SEQUENCE AND STRUCTURE BASED CLASSIFICATION AND PREDICTION OF ANTIMICROBIAL PEPTIDES

by

Krista Smith

A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Bioinformatics and Computational Biology

Committee:

IVaisman	Dr. Iosif Vaisman, Committee Chair
	Dr. Dmitri Klimov, Committee Member
Monique van Hoek	Dr. Monique Van Hoek, Committee Member
IVaisman	Dr. Iosif Vaisman, Director, School of Systems Biology
Donna M. Fox	Dr. Donna M. Fox, Associate Dean, Office of Student Affairs & Special Programs, College of Science
	Dr. Fernando R. Miralles-Wilhelm, Dean, College of Science

Date: November 18, 2021

Fall Semester 2021
George Mason University
Fairfax, VA

Sequence and Structure Based Classification and Prediction of Antimicrobial Peptides

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

by

Krista Smith
Masters of Biology
Wright State University, 2004
Bachelors of Science
Carnegie Mellon University, 1997

Director: Iosif Vaisman, Professor
Department of Computational Biology and Bioinformatics

Summer Semester 2021
George Mason University
Fairfax, VA

Copyright 2021 Krista Smith
All Rights Reserved

DEDICATION

This is dedicated to my three sons, Ethan, Ryan, and Zachary and to my wonderful husband Brad.

ACKNOWLEDGEMENTS

I would first like to thank the staff at the George Mason University School of System's Biology, Chris Ryan, Kimberly Harris, Monique Sweeney and notably, Diane St. Germain, who convinced me to start this whole crazy journey in the first place. I would like to offer my thanks to my thesis committee members, Dr. Dmitri Klimov and Dr. Monique Van Hoek for their assistance and insightful critique. Dr. Iosif Vaisman, my advisor, was an invaluable resource and source of support. I'd also like to thank my fellow graduate students, including but not limited to Andy Hoang, Fayaz Seifuddin, Jasmine Amirzadegan, and Shengyuan Wang without whom I would have never passed my comprehensive exams. Finally, I'd like to offer my most heartfelt thanks to my children, Ethan, Ryan, and Zachary, and my husband Brad. Their tireless support and encouragement have made this work possible

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

Amino Acid Composition model ..... AAC
Antimicrobial peptides ..... AMP
Artificial Neural Fuzzy-Interface-Systems ..... ANFIS
Artificial Neural Networks ..... ANN
Basic Local Alignment Search Tool ..... BLAST
Blocks substitution matrix ..... BLOSUM
Critical Assessment of protein Structure Prediction ..... CASP
Define Secondary Structure of Proteins ..... DSSP
False Negative ..... FN
False Positive ..... FP
Fuzzy K-Nearest Neighbor ..... FKNN
Gradient Boosted Classifier ..... GBC
Hidden Markov Models ..... HMM
Linear Discriminant Analysis ..... LDA
Matthew's Correlation Coefficient ..... MCC
Monte Carlo ..... MC
Miyazawa and Jernigan matrix ..... MJ matrix
Pseudo-Amino Acid Composition ..... PseAAC
Qualitative matrices ..... QM
Quantitative structure-activity relationship ..... QSAR
Random Forest ..... RF
Receiver operating characteristic ..... ROC
Support Vector Machines ..... SVM
True Negative ..... TN
True Positive ..... TP

ABSTRACT

SEQUENCE AND STRUCTURE BASED CLASSIFICATION AND PREDICTION OF ANTIMICROBIAL PEPTIDES

Krista Smith, PhD

George Mason University, 2021

Dissertation Director: Dr. Iosif Vaisman

In recent years pan-resistant microbes have begun to pose a significant risk, particularly in clinical settings. To combat this emerging threat new antimicrobial therapies are required. Antimicrobial peptides (AMPs) are a promising, and until recently, mostly underutilized resource. A large number of AMPs have been experimentally identified and predicted, very few of them are approved for clinical use, but thousands more may be hiding in plain sight in various databases. Machine learning offers a powerful technique to mine already available protein sequences for those with high potential to exhibit antimicrobial properties. This work is focused on creating and testing a novel set of descriptors based on reduced amino acid residue alphabets, structural, and topological properties of AMPs. These novel descriptors were used in the

machine learning models capable of discriminating AMPs from non-AMPs. Such models may be used to screen proteins with known structures for potential antimicrobial activity.

INTRODUCTION

The Significance of Antimicrobial Resistance

Current antibiotics treatments are quickly becoming less effective as the microbes that they are designed to combat acquire resistance. As the usefulness of these first-line, and eventually second- and third-line, antibiotics is reduced, clinicians are forced to use less effective, more toxic, and more costly therapies. In the United States alone, more than 23,000 people die each year as a direct result of antibiotic resistant infections, with many more fatalities attributed to complications of these infections. Antibiotic resistant microbes lead to longer, and more expensive hospital stays, additional doctor visits and lost productivity. While estimates of the economic cost of antibiotic resistance are difficult to calculate direct medical costs to individual patients range from more than $18,000 to more than $29,000, totaling about $20 billion per year in the United States alone. Hospital stays for patients with antibiotic resistant infections are increased by 6 to 12 days resulting in over $35 billion in lost wages to US households every year.[1]

The situation is even more dire in low- and middle-income countries. While high-income countries have the advantages of sanitation and improved nutrition to reduce the effects of infectious disease, developing countries suffer from the burden of poor public health facilities coupled with the reduced effectiveness of antibiotics used to mitigate these deficiencies. Of particular interest are resistant pathogens associated with neonatal

infection. Data suggests that 71% of *Klebsilla* and 50% of *E.coli* are resistant to the WHO recommended regimen of ampicillin and gentamicin.[2] Unlike in first-world countries, expensive secondary and tertiary treatments are often unavailable in developing countries resulting in increasing morbidity and mortality.[3]

Without intervention, the antibiotic crisis will only worsen. Microbes are the most adaptable organisms on earth. Over the past 3.5 billion years they have evolved to inhabit every environment on the planet, from sub-zero arctic frost to the boiling depths of the Atlantic ocean's thermal vents. It should, therefore, be no surprise that microbes have evolved to survive the killing effects antibiotics. Bacteria are masters of adaptation allowing them to quickly develop resistance to antimicrobials through selective pressure. Conjugation allows resistant bacteria to pass resistance genes contained on plasmids to other bacteria that may have never even been exposed to the antimicrobial agent in question.

