to separate the cellular debris in the organic phase from the DNA in the aqueous phase (McKiernan & Danielson, 2017). After the five minutes of centrifuging, the top aqueous phase, containing DNA, was transferred into a new microcentrifuge tube. When performing this extraction with the Sterivex filters, there was usually a layer of the filter material that helped to separate the organics and the aqueous DNA. Once the top layer of aqueous DNA was transferred to a new tube the new tube was filled with phenol-chloroform and centrifuged again as a second wash. Then the top aqueous DNA layer from the second wash was transferred to a new microcentrifuge tube (Crump, 2007). A visual summary of this extraction process can be seen in Figure 5 below.

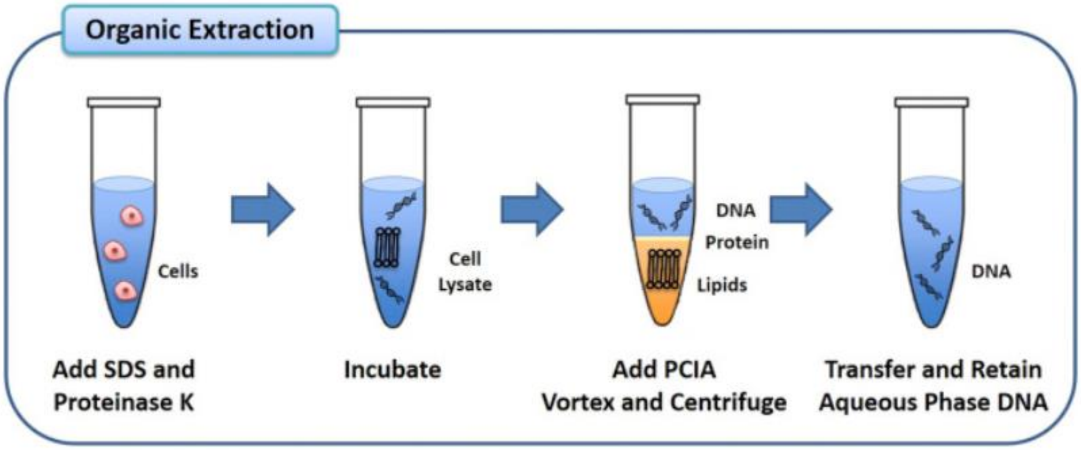


Figure 5: A visualization of DNA extraction using phenol-chloroform-isoamyl alcohol (PCIA) (McKiernan & Danielson, 2017)

After the second wash with phenol-chloroform the resulting buffer volume was estimated. This volume was then used to determine what volume of isopropanol to add by multiplying the buffer volume by 0.6. This volume of room temperature isopropanol was added to the buffer and the tube was gently inverted to mix. These tubes were then placed in a drawer in the dark at room temperature to precipitate overnight (Crump, 2007). Isopropanol is used in DNA precipitation because DNA is insoluble in isopropanol so adding it to the buffer causes the DNA to come out of the solution (BiteSizeBio, 2018).

After leaving the tubes overnight they were placed in a microcentrifuge at 13000 rpm for 30 minutes. Then the solution was removed with a pipette making sure not to disturb the pellet of DNA. After the solution was removed and disposed of, 1 mL of 70% EtOH was added to each tube and the tubes were inverted several times. Then the tubes were placed in the centrifuge again at 13000 rpm for 10 minutes. After removing the tubes from the centrifuge, the EtOH was removed with a pipette and 1 mL of 70% EtOH was added for a second rinse. This EtOH rinse washes away any salt that may have precipitated from the pelleted DNA. After the second rinse the EtOH was removed from the tube and the tubes were placed open in the roto-evaporator for 15 minutes or until the pellets were dry (Crump, 2007). An image of a pellet resulting from this study can be seen in Figure 6 below.

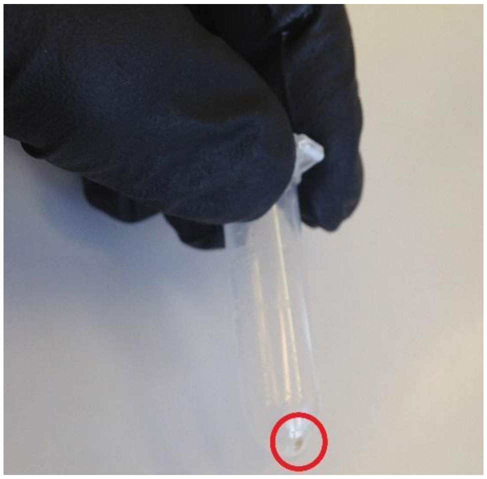


Figure 6: An image of a pellet (in the red circle) from this study (Dawn Urycki)

After the pellets were dry they were resuspended in 250 µL of autoclaved UV sterile ultra-pure water. The liquid was flicked at the bottom of each tube to ensure that all possible DNA was covered. The tubes were then left in the refrigerator for 2 hours. After 2 hours 100 µL of the solution was put in a working screw-cap cryovial and 150 µL of the solution was put in an archive screw-cap cryovial (Crump, 2007). These tubes were then frozen in a -80 °C freezer until they were needed for PCR.