Coupled with bacteria's impressive adaptive abilities, antibiotics have become a victim of their own success. Further accelerating resistance acquisition, is the non-judicious manner in which antibiotics have historically been overused and misused. In many countries, antibiotics are available over the counter with little to no medical guidance. Non-prescription antimicrobial use is more highly correlated with shorter courses, as well as inappropriate drug and dosing choices. This inappropriate use of antimicrobials has been associated with high levels of community antimicrobial resistance. [4]

Given a microbe's propensity for adaptation, the question of resistance to any given anti-infective is not a matter of if but when. It is for this reason that there is constant pressure for the development of new antimicrobials. As of September 2019, there were 44 prospective antimicrobials with the potential to treat serious microbial infection under clinical trials in the US.[4] Historically only 1 to 5 percent of these compounds will be approved for clinical use. Furthermore, most of these are modifications to one of the already common structural classes of antibiotics. While these modifications may induce new activity, they are not a significant shift in the underlying mechanism, meaning that they are still susceptible to resistance. Economic and market pressures have also contributed to the lagging development of new clinical antimicrobial solutions. Antimicrobial drugs are not profitable. They are commonly taken for only short courses and sold for low prices when compared to drugs such as those used to treat high cholesterol or cancer. Additionally, resistance to these drugs begins to develop as soon as they are available for clinical use. If a new drug with significant anti-infective potential is developed, it is stored away for only the direst of circumstances, resulting in even less profit for the drug company holding the patent. For these reasons, the rate at which new antibacterial entities are approved by the FDA has dropped precipitously since 1990. Between 1999 and 2008 only 17 new antimicrobials were approved while 34 were removed from the market.[5]

In addition to the threat to global heath, antibiotic resistant pathogens have been identified as a potential biological weapon. A plasmid-mediated multidrug resistant strain of *Yersinia pestis*, as well as a streptomycin resistant strain have been independently

isolated in Madagascar.[6,7] There have been assertions that scientists in the former USSR were working on developing weaponized multi-drug resistant strains of *Y. pestis.*[8] The CDC has classified *Y. pestis* as a category A critical biological agent due to its high rate of infectivity as well as mortality coupled with the ease with which it may be disseminated. Also cited as a potential bioterrorism threat is multi-drug resistant tuberculosis. The potential for these agents to cause widespread loss of life and public panic have lead the CDC to recommend preparedness through ongoing research to improve disease treatment among other interventions.[9]

Antimicrobial Peptides

Antimicrobial peptides (AMP's) are oligopeptides ranging from 5 to over 100 amino acids. They were first discovered in 1939 with the extraction of what would later be identified as an AMP which was named gramicidin.[10,11] Gramicidin was derived from a *Bacillus* strain and showed activity in protecting mice from pneumococci infection and later used during World War II to treat wounds and ulcers.[12] There are currently more than 2500 known amps cataloged in web based AMP databases such as APD3,[13] YADAMP,[14] and CAMPR3[15] with activities ranging from antifungal to insecticidal to anti-HIV. Antimicrobial peptides form a major component of the innate immune system and are found across every phyla of life. They can vary widely in structure, target specificity and mode of attack, making them difficult to characterize.

The role of computation in AMP classification

The process of testing peptides for antimicrobial properties in the lab is lengthy and expensive. Through the use of machine learning, candidates with higher potential for

activity can be identified. Similarly, features that are strongly correlated with antimicrobial properties can be identified resulting in more accurate hypothesis regarding the design of AMPs. Computational methods can be used to identify the sequential and structural components most strongly associated with antimicrobial activity. While evidence suggests that AMPs do not assume their final structural conformation until they are in contact with the cellular membrane, there has been little research utilizing the secondary and tertiary structures of peptides for prediction.

Machine learning

Machine learning is a field of computer science that utilizes statistical techniques to simulate the ability to learn from data without being specifically programmed. Given a well annotated dataset the machine is allowed to find patterns to make decisions. These decisions are compared to the ground truth and, using statistical and mathematical techniques, the machine modulates the decision-making process in order to make better decisions. Machine learning has been used in a number of applications within the field of bioinformatics including the prediction of protein-protein interactions,[16] classification of protein function recognition,[17] and prediction of cancer progress and prognosis.[18]

Problem Statement and Objectives

Given the current state of antimicrobial resistance it is imperative that new forms of antibiotics be identified. Current processes are inadequate to screen the vast numbers of peptides already isolated and do not provide adequate predictive guidance. Furthermore, AMP databases do not incorporate an effective mechanism for the encoding of secondary and tertiary structures usable for machine learning. This dissertation addresses these

issues by first constructing a methodology for expressing three-dimensional peptide structures numerically, as well as producing a dataset composed of known AMPs coupled with negative peptide sequences. Additionally, a computational model for the prediction of antimicrobial activity will be presented. This dissertation offers the following contributions to the field:

1. **Description of Antimicrobial peptides:** A description of antimicrobial peptides including their basic biological and biochemical principles is presented in chapter 2.
2. **Role of computation in peptide prediction:** Background information regarding machine learning and its role in AMP classification is presented in chapter 3
3. **Methodology for the encoding of tertiary structures:** Data encoding methodologies, presented in chapter 4 describe a mechanism for representing peptide tertiary structures in a manner accessible for machine learning.
4. **AMP classification:** predictive models described in chapter 5 offer a method for screening peptides for antimicrobial potential.
5. **Antimicrobial Feature identification:** Utilizing feature reduction techniques, the predictive model can be used to identify the peptide features most important in discriminating AMPs from non-AMP peptides.

CHAPTER TWO: RELATED WORK

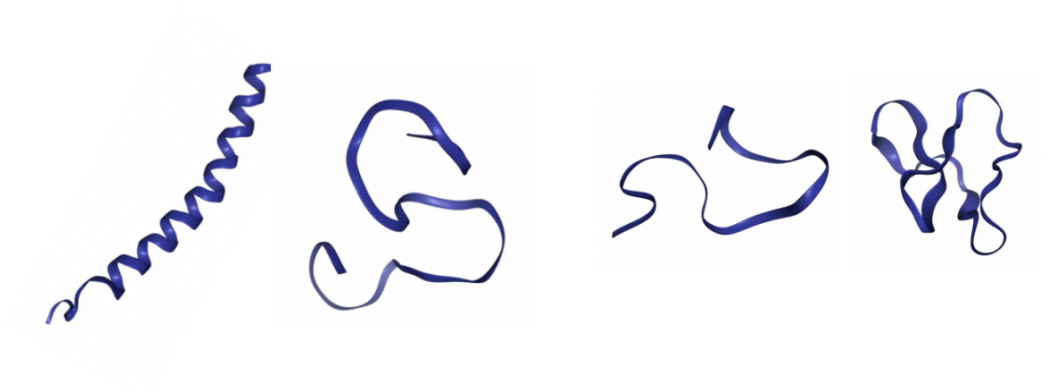


Figure 1 - Typical examples of AMPs based on structural classifications.

Antimicrobial Peptides

Antimicrobial peptides form a major component of the innate immune system and are found across every phyla of life. They can vary widely in structure, target specificity and mode of attack, making them difficult to characterize. Most recently, they have been classified into four categories based on peptide chain connection patterns. Linear Class I (UCLL) consists of peptides lacking any side chain interactions such as human cathlicidin LL-37. Class II, side chain linked peptides (UCSS) is composed of those peptides with sidechain to sidechain interaction, such as the bacterial lantibiotic actagardine. Peptides with sidechain to backbone interactions are placed in class III

(UCSB), with daptomycin being a typical example. Finally, peptides in Class IV (UCBB) are characterized by backbone to backbone interactions such as the plant AMP kalata B [Figure 1].

Antimicrobial resistance

Antimicrobial peptides have coevolved alongside bacteria for millions of years and, as a result have developed microbicidal mechanisms more robust against acquired resistance than their pharmaceutical counterparts.[19,20] Vancomycin, for instance, inhibits peptidoglycan synthesis by binding to the D-alanine-D-alanine terminal residues of the peptidoglycan pentapeptide linker. This action inhibits cross-linking and lowers the strength of the bacterial cell wall resulting in cellular death. However, if the final alanine is substituted with a lactase the affinity of vancomycin for the pentapeptide is reduced 1000-fold rendering the microbe vancomycin resistant.[21] Many AMP's also disrupt the cellular membrane, but in contrast to Vancomycin, AMP's are believed to permeabilize the membrane through electrostatic interactions with the anionic lipids commonly found in prokaryotic, but not eukaryotic cells.[11] When compared to a single amino acid substitution, a complete overhaul of the cellular membrane chemistry is a daunting evolutionary undertaking.

Mechanisms of Action

AMPS exhibit both receptor mediated and non-receptor mediated mechanisms of action. The most common non-receptor mediated mechanism of action involves the binding of AMPs to the cellular membrane. In general terms, the peptides bind to the surface of the cell until a critical threshold has been reached, at which time they self-organize to

permeate the membrane. Three primary mechanisms for this permeation have been proposed. The carpet model, exhibited by cecropins[22] and aurein,[23] according to this model, peptide monomers accumulate on the membrane surface eventually reaching a concentration sufficient to destabilize the phospholipid packing of the membrane leading to its eventual disintegration. The Barrel and stave model, as in the case of pardaxin and its analogues[24] and alamethicin[25] this model postulates that the peptides organize across the cellular membrane to form a permeation pathway which allows ions to leak from the cytosol leading to the disruption of bacterial metabolism. Finally, the toroidal pore model, exhibited by and magainins[26] and metellins[27] in which the peptide and lipid head groups of the target membrane align together to form a pore [Figure 2].

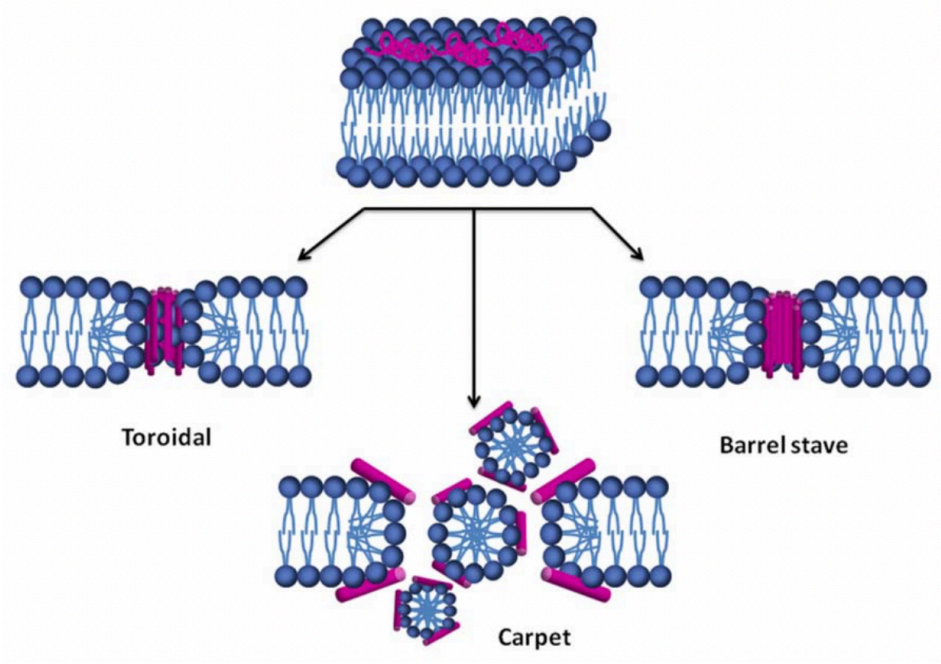


Figure 2 - Non-receptor mediated modes of action. Adapted from 28

While membrane interactions are the most notable mechanism of antimicrobial activity, AMPS also employ a number of alternative mechanisms including traversing the cell membrane to interact with intracellular targets. Pleurocidin, an AMP derived from winter flounder, has demonstrated the ability to inhibit intercellular process such as macromolecule and RNA synthesis without damaging the cytoplasmic membrane at sub-lethal concentrations.[29] Anionic peptides found in the mucous and serous respiratory secretions, as well as cationic, amphipathic AMPs are thought to induce intracellular biomass flocculation as the mechanism of bacterial killing.[30,31] Other AMPs such as buforinII,[32] MicrocinB17 [33] and certain indolicidin analogs[34] have been shown to bind to DNA or RNA, inhibiting protein synthesis. These variations in microbial killing mechanism demonstrate the versatility of AMPs.

AMP Target Selection

The majority of AMP's target their microbial victims via a non-receptor mediated mechanism. Unlike traditional antibiotics these AMPs do not target a specific receptor, but instead the more general target of the bacterial membrane. Both gram negative and gram positive bacteria exhibit a net negative charge on their outer surfaces due to the respective presence of lipopolysaccharides and acidic polysaccharides. Additionally, both types of bacteria possess negatively charged inner membranes due to the presence of negatively charged phospholipids. This contrasts directly to mammalian cells which exhibit zwitterionic phosphatydilcholine on the outer leaflet and negatively charged phosphatydilserine on the inner leaflet.[35] This facilitates the preferential binding of largely cationic AMPs to microbial membranes over mammalian membranes. This theory

has been studied through the creation of enantiomers of AMPs such as melittin, cecropin, magainin and androctonin, which possess identical lytic behavior to their all L-amino acid counterparts.[36–38] These studies lead to the conclusion that the chirality of the peptide is not a critical feature. However, replacement of a single L-amio acid with its D enantiomer in melittin results in a change in the amphipathic nature of the peptide and an associated loss of antimicrobial activity[39] leading to the conclusion that the amphipathic structure is critical the AMPs antimicrobial activity.

CHAPTER 3: COMPUTATION IN PEPTIDE PREDICTION, CLASSIFICATION AND DESIGN

Protein Structure Prediction

A great deal of research has been done to develop and refine methods of peptide prediction in general and antimicrobial peptide classification specifically. Beginning in the late 1980's neural networks were applied to the problem of secondary structure prediction.[41] A feed forward network was trained using existing protein structures to predict the secondary structure of a local sequence of amino acids. This method achieved an accuracy of between 60 and 70%, a significant improvement over contemporary methods. Rost and Sandler took advantage of the conserved nature of protein secondary structures across protein homologs by training a neural network using multiple sequence alignments. The resulting accuracy was improved to between 70-74%.[40]

Since 1994 the Protein Structure Prediction Center has hosted the biannual Critical Assessment of protein Structure Prediction, or CASP, experiment to allow researchers to objectively measure the success of their computational algorithms for protein structure prediction. Prediction algorithms have progressively improved, in the first CASP competition only about 15% of the most difficult protein structures were predicted accurately, compared with nearly 60% in a recent CASP12(2016) competition.[41] Many of these improvements can be attributed to more powerful computational models utilizing machine learning.[42,43]

Protein Encoding

Previous research to classify peptides as AMP or non-AMP has relied primarily upon machine learning algorithms in which the function label is coupled with an encoding of the peptide structure. The issue of encoding a variable length peptide sequence with complex amino acid interactions into a fixed-length numerical vector is not a trivial endeavor. While the use of the complete amino acid sequence seems to be the most intuitive method of peptide encoding this method is often unsuitable to machine learning as peptides are of varying lengths and most machine learning algorithms require fixed length inputs. In order to fully harness the power of machine learning, discrete feature vector models for peptide encoding must be developed.