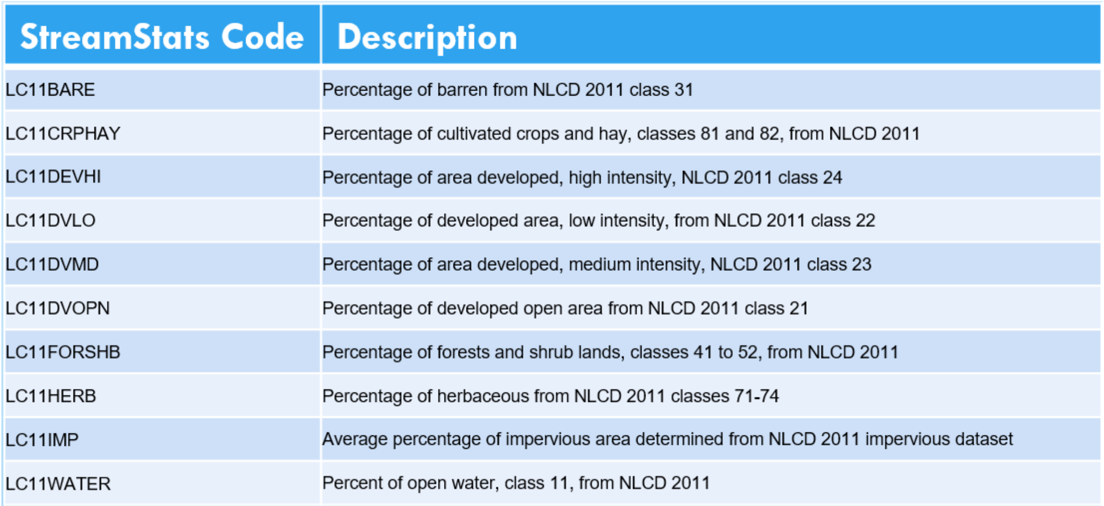
After the extraction, bacteria-specific primers (515F GTGCCAGCMGCCGCGGTAA, and 806R GGACTACHVGGGTWTCTAAT). were used to PCR-amplify the V4 region of the16S rRNA gene. Then the PCR products were analyzed with agarose gel electrophoresis to make sure that the PCR was successfully amplified. PCR products were purified, and concentrations normalized using the SequalPrep Normalization plates (Thermo-Fisher). Next, PCR products were combined in equimolar quantities and sent to the Oregon State University Center for Genome Research and Biocomputing to get sequenced. Then the sequences were quality filtered and grouped into operational taxonomic units (OTUs) with 95% similarity. Then the OTUs were rarefied to 2000 sequences per sample and this was the dataset that was used in the SVR analysis in python (Urycki et al, 2019).

2.3 Data Analysis

2.3.1 *StreamStats*

The USGS tool StreamStats was used to obtain watershed characteristics for each of the 62 sampling sites. This was done by first going to the StreamStats application and entering the latitude and longitude of a site into the search bar. Once this location was found the state or regional study area of Oregon was selected. Then once at zoom level 15 or greater the delineate button was selected. Next the spot closest to the sampling site within the available points was selected. Then the option to continue was chosen and all basin characteristics were selected and continue was chosen twice more to produce a basin characteristics report for that location. An excel spreadsheet was compiled containing each characteristic for the 62 different locations. Eventually through the python code discussed in the next section, the percent land cover characteristics were extracted for each site from the excel file (USGS, 2019). The characteristics used in data analysis can be seen below in Table 1.

Table 1: StreamStats percent landcover characteristics used in data analysis (USGS, 2019)



StreamStats Code	Description
LC11BARE	Percentage of barren from NLCD 2011 class 31
LC11CRPHAY	Percentage of cultivated crops and hay, classes 81 and 82, from NLCD 2011
LC11DEVHI	Percentage of area developed, high intensity, NLCD 2011 class 24
LC11DVLO	Percentage of developed area, low intensity, from NLCD 2011 class 22
LC11DVMD	Percentage of area developed, medium intensity, NLCD 2011 class 23
LC11DVOPN	Percentage of developed open area from NLCD 2011 class 21
LC11FORSHB	Percentage of forests and shrub lands, classes 41 to 52, from NLCD 2011
LC11HERB	Percentage of herbaceous from NLCD 2011 classes 71-74
LC11IMP	Average percentage of impervious area determined from NLCD 2011 impervious dataset
LC11WATER	Percent of open water, class 11, from NLCD 2011

2.3.2 *SVR Analysis in Python*

All the data analysis was conducted in Python. The first step included importing both the excel file containing the OTUs for each sampling site and the file containing all the compiled StreamStats land cover characteristics. It was decided to only use the percent land cover characteristics above in Table 1 so only those columns of the StreamStats excel file were imported. The OTU data frame was standardized by dividing the entire spreadsheet by 2000 since each sites OTUs summed to 2000. Then a prediction was defined to use support vector regression (SVR) to predict land cover characteristics for a site using the bacterial DNA present in the water at that site. The inputs for this prediction are the training OTUs, the OTUs for that site, and the training StreamStats data. The training OTUs are the OTU spreadsheet without the column of OTUs for the site the land cover is being predicted for. The OTUs for that site would be the column removed from the training OTUs. The training StreamStats data is the column from the StreamStats data that contains the land cover characteristic being predicted without the value for the site that is being predicted. Then the output of this prediction is a predicted value for that land cover characteristic at that site that is calculated through the model developed by SVR.

The predictions were evaluated by calculating the Nash-Sutcliffe efficiency (NSE). The equation for this metric can be seen below in Figure 7. NSE is a way to determine if a prediction is better than just using the mean of the data. NSE values can range from negative infinity to 1 with 1 being the best possible value which would occur if the modeled value was the same as the observed value. An NSE value of 0 would mean that the error associated with the model is equal to the error associated with using the mean of the observed values. A negative value of NSE would mean that the error associated with the model is greater than the error associated with using the mean of the observed values as a prediction. A positive value would mean the opposite.

$$NSE = 1 - \frac{\sum(modeled - observed)^2}{\sum(observed - \overline{observed})^2}$$

Figure 7: Equation for Nash-Sutcliffe efficiency

This prediction relies on support vector regression (SVR) from the sklearn package in python. SVR has many inputs but the inputs that were manipulated in this study to develop a better model were the kernel and C value. The kernel in SVR specifies the kernel type to be used in the algorithm. The options for the kernel are ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’, or a callable. The default kernel if not specified otherwise is ‘rbf’. Only ‘linear’, ‘rbf’, and ‘poly’ were considered for this model. The C value is a penalty parameter C of the error term. If a larger value of C is specified that means that a smaller margin is accepted by the decision function. Therefore, a higher C value should result in better classifying or training of the model (Scikit-learn, 2007 - 2019).

The first test involved using the prediction for all land cover characteristics and OTUs. The values for C tested ranged from $10^0 – 10^5$ with 60 log spaced values in-between. The kernels evaluated in this first test included ‘linear’, ‘rbf’, and ‘poly’. For loops were used to test the prediction with each possible combination of the C values and kernels mentioned. To understand how well this model predicted land cover both RMSE and NSE were calculated for every combination of kernel and C. The only land cover characteristic that had a positive NSE was percentage of forests and shrublands (LC11FORSHB).