The simplest discrete model for peptide encoding is the amino acid composition model (AAC) [44] in which each peptide is represented by a vector **V**.

$$\boldsymbol{V} = [f_1, f_2, f_3, \ldots f_{20}] \quad [1]$$

In which f_i is the normalized frequency of each of the 20 naturally occurring amino acids in the peptide. Many methods of peptide function prediction have been based on the amino acid composition model [45,46] including several for the prediction of AMPs.[45–47] However, this model does not preserve the sequential nature of the peptide resulting in limited predictive power. Several concepts have been proposed to address this matter. Chos's pseudo-amino acid composition (PseAAC)[48] incorporates a set of discrete sequence correlation factors with the 20 values for amino acid composition to partially preserve the effects of sequence order while organizing the data in a manner amenable to

computational analysis. PseAAC has been used extensively for protein prediction in general as well as AMP prediction specifically.[47,49–51]

In order to capture the physiochemical nature of the peptide alternative methods have incorporated a variety of physiochemical features such as hydrophobicity,[50–52] dissociation constants,[50] isoelectric point,[50] molecular weight,[50] polarity,[51,53] secondary structure predictors,[51,53] molecular volume,[53] codon diversity,[53] solvent accessibility,[51] normalized van der Waals volume,[51] electrostatic charge,[51,53] and propensity for aggregation.[52]

N-gram encoding

Given an alphabet A, and sequence S, an n-gram is any n-long subsequence of consecutive tokens of A. For any sequence S of length *N*, there are *N*-(n-2) n-grams. More simply, an n-gram is an n item long portion of a longer sequence. The items may be words in a sentence, letters in a word or, as in the case of peptide analysis, amino acids or structural classifications in a peptide sequence. N-grams are widely used in a wide range of disciplines including communication theory, data compression and computational biology. The concept of n-grams can be attributed to Claude Shannon's work on information theory. He proposed that, given a training set, one can derive a probability distribution for the next item in a sequence of n items. N-gram probability analysis has been used in a variety of natural language processing applications such as text classification,[54] authorship attribution,[55] and sentiment analysis.[56] In the context of a peptide, an n-gram is a contiguous sequence of n-amino acids. N-grams have been successfully employed in the field of bioinformatics for protein classification,[47,57]

clustering of genome sequences,[58] as well as prediction of antimicrobial peptides.[59,60] N-grams comprised on 3 symbols, or trigrams, have demonstrated success in the prediction of antimicrobial peptides in the past[60]. By comparing the probability profile of an unknown peptide to those developed from a positive and negative training set we can assess the peptide's potential for antimicrobial activity. The use of n-grams for protein analysis allows for the development of discrete factors to represent the amino acid composition of the peptide while maintaining some of the integrity of the original sequence.

Alphabet reduction

Given an alphabet of 20 amino acids there are 20^3, or 8000, possible n-grams that can be derived from any given peptide. This phenomenon, termed exponential explosion or combinatorial explosion, results in a highly sparse dataset in which many of the values are zero. In order to minimize the effects of exponential explosion a method of alphabet reduction has been employed. This technique utilizes a set of predefined alphabet reduction schemes based upon characteristics of the amino acids to group each amino acid into one of three groups. In this way the number of potential amino acid combination to can be reduced to 3^3, or 27, simultaneously reducing the amount of sparsity within the dataset. It has been found that alphabet reduction schemes using structural similarities among amino acids is a viable approach to analyze peptide structures while reducing the limitations resulting from limit data sets.[61]

Machine Learning Algorithms

Binary classification

There are a number of machine learning algorithms applicable to the prediction of peptide activity. Commonly these methods represent a binary classification effort in which a molecule is classified as either having antimicrobial activity or not.[23,49,50,53,62–66] The algorithm is first trained on a dataset containing examples from each class. This allows the machine to develop a method for pattern recognition that distinguishes between the two classes. It is at this time that a comprehensive dataset is required. The goal of machine learning is to recognize patterns that generalize to future, unseen, data. If the data presented for training is overly specific, or does not contain sufficient levels of noise, the classifier will suffer from overfitting. Overfitting results in the ability of a classifier to perform well on test data that is highly similar to the training data, but to perform poorly on unseen data that may not conform to the specifications of the training data. Conversely, underfitting may occur if the sample size is insufficient to represent the actual distribution of data or an incorrect training model is used, a linear model is used to represent polynomial data, for example.

Algorithms previously used for the binary classification of AMPs include Artificial Neural Networks (ANN), Artificial Neural Fuzzy-Interface-Systems (ANFIS), Linear Discriminant Analysis (LDA), Qualitative matrices (QM), Fuzzy K-Nearest Neighbor (FKNN), Hidden Markov Models (HMM), Random Forest (RF), and Support Vector Machines (SVM).

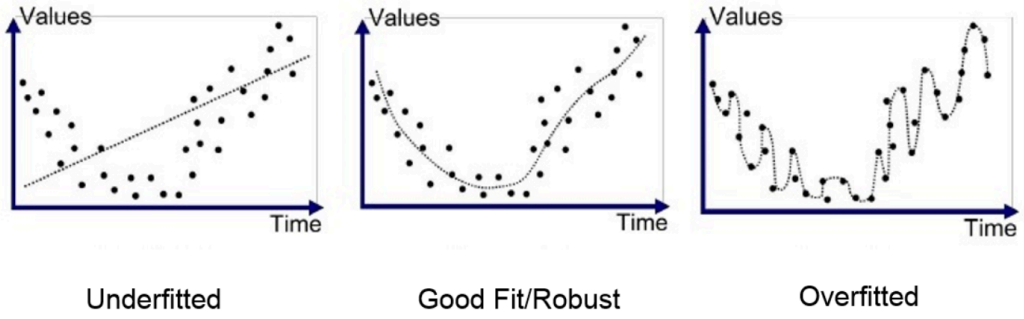


Figure 3 - Overfitting vs underfitting

Artificial neural networks

Artificial neural networks (ANN) are a type of machine learning framework loosely based on biological neurological neural networks. A network of computational neurons, also known as perceptions, learn to classify objects by training on a large set of annotated examples. The network is made up of several layers of fully connected perceptions. By presenting the network with a large number of both positive and negative examples the model can learn to perform a task without being explicitly programmed through the learned recognition of discriminatory features. A number of previous studies have attempted to predict active antimicrobial peptides using quantitative structure-activity relationship features (QSAR) with ANN. [66–70]