This information was used in the second test where the kernel was specified as linear and the C value was set to 3.22. With these two parameters already set different prevalence values for OTUs were tested to try to improve the model. Before this could be done the prevalence of each OTU had to be calculated. This was done by replacing any value other than 0 in the OTU data frame with a 1 to represent that a certain OTU was present at a site. Then the rows were summed across and divided by the 62 total sites and multiplied by 100 to determine the percentage of sites that a certain OTU was present in. Then a for loop was created to cycle through all the possible percentage values (0-99) and determine where in the prevalence data frame there was a value greater than that percentage. The 'where' function in python returns an index, so within the loop that index was used to find the original OTU values for those locations and add them to a new data frame. The result of this loop was a new data frame filled with only values for OTUs that had a prevalence greater than the percentage specified. After this code was developed the for loop was setup to predict land cover with a 'linear' kernel and a C value of 3.22 with all possible prevalence values. It was then considered that different C values might work for different prevalence values.

A third test was initiated that only used a 'linear' kernel but looped through different C values and prevalence values. A range of $10^0 – 10^2$ with 30 values log spaced was used for C and 0-99 was used for prevalence. After this it was considered that different prevalence values may be better for different kernels.

The fourth test involved using for loops to test every possible combination of prevalence, 5 log spaced values of C within the range $10^0 – 10^5$, and kernel. The highest NSE resulting from this test was 0.21 with a rbf kernel, C value of 100,000, and a prevalence greater than 86%. Because the largest NSE was the last value of C tested the same test was run again but with 5 log spaced values of C within the range $10^0 – 10^6$. The results from each test described can be compared in the side by side image in Figure 16 below. The entire code in python for this last analysis in step 4 can be seen in Appendix A.1.

3. Results

Step 1:

The highest NSE value for this land cover was 0.171 with a 'linear' kernel and a C value of 3.22. A graph of all the NSE values for each kernel resulting from the range of C values specified can be seen below in Figure 8. A graph of measured and predicted values of forest and shrub cover using a linear kernel and a C value of 3.22 can be seen below in Figure 9.

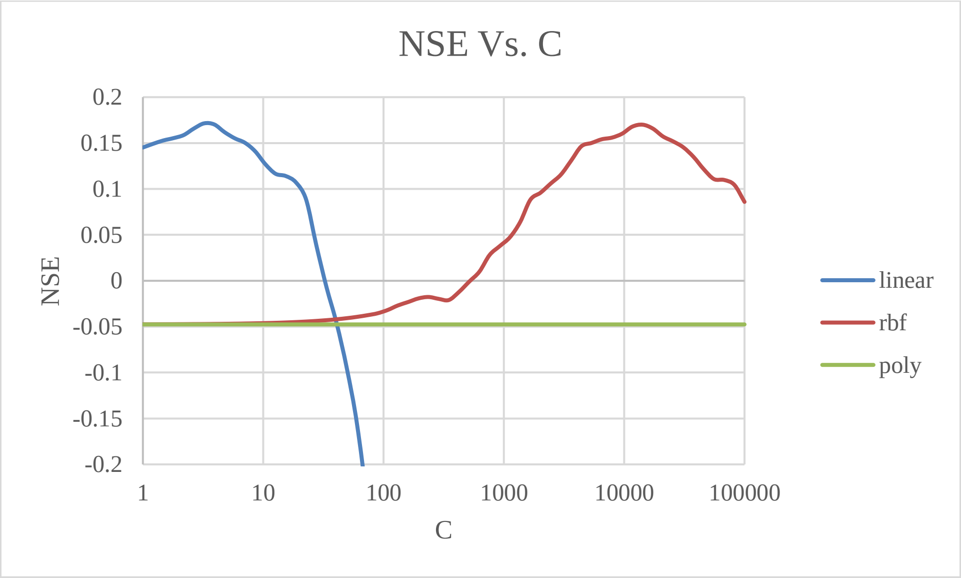


Figure 8: Graph of C values tested in step 1 with all three kernels and their resulting NSE.

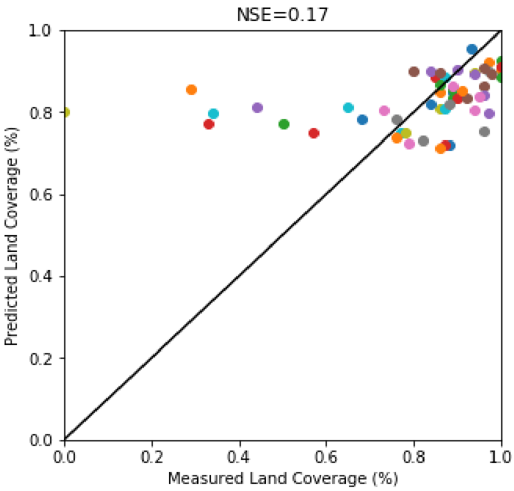


Figure 9: A graph of the measured and predicted LC11FORSHB resulting from step 1.

Step 2:

The largest NSE value was 0.177 with a prevalence of greater than 20%. A graph of the NSE from different prevalence values can be seen below in Figure 10. A graph of measured and predicted values of forest and shrub cover using a linear kernel, a C value of 3.22, and OTU prevalence of >20% can be seen below in Figure 11.

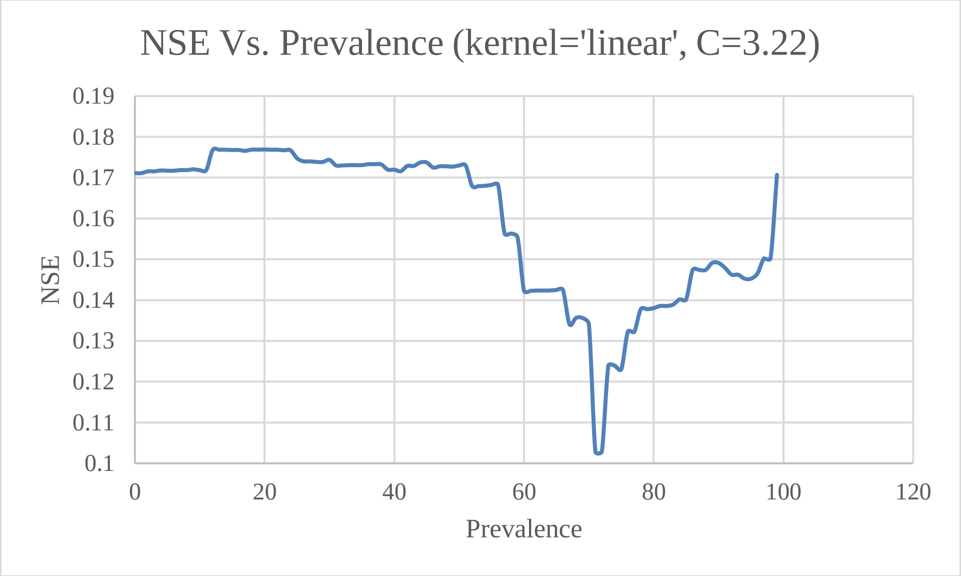


Figure 10: NSE values for the range of OTU prevalence with a linear kernel and a C value of 3.22.