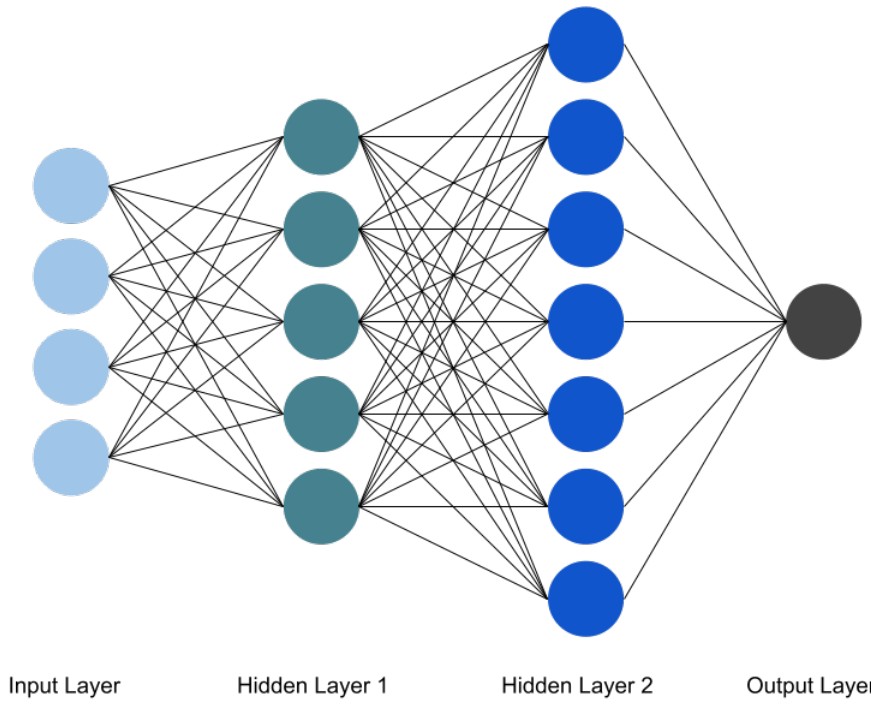


Figure 4 - Representative Artificial Neural Network

Artificial Neural Fuzzy-Interface-Systems

Artificial Neural Fuzzy-Interface-Systems (ANFIS) are a sort of artificial neural network which integrates both neural networks and fuzzy logic. ANFIS maps outputs to inputs using a set of fuzzy if-then rules coupled with a supervised feed-forward neural network. A typical ANFIS is based on five connected network layers. The first layer consists of input variables with a membership function to map each point on the input

space to a membership value. The second layer is a membership layer checks the weights of each membership function. The output of layer 3 represents the rule layer in which each node performs the pre-condition matching of the fuzzy rules. Layer 4 is the defuzzification layer that provides the output values resulting from the inference of the rules. Finally, the fifth layer aggregates all of the values from the previous fuzzy layer into single predictive value. An ANFIS created using two trapezoidal membership functions and trained on 10 epochs was used with a dataset consisting of aggregation and physiochemical distinguish between positive and negative AMPS.[71]

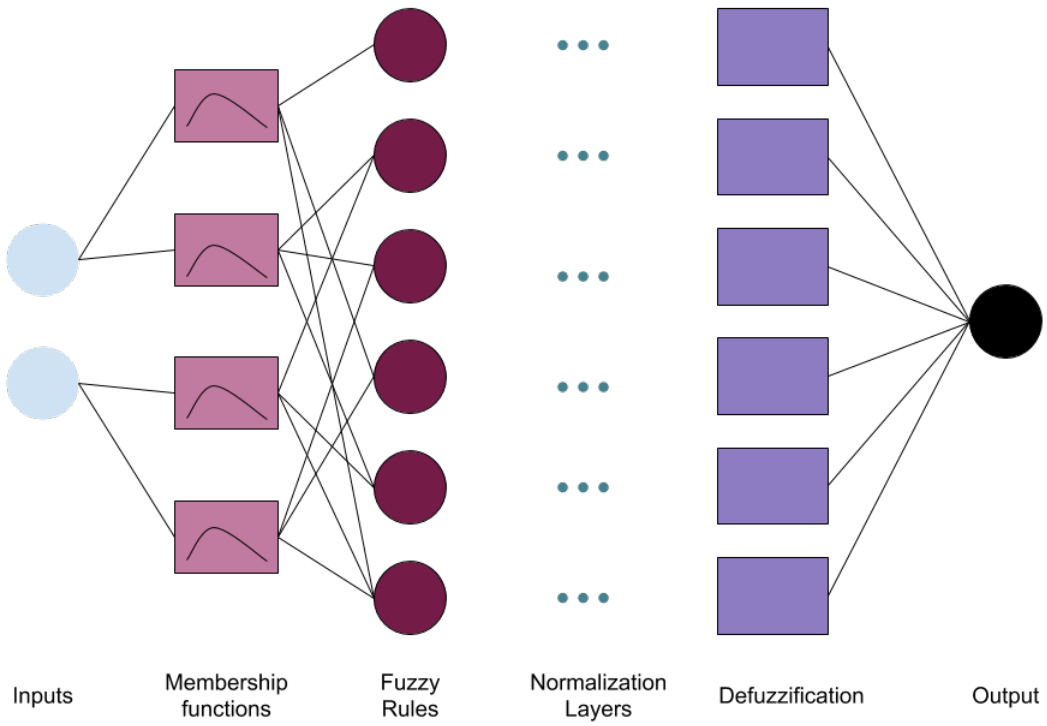


Figure 5 - Representative ANFIS function

Linear Discriminant Analysis

Discriminant analysis is a classification algorithm that attempts to predict the group membership of an independent variable based on a linear transformation of a set of independent variables. Given an input vector of x the algorithm attempts to define a function

$$y = f(\underset{w}{\rightarrow} \cdot \underset{x}{\rightarrow}) = f(\sum_{j} w_j \cdot x_j) \quad [2]$$

Where w is a vector of weights that is learned from the training set. Linear discriminant analysis is generally faster than other types of classifiers and works well with datasets of high dimensionality. Previous research has made use of LDA along with physiochemical and peptide composition features.[51]

Qualitative Matrices

A qualitative matrix is a measure of the propensity for each residue at a particular position within the peptide. The following equation is used to generate the qualitative matrix

$$P_{(i,r)} = \frac{E_{i,r}}{N_{i,r}} \quad [3]$$

Where $P_{(i,r)}$ is the probability of reside i at position r, $E_{i,r}$ is the number of residue r at position i and $N_{i,r}$ is the number of peptides. Lata 2007 developed matrices for the N- and

C- terminal amino acids of both antibacterial and non-antibacterial peptides. A weight matrix describing the difference between probabilities for each residue at each position for antimicrobial peptides compared with non-antimicrobial peptides was developed using the following equation:

$$Q_{(i,r)} = P_{AMP} - P_{Non-AMP} \quad [4]$$

By developing a score for each peptide using the formula:

$$score = \sum_{i=1}^{L} Q_{(i,r)} \quad [5]$$

Where L equals the length of the peptide, they were able to predict the class of the peptide with an MCC of 0.74 when considering the 15 amino acids of the N and C terminus.[64]

Fuzzy K-Nearest Neighbor

FKNN is an implementation of the common K-nearest neighbor algorithm with the addition of a fuzzy coefficient to determine the weight of each nearest neighbor's contribution to the membership value. A standard k-nearest neighbor algorithm clusters instances into groups of k instances with the goal of minimizing the intra-group distribution while maximizing the inter-group distribution. Distances between instances may be defined in a number of ways, but Euclidian distance is the most common. KNN assigns the same level of importance to each neighbor, assuming that the boundaries between classes are perfectly defined by the training set, which is often not the case. The