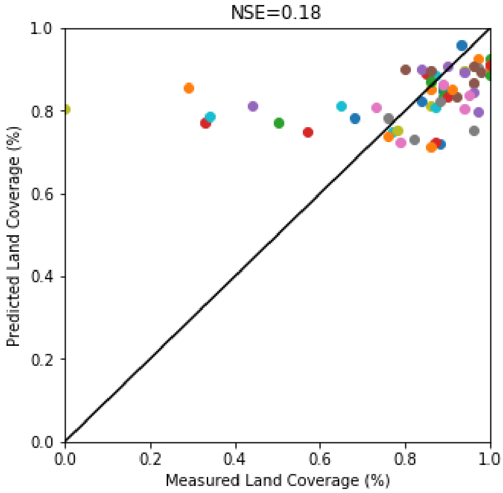


Figure 11: A graph of the measured and predicted LC11FORSHB resulting from step 2.

Step 3:

The largest NSE value that resulted from this was 0.2 with a C value of 100 and a prevalence greater than 99%. A plot of the NSE resulting from different combinations of C and prevalence can be seen below in Figure 12. A plot of the predicted forest and shrub cover and the StreamStats measured forest and shrub cover with a linear kernel, C value of 100 and a prevalence greater than 99% can be seen below in Figure 13.

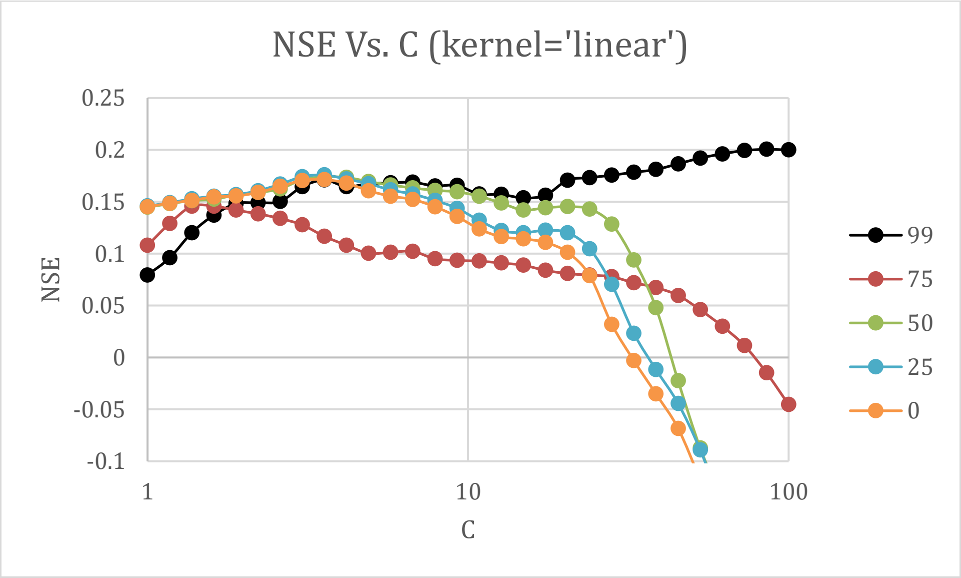


Figure 12: Graph of resulting NSEs from different C and prevalence values with a ‘linear’ kernel.

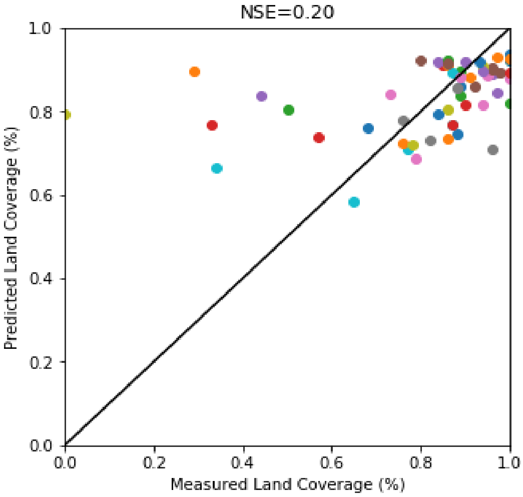


Figure 13: A graph of the measured and predicted LC11FORSHB resulting from step 3.

Step 4:

The largest NSE from this test was 0.22 with a rbf kernel, C value of 31,622.78, and a prevalence greater than 91%. The NSE values from this test with different prevalence values can be seen in Figure 14 below. In this graph a prevalence value of 91 has the highest peak which appears to be between C values of 10,000 and 100,000. This test was modified again to have 30 log spaced values of C within the range $10^4 – 10^5$. The largest NSE from this modified test was 0.26 with a rbf kernel, C value of 20,433, and a prevalence greater than 91%. A plot of the predicted forest and shrub cover and the StreamStats measured forest and shrub cover with a rbf kernel, C value of 20,433 and a prevalence greater than 91% can be seen below in Figure 15.