FKNN algorithm incorporates a fuzzy logic membership function designed to weight the computed distance between instances where the probability of membership in any given class is given by the following equation:

$$\mu_i(P) = \frac{\sum_{j=1}^{K} \mu_i\left(P_j^*\right) d\left(P, P_j^*\right)^{-\frac{2}{\varphi-1}}}{\sum_{j=1}^{K} d\left(P, P_j^*\right)^{-\frac{2}{\varphi-1}}} \qquad [6]$$

Where $\mu_i\left(P_j^*\right)$ is the fuzzy membership value of training sample P_j^* to the i-th class, $\mathrm{d}\left(P, P_j^*\right)$ is the distance between P and peptide P_j^*. Both K and φ are tunable parameters defining the number of neighbors to consider for each query peptide and the degree to which to weight the distances calculated for each nearest neighbor. The function to calculate $\mu_i\left(P_j^*\right)$ is dependent upon the desired classification outcomes. The addition of this fuzzy logic allows imprecise knowledge to be incorporated into the algorithm and results in higher classification success in many applications including protein identification and prediction of AMPs.

The iAMP-2L classification method makes use of a two-tiered FKNN algorithm in which peptides are first classified as either AMP or non-AMP using where $\mu_i\left(P_j^*\right)$ is defined as:

$$\mu_i\left(P_j^*\right) = \begin{cases} 1, & if\ P_j^* \in C_i \\ 0, & otherwise \end{cases} \qquad [7]$$

With the final class for peptide P being assigned to the class with the highest membership value:

$$C_u = \boldsymbol{argmax_i}\{\mu_i(P)\} \quad [8]$$

The second tier of the iAMP-2L algorithm attempts to predict the peptides functional group. For this tier [5] is replaced by a multilabel classifier:

$$\mu_i\left(P_j^*\right) = \begin{cases} \frac{1}{n(hit)}, & if\ P_j^*\ \in\ C_i \\ 0, & otherwise \end{cases} \quad [9]$$

Where n(hit) is defined as the number of different classes that were hit by P_j^* during the predication phase. This method resulted in an MCC of 0.73 for the prediction of AMP vs non-AMP.[50]

Hidden Markov Models

HMMs assume that the process being modeled is a Markov process in which the transition states are hidden. A Markov process is a stochastic, memoryless process in which the probability of subsequent events depends only on the state obtained by the previous event. In a hidden markov model the intermediate states between the input and the output are unknow to the user. HMMs can be represented as simple dynamic Bayesian models in which adjacent variables are related to each other by some probability.

HMMs have shown great promise in peptide prediction. HMM models for peptides are probabilistic models of amino acid sequences for a particular peptide family. Fjell et al developed HMM models for clusters of AMPs that were subsequently used to scan Swiss-Prot for additional sequences to add to each cluster. After the addition of a new sequence to any cluster an updated HMM model was constructed. In this manner, they were able to iteratively add 229 peptides to their AMP database, 195 of which contained annotations demonstrating antimicrobial activity.[72]

Random Forest

Random Forest (RF) is an ensemble learning method in which a multitude of decision tress are built with the mode of all trees used as the final output. A single decision tree is a decision support tool that uses a tree-like model to break complex problems into smaller parts based on a given query. Each node divides the training set based on a single feature. These features may be nominal or categorical. A simple decision tree describing an algorithm to classify the well known Iris dataset is presented in Figure 6.

In 2001 Brieman introduced the Random forest algorithm which applies the concept of bootstrap aggregation by training on a random sample with replacement of the full training set. Once training is complete predictions are made on unseen samples by taking the mode of the predictions of all trees. As the number of trees in the RF becomes larger the generalization error converges at some limit defined by the strength of the individual trees and the correlation between them.[73] In addition to sample bagging RF also utilizes feature sub-selection, using only a sub-set of features to construct each tree

and then assessing the accuracy on unseen data. This allows for error estimation, known as out of bag error without the need for cross validation or a validation set.[74]

RF algorithms can also be used for regression analysis where the tree predictor takes on a numerical value as opposed to a class prediction. This application is, however, not applicable to this research. RF algorithms have been used successfully to classify peptides antimicrobial activity based on the distribution patterns of amino acid properties along the sequence[75] as well as their basic physiochemical properties[51] as well as combinations of additional features.[76]

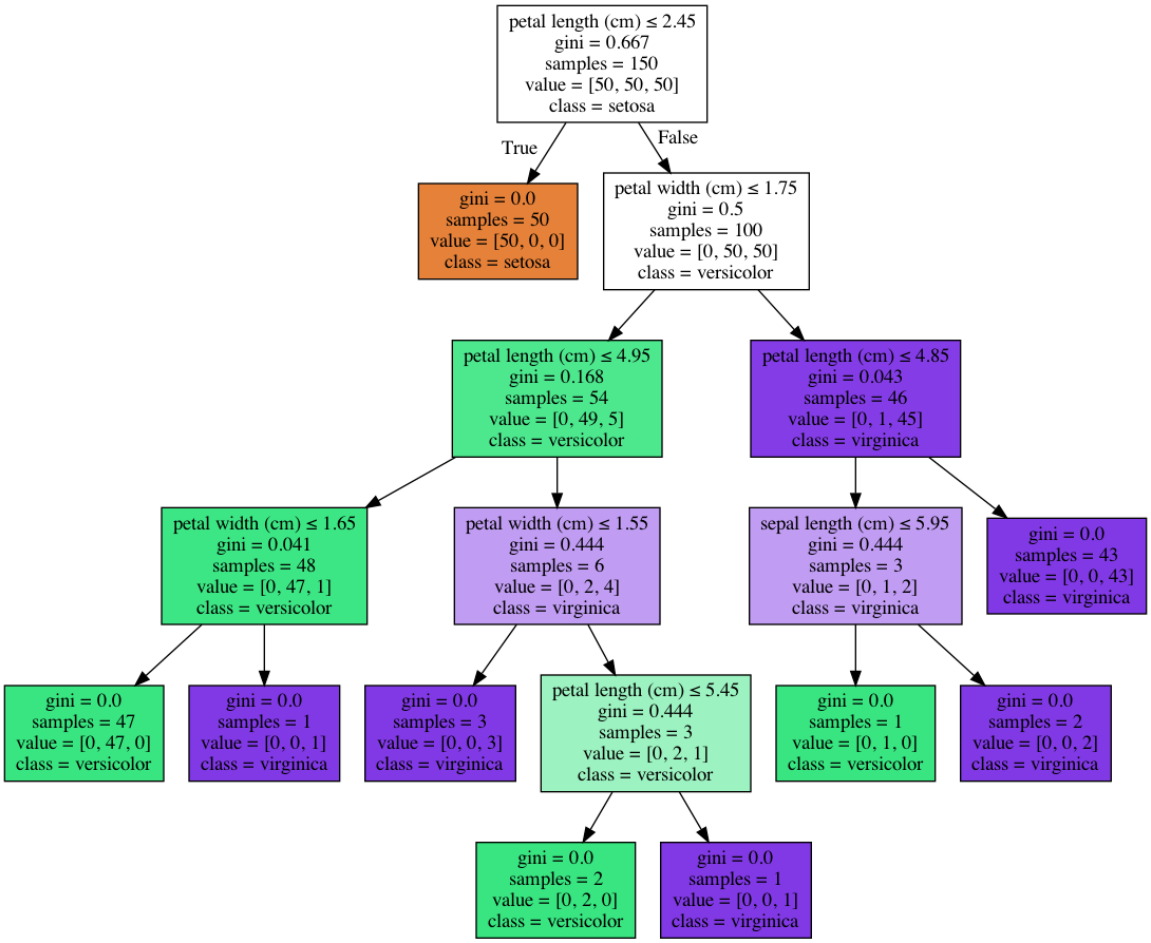


Figure 6 – Decision tree for the Iris Dataset

Support Vector Machines

Support Vector machines (also called support vector networks[77]) use a non-linear transformation to map input data to a very high dimensional space. By mapping data points to successively higher and higher dimensions via a non-linear transformation, a hyperplane can eventually be discovered to separate the classes. as illustrated in Figure 7. This process, known as kernelling, is especially useful for small, clean datasets, but becomes computationally intractable with large or poorly separable datasets. A number of