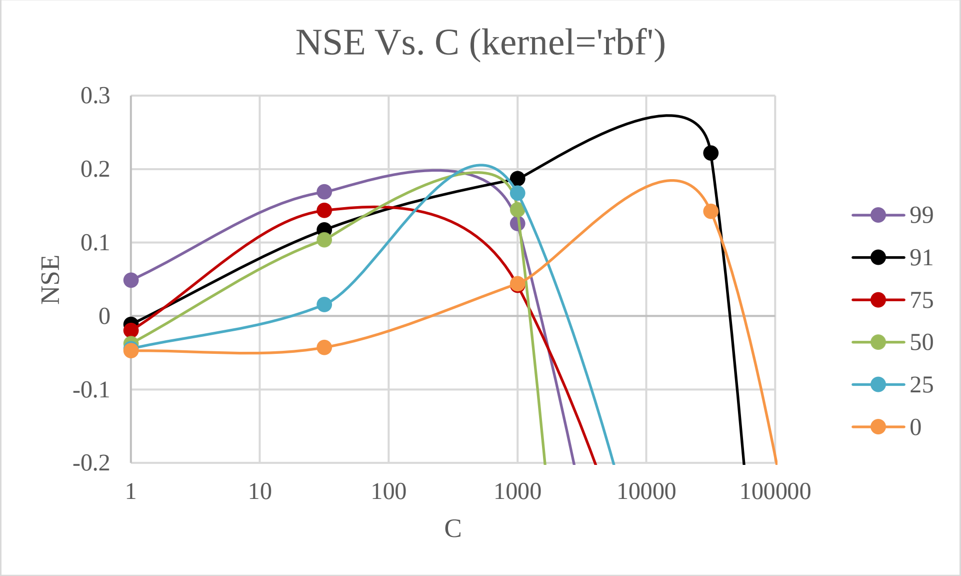


Figure 14: Graph of NSEs from different C ($10^0 – 10^6$) and prevalence values with a ‘rbf’ kernel.

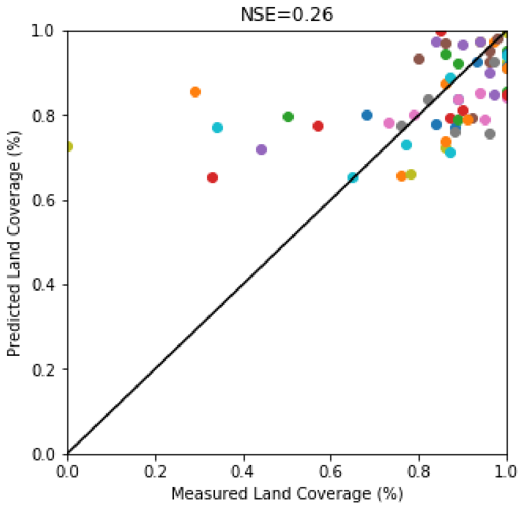


Figure 15: A graph of the measured and predicted LC11FORSHB resulting from step 4.

Overall Results:

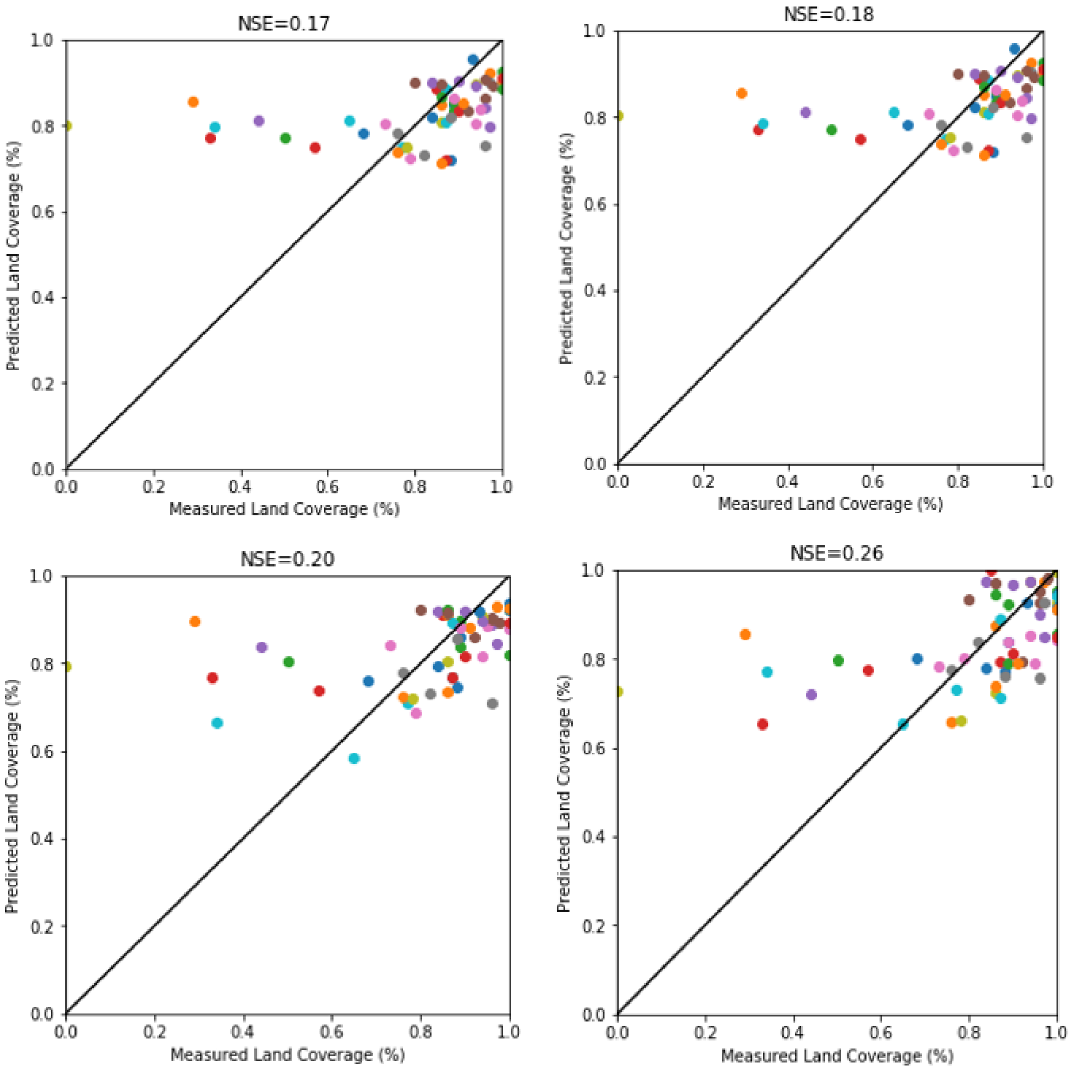


Figure 16: Comparison of model performance at each of the four steps in the python methods

4. Discussion

In the first step of python analysis Figure 8 shows that different kernels have different ideal values of C. This can be seen in the two different peaks for the linear and rbf kernels. It is important to understand that there may be multiple ranges of ideal values of C when applying and tuning this model. The NSE then increased from 0.17 to 0.18 when prevalence was accounted for. As an input to the model, it makes sense that an OTU that may only be present in one site may not be a significant indicator when tuning the model. A prevalence of greater than 99% resulted in the best model for a linear kernel but greater than 91% resulted in the best model for a rbf kernel which can be seen in Figure 12 and Figure 14 above. This shows that as the kernels change so do the ideal tuning parameters. The final step resulting in the highest NSE value of 0.26 showed that if multiple parameters are being tuned that they need to all be tuned together because of their dynamic relationships that influence the efficiency of the model.

The 0.26 value obtained at the end of this testing means that the model is better than just using the average of a land cover characteristic to make a prediction for a watershed. This conclusion is made by simply looking at the equation for NSE in Figure 7. An NSE value is 0 if a model produces the same results as taking the average of the characteristic. An NSE value of 1 would represent a perfect model because the only way 1 is the result from that equation is if the observed land cover is equal to the predicted land cover. A positive NSE value means that the difference between the predicted value and the mean of the observed value is less than the difference between the observed value and the mean of the observed. If the equation is manipulated with the 0.26 value one can see that the error associated with the model is 0.74 times the error associated with using the mean. This shows that a model was developed in python that could predict watershed characteristics using bacterial DNA better than using the average of the observed land cover values to make a prediction.

4.1 Application

This model could most likely be applied to locations within Oregon with little manipulation. The model would have to be modified however if it was used outside of Oregon. The effect of changing input data was seen in the wide range of NSE values that resulted from testing different OTU prevalence values. If the model was applied in a different area the input data would likely vary from what was used to develop the model due to differences in bacteria and land cover. This means it may be necessary to search again for the values of parameters (kernel, C, and prevalence) that result in the best prediction using the new input data. The methods from this study could still be used to develop a model for a different location but the final model that resulted from this research may not be applicable to all locations. In different locations it is also important to consider the different land covers that may be indicators for bacterial DNA. It is possible that if this model was applied in a different location that it could predict a land cover characteristic other than forest and shrub cover most efficiently using bacterial DNA

4.2 Reasoning for Result

The resulting NSE value was not as high as expected and this can be explained by many factors. When examining the StreamStats data frame not all the land cover percentages summed to 100 for each site and some of them were over 100. This means that there is likely some overlap of the land cover categories or that there is some error in the StreamStats estimation. The forest and shrub cover values were very similar for most of the sites since the sampling was all done in Oregon. It is important to consider that this may have made it easier to predict than other land covers due to this similarity in values.

Another factor that was considered when examining the results was the time of year the sample was collected. It was expected that part of the contribution of bacteria from different land covers would be from runoff, but the amount of runoff depends on the time of year. It is also important to consider that there is likely a higher concentration of these bacteria in the runoff from the first major rain event due to build up over the dry season. The samples used in the data analysis were mainly spring samples and there is usually a lot of rainfall in spring in Oregon, but it would not be part of the initial flush.

4.3 Future Research

There are many opportunities for future research with this model. There are still many Sterivex filters that have not been extracted yet that include additional sampling sites from John Day and the sampling sites used for this study but sampled at different times of the year. This provides plenty of data with which this model can be tested. It would be interesting to see how the bacterial composition changes seasonally at some of the sampling sites. If data from a fall collection was used, which is likely around the time of the first heavy rainfall event, it would be interesting to see if the model could predict land cover characteristics such as agriculture more effectively. The bacterial DNA from John Day sampling could be used to diversify the forest and shrub cover data since the John Day area generally has less forest cover than the Willamette Basin. This would be a good test of the model to see if it would get better or worse at predicting forest and shrub cover with more diverse data.

5. Bibliography

Abu-Ashour, J., Joy, D. M., Lee, H., Whiteley, H. R., & Zelin, S. (1994). Transport of microorganisms through soil. *Water, Air, & Soil Pollution,75*(1-2), 141-158. doi:10.1007/bf01100406

BiteSizeBio. (2018, November 23). DNA Precipitation: Ethanol vs. Isopropanol. Retrieved May 16, 2019, from https://bitesizebio.com/2839/dna-precipitation-ethanol-vs-isopropanol/

Crump, B. (2007, October 30). DNA extraction from Sterivex filters. Retrieved May 15, 2019, from http://people.oregonstate.edu/~crumpb/Methods/Protocols/DNAExtractionSterivex20071101.pdf

Genlantis. (2017). Cell Lysis. Retrieved May 16, 2019, from http://www.genlantis.com/cell-lysis.html

Google Earth. (2019). Overview – Google Earth. Retrieved May 8, 2019, from https://www.google.com/earth/

Hermans, S. M., Buckley, H. L., Case, B. S., Curran-Cournane, F., Taylor, M., & Lear, G. (2016). Bacteria as emerging indicators of soil condition. *Applied and Environmental Microbiology*. doi:10.1128/aem.02826-16

McKiernan, H., & Danielson, P. (2017). Molecular Diagnostic Applications in Forensic Science. *Molecular Diagnostics,*371-394. doi:10.1016/b978-0-12-802971-8.00021-3

Scikit-learn. (2007 - 2019). Sklearn.svm.SVR. Retrieved May 20, 2019, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

Statistics How To. (2017, November 14). RMSE: Root Mean Square Error. Retrieved May 16, 2019, from https://www.statisticshowto.datasciencecentral.com/rmse/

UCMP. (2019). Bacteria: Life History and Ecology. Retrieved May 17, 2019, from https://ucmp.berkeley.edu/bacteria/bacterialh.html