algorithms have been developed for AMP classification via SVM such as ClassAMP,[76] AntiPB,[64] iAMPpred,[49] and many others.[62,66,78–80]

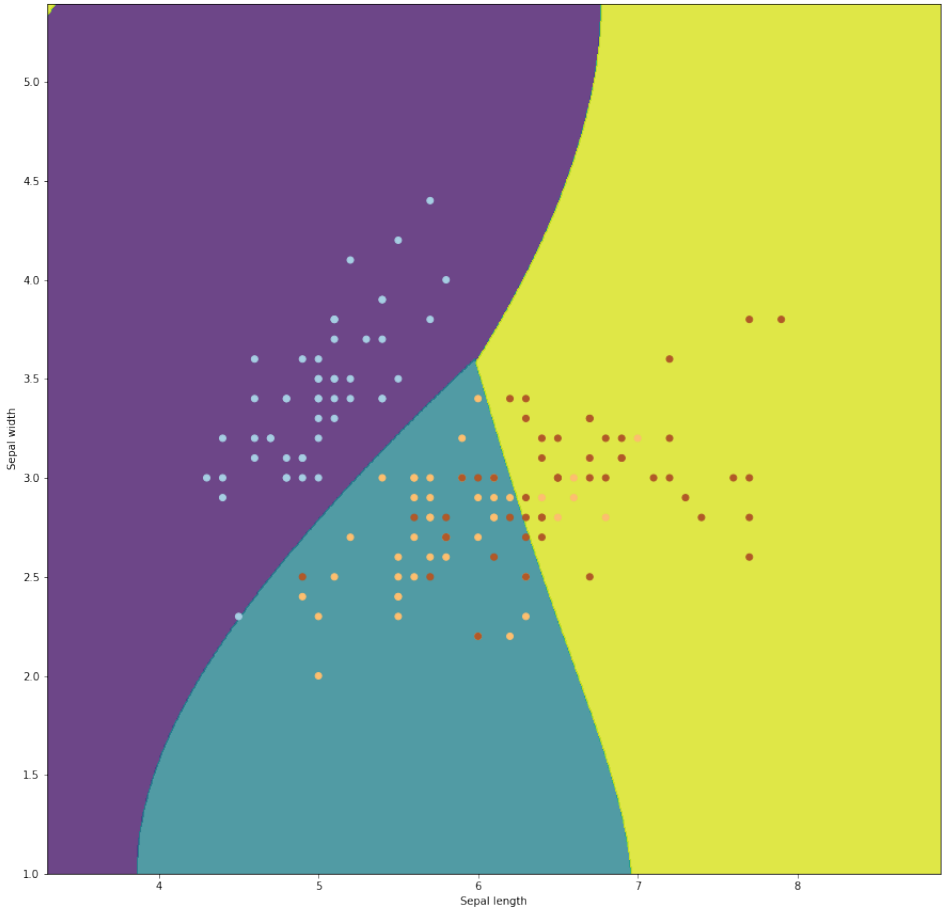


Figure 7 – Two dimensional SVM using the Iris dataset

Table 1 - MCC values for AMP prediction algorithms

Prediction Algorithm	**Feature Set**	**MCC (Test Dataset)**
FKNN[50]	PseAAC & physiochemical properties	0.73
SVM[53]	PseAAC, NAAC, physiochemical & structural	0.76

SVM[49]	AAC, PAAC, NAAC, Physiochemical & structural	0.89
QM[64]	N- and C- terminus fragments	0.74
ANN[64]	N- and C- terminus fragments	0.86
SVM[64]	N- and C- terminus fragments	0.82
ANN[66]	Aggregation and physiochemical properties	0.74
SVM[63]	PseAAC	0.83
ANFIS[71]	Aggregation and physiochemical properties	0.94
DA[51]	Physiochemical and peptide composition	0.74
RF[51]	Physiochemical and peptide composition	0.86
SVM[51]	Physiochemical and peptide composition	0.82
RF[75]	distribution patterns of amino acid properties	0.90

Additional Machine learning Algorithms evaluated in this work

In an attempt to fully quantify the success of the addition of 3D structural data to the AMP dataset a number of additional machine learning algorithms were evaluated in this work. The methods for optimization of these algorithms as well as the parameters used in this research will be more fully discussed in the methods section. The information presented below contains a general description of each algorithm.

Adaboost

Adaboost, short for adaptive boosting, is a dynamic allocation algorithm used with other machine learning algorithms to improve performance. By assigning higher weights to instances that are more difficult to classify correctly and to weak learners that correctly classify those instances an ensemble of weak learners is developed that classify instances more accurately than individual heuristics or ensembles where a simple mean is used for prediction.[79]

GaussianNB

The GaussianNB classifier is a probabilistic classifier based on applying Bayes' theorem that assumes that data features follow a gaussian distribution. The Naïve Bayes classifier assumes that all features are independent of each other (hence the term naïve) and attempts to generate a function to predict the probability of an unseen instance belonging to a certain class based on its features. Naïve Bayes can be generalized using the following equation:

$$p(AMP|f_1, f_2, \dots f_n) = \frac{p(AMP) * p(f_1, f_2, \dots f_n|AMP)}{p(f_1, f_2, \dots f_n)} \quad [10]$$

Since we assume that every feature is independent of every other feature and follows a gaussian distribution, we can calculate $p(f_i|AMP)$ as:

$$p(f_i|AMP) = \frac{1}{\sqrt{2\pi\sigma_{amp}^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_{amp}^2}\right)$$

$$Where\ \sigma_{amp}\ and\ \mu_{amp}\ are\ estimated\ using\ maximum\ likliehood. \quad [11]$$

The Naïve Bayes classifier has been used extensively in machine learning as well as for the in-silico classification of antimicrobial and anti-cancer peptides.[80–82]

Quadratic discriminant analysis (QDA)

QDA is a generalization of the linear Discriminant model described above. QDA follows that assumption that the measurements from each class are normally distributed but does not follow the assumption that the covariances for each class are identical. This allows the algorithm to learn non-linear classification boundaries making it more flexible.