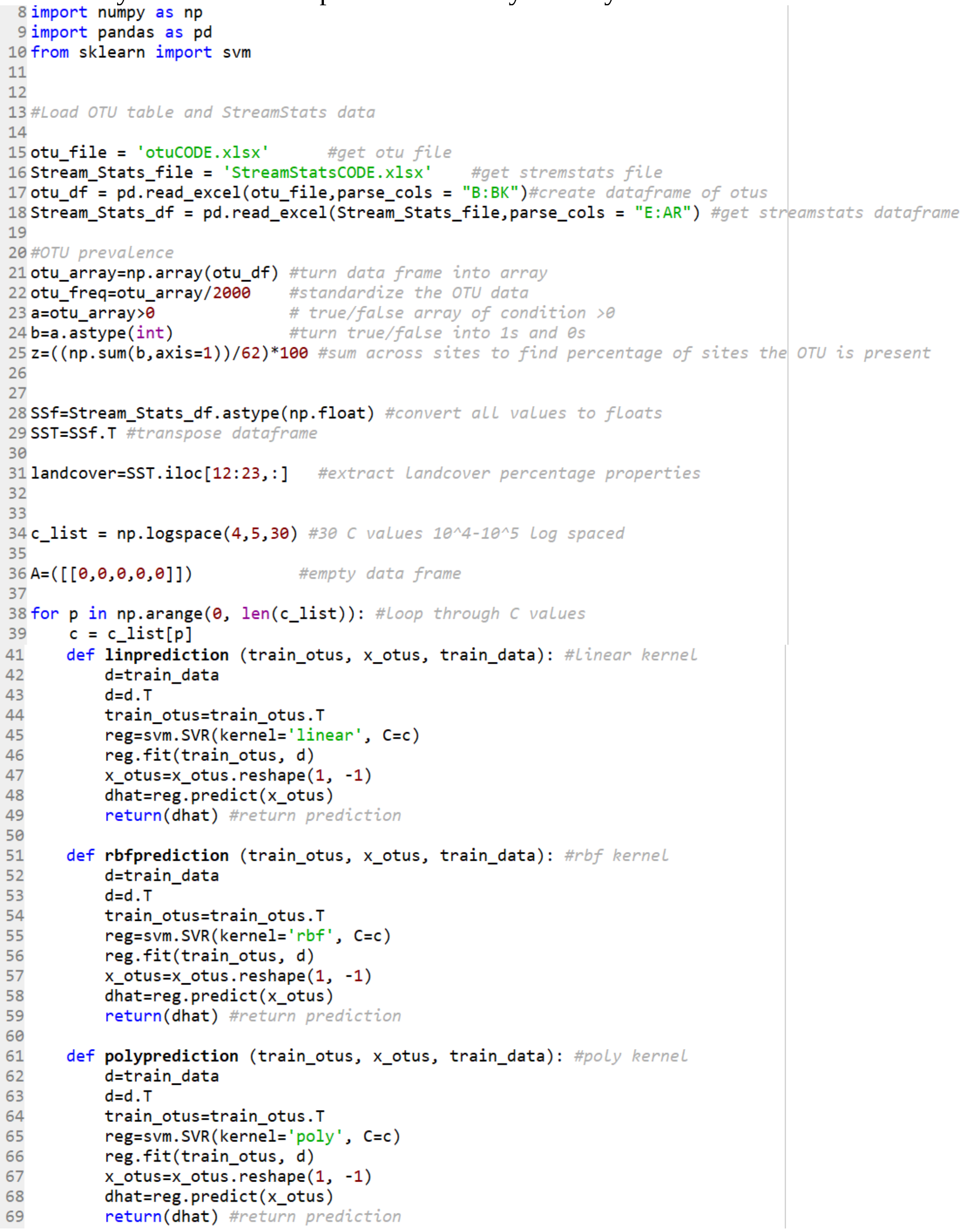
Urycki, D. R., Good, S. P., & Crump, B. C. (2019). Using Stream Bacterial DNA to Estimate Macroscale Catchment Function. *Oregon State University Poster*. Retrieved May 17, 2019.

USGS. (2019). StreamStats. Retrieved May 16, 2019, from https://streamstats.usgs.gov/ss/

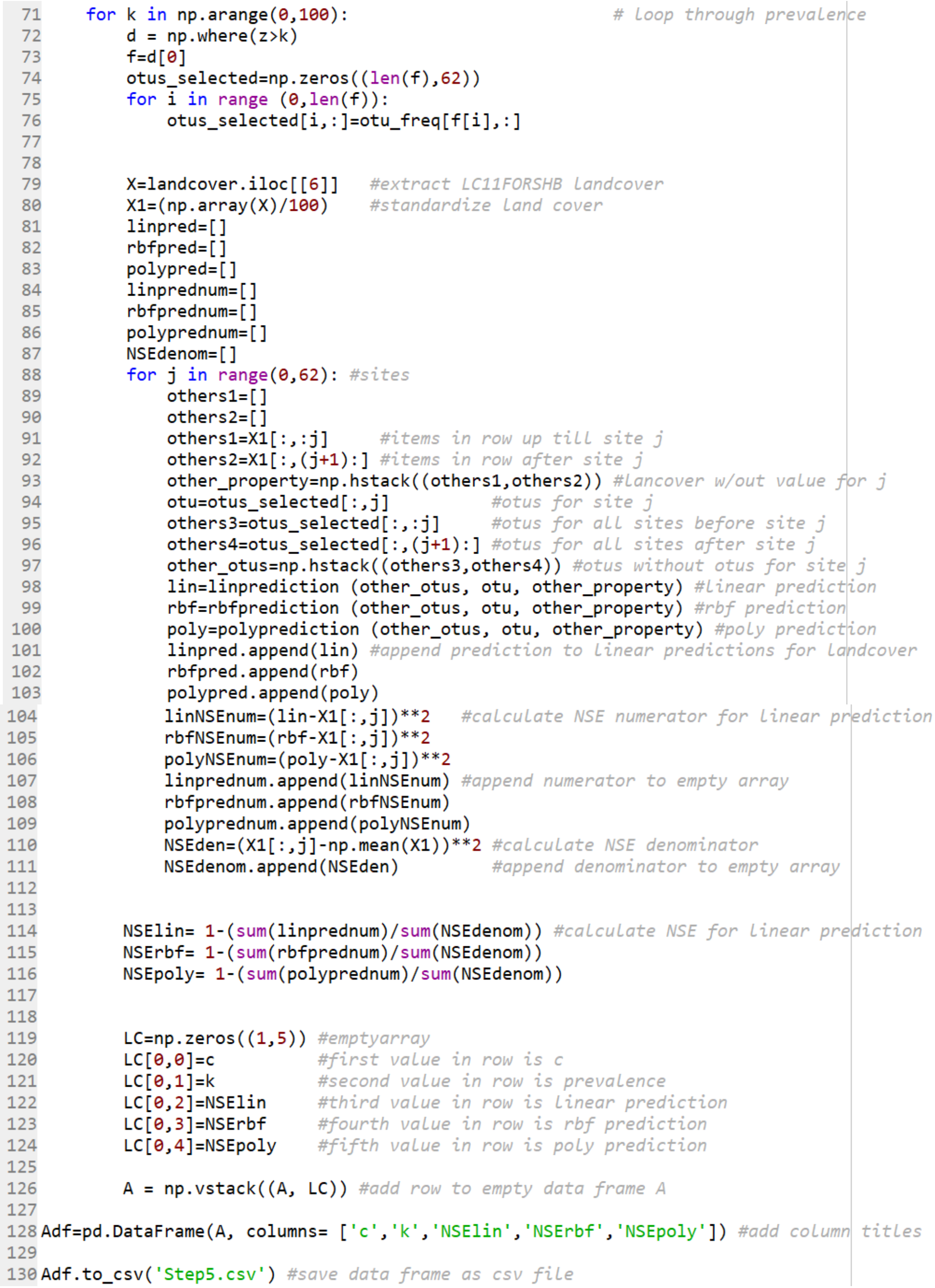
Zhang, H., Gao, Z., Shi, M., Fang, S., Xu, H., Cui, Y., & Liu, J. (2019). Study of the Effects of Land Use on Hydrochemistry and Soil Microbial Diversity. *Water,11*(3), 466. doi:10.3390/w11030466

6. Appendix

A.1: Python code for step 4 in SVR Analysis in Python methods section



```python
import numpy as np
import pandas as pd
from sklearn import svm


#Load OTU table and StreamStats data

otu_file = 'otuCODE.xlsx'      #get otu file
Stream_Stats_file = 'StreamStatsCODE.xlsx'    #get stremstats file
otu_df = pd.read_excel(otu_file,parse_cols = "B:BK")#create dataframe of otus
Stream_Stats_df = pd.read_excel(Stream_Stats_file,parse_cols = "E:AR") #get streamstats dataframe

#OTU prevalence
otu_array=np.array(otu_df) #turn data frame into array
otu_freq=otu_array/2000    #standardize the OTU data
a=otu_array>0              # true/false array of condition >0
b=a.astype(int)            #turn true/false into 1s and 0s
z=((np.sum(b,axis=1))/62)*100 #sum across sites to find percentage of sites the OTU is present


SSf=Stream_Stats_df.astype(np.float) #convert all values to floats
SST=SSf.T #transpose dataframe

landcover=SST.iloc[12:23,:]   #extract landcover percentage properties


c_list = np.logspace(4,5,30) #30 C values 10^4-10^5 log spaced

A=([[0,0,0,0,0]])           #empty data frame

for p in np.arange(0, len(c_list)): #loop through C values
    c = c_list[p]
    def linprediction (train_otus, x_otus, train_data): #linear kernel
        d=train_data
        d=d.T
        train_otus=train_otus.T
        reg=svm.SVR(kernel='linear', C=c)
        reg.fit(train_otus, d)
        x_otus=x_otus.reshape(1, -1)
        dhat=reg.predict(x_otus)
        return(dhat) #return prediction

    def rbfprediction (train_otus, x_otus, train_data): #rbf kernel
        d=train_data
        d=d.T
        train_otus=train_otus.T
        reg=svm.SVR(kernel='rbf', C=c)
        reg.fit(train_otus, d)
        x_otus=x_otus.reshape(1, -1)
        dhat=reg.predict(x_otus)
        return(dhat) #return prediction

    def polyprediction (train_otus, x_otus, train_data): #poly kernel
        d=train_data
        d=d.T
        train_otus=train_otus.T
        reg=svm.SVR(kernel='poly', C=c)
        reg.fit(train_otus, d)
        x_otus=x_otus.reshape(1, -1)
        dhat=reg.predict(x_otus)
        return(dhat) #return prediction
```



```python
    for k in np.arange(0,100):                          # loop through prevalence
        d = np.where(z>k)
        f=d[0]
        otus_selected=np.zeros((len(f),62))
        for i in range (0,len(f)):
            otus_selected[i,:]=otu_freq[f[i],:]


        X=landcover.iloc[[6]]   #extract LC11FORSHB landcover
        X1=(np.array(X)/100)    #standardize land cover
        linpred=[]
        rbfpred=[]
        polypred=[]
        linprednum=[]
        rbfprednum=[]
        polyprednum=[]
        NSEdenom=[]
        for j in range(0,62): #sites
            others1=[]
            others2=[]
            others1=X1[:,:j]     #items in row up till site j
            others2=X1[:,(j+1):] #items in row after site j
            other_property=np.hstack((others1,others2)) #lancover w/out value for j
            otu=otus_selected[:,j]          #otus for site j
            others3=otus_selected[:,:j]     #otus for all sites before site j
            others4=otus_selected[:,(j+1):] #otus for all sites after site j
            other_otus=np.hstack((others3,others4)) #otus without otus for site j
            lin=linprediction (other_otus, otu, other_property) #linear prediction
            rbf=rbfprediction (other_otus, otu, other_property) #rbf prediction
            poly=polyprediction (other_otus, otu, other_property) #poly prediction
            linpred.append(lin) #append prediction to linear predictions for landcover
            rbfpred.append(rbf)
            polypred.append(poly)
            linNSEnum=(lin-X1[:,j])**2   #calculate NSE numerator for linear prediction
            rbfNSEnum=(rbf-X1[:,j])**2
            polyNSEnum=(poly-X1[:,j])**2
            linprednum.append(linNSEnum) #append numerator to empty array
            rbfprednum.append(rbfNSEnum)
            polyprednum.append(polyNSEnum)
            NSEden=(X1[:,j]-np.mean(X1))**2 #calculate NSE denominator
            NSEdenom.append(NSEden)         #append denominator to empty array


        NSElin= 1-(sum(linprednum)/sum(NSEdenom)) #calculate NSE for linear prediction
        NSErbf= 1-(sum(rbfprednum)/sum(NSEdenom))
        NSEpoly= 1-(sum(polyprednum)/sum(NSEdenom))


        LC=np.zeros((1,5)) #emptyarray
        LC[0,0]=c          #first value in row is c
        LC[0,1]=k          #second value in row is prevalence
        LC[0,2]=NSElin     #third value in row is linear prediction
        LC[0,3]=NSErbf     #fourth value in row is rbf prediction
        LC[0,4]=NSEpoly    #fifth value in row is poly prediction

        A = np.vstack((A, LC)) #add row to empty data frame A

Adf=pd.DataFrame(A, columns= ['c','k','NSElin','NSErbf','NSEpoly']) #add column titles

Adf.to_csv('Step5.csv') #save data frame as csv file
```