Gradient Boosting Classifier (GBC)

GBC is a boosting algorithm in the same vein as the AdaBoost algorithm where an ensemble of weak learners is developed to produce a strong learning algorithm. Unlike the Adaboost classifier, the Gradient boosting classifier iteratively adds additional algorithms to reduce the residuals resulting from its predecessor. Each of these new weak learners is added to the ensemble algorithm based on the gradient descent optimization process. Given an imperfect ensemble learner, $F(x)$, and a perfect weak learner, $h(x)$, the gradient boosting algorithm would result in the following equation:

$$y = F_m(x) + h(x) = F_{m+1}(x) \qquad [12]$$

Supervised vs Unsupervised Classification algorithms

Machine learning tasks can be subdivided into supervised learning or unsupervised learning tasks. In the case of supervised learning data sets with known classification labels are available. These datasets can be subdivided into training, testing and validation sets. The chosen model architecture is trained using the training data after which a validation score is obtained using the previously unseen validation data. This

validation score is used to fine tune the various parameters unique to the chosen architecture. Parameters may include learning rates and numbers of layers for neural networks, maximum depth and number of estimators for tree-based classifiers or loss and normalization functions for many other algorithms. Once the model has been finalized it is scored using the testing dataset. This dataset has not been previously seen by the classifier and reduces the chances that the parameters have been tuned in a manner to overfit to the validation data. There are a number of ways to evaluate the score for a learning algorithm which will be discussed in the next section.

In an unsupervised learning task the true labels for the input data is unknown. In these types of tasks clustering algorithms, such as *k*-nearest neighbors or Agglomerative clustering, are used to group instances into clusters with similar properties. The classification label of each cluster is dependent upon the user. Since there is no known ground truth in an unsupervised learning task, evaluation of the technique is not a trivial as counting the number of times that the algorithm's prediction is correct. Techniques such as the measuring reconstruction error of a holdout set or by measuring intra-cluster density versus inter-cluster density, also known as the Calinski-Harabaz index.[83]

Evaluation of Classification algorithms

There are a number of metrics that may be employed for the evaluation of a classification algorithm. The utility of each of these metrics is based on the purpose of the algorithm and the available data. The overriding goal of evaluation is to reduce errors of both type I and type II. Type I error, the rejection of a true null hypothesis, or high false positive, would result in the classification of a peptide lacking antimicrobial activity as an

AMP, while type II error, the failure to reject a false null hypothesis, or high false negative, would result in the exclusion of a true AMP from the set of possible peptides with antimicrobial activity. Due to the desire to identify as many potential AMPs within the set of known peptides I will preferentially attempt to reduce type II errors.

k-fold cross validation

Machine learning is a data hungry procedure and only works well with large datasets. For this reason, researchers are often loath to reserve valuable training data for validation purposes, particularly when the available labeled datasets are already small (typically on the order of less than 1000 instances), as is most often the case in biological research. In these cases, a technique called k-fold cross validation may be used. Cross validation is a resampling technique in which the complete dataset is randomly divided into k groups (or folds) of approximately equal size. Each fold is then successively treated as the validation set while the algorithm is fit using the remaining k-1 folds. The mean squared error (MSE) for each holdout fold is computed resulting in k estimates of testing error. The k-fold cross validation error is computed by averaging these error values.[74]

$$\frac{1}{k}\sum_{i=1}^{k} MSE_i \qquad [13]$$

Evaluation Metrics

Confusion Matrix

A confusion matrix, also called an error matrix, is a graphical representation of the performance of an algorithm, typically a supervised learning algorithm. The rows of the matrix represent the number of predicted instances in each class while the columns represent the true number of instances in each class [Figure 8 - Example confusion matrix]. (There is no accepted convention for the construction of a confusion matrix, so the rows and columns may be reversed.)

		Predicted Class	
		True	False
True Class	True	True Positive	False Negative
	False	False Positive	True Negative

Figure 8 - Example confusion matrix

The confusion matrix is the bases for a number of other evaluation metrics that will be discussed below.

Accuracy

Accuracy is the degree to which a measurement or specification agrees with a reference value.[84] In terms of machine learning classification, it is the degree to which an algorithm predicts the true class of an instance. It can be calculated as follows:

$$ACC = \frac{TN + TP}{TN + TP + FN + FP} * 100\% \quad [14]$$

which can be simplified to:

$$ACC = \frac{total\ correct\ predictions}{total\ number\ of\ predictions\ made} * 100\% \quad [15]$$

Values may range from 0 to 100 percent with lager representing better classification performance.

Accuracy has the benefit of being an easy to understand metric but is only a starting point for algorithm assessment. In unbalanced sample sets, accuracy can be a poor measure of prediction success. Consider the goal of predicting cancer given a set of medical tests. The sample set consists of 500 individuals, only 30 of which actually have cancer. A model that predicts zero cancer diagnosis would be 94% accurate [16], but would miss 30 cancer diagnosis, potentially leading to disastrous outcomes (also known as high type II error).

$$ACC = \frac{470 + 0}{500} * 100\% = 94\% \quad [16]$$

In these cases, it may be preferable to use a less accurate predictive algorithm to avoid a specific type of error.

Sensitivity and specificity

Sensitivity and specificity are evaluation metrics devised by Jacob Yerushalmy in 1947 for the evaluation of binary classifications methods. Sensitivity, also known as the recall and true positive rate, measures the number of truly positive instances that are classified as positive.

$$Sensitivity = TPR = \frac{TP}{TP + FN} \quad [17]$$

Specificity, also called the true negative rate, is a measure of the truly negative instances that are classified as such.

$$Specificity = TNR = \frac{TN}{TN + FP} \quad [18]$$

High sensitivity means that few positive instances are overlooked, resulting in few false negatives. Conversely, high specificity means that few negative instances are classified as positive, resulting in few false positives.[85]

Matthew's Correlation Coefficient

The Matthew's Correlation Coefficient (MCC) is a measure of the quality of a binary classification.[86] The MCC is considered a good evaluation metric when the datasets contain different numbers of instances of each class. MCC can be calculated as:

$$MCC = \frac{(TP * TN) - (FN * FP)}{\sqrt{(TP + FN) * (TN + FP) * (TP + FP) * (TN + FN)}} \qquad [19]$$

MCC returns values between -1 and 1 with -1 indicating complete disagreement between the predicted results and the actual classification, 0 indicates predictions no better than chance and 1 represents complete agreement between the predicted and observed classes.

Receiver operating characteristic curves

Receiver operating characteristic curves or ROC curves are a graphical representation of the classification ability of an algorithm as its discrimination threshold is varied. A ROC curve is generated by plotting the true positive rate versus the false positive rate at different threshold values. In the case when a classifier returns a real value a threshold value is used to determine the cutoff point between a positive classification and a negative classification. In these cases, the choice of a threshold will directly affect the number of true positive vs false positives. A good classifier, resulting in classes with well separated distributions, would generate a ROC curve with a point close to the upper left position indicating high specificity (few false negatives) and high sensitivity (few false positives). As the classifier become less able to separate the instances based on the

input the ROC curve will more closely approach the diagonal line from the lower left to the upper right indicative of random guessing. A classifier that performs worse than guessing will exhibit a convex ROC curve. In this instance, were choice of threshold would depend on the purpose of the algorithm and a determination as to the desire to suppress type I or type II errors.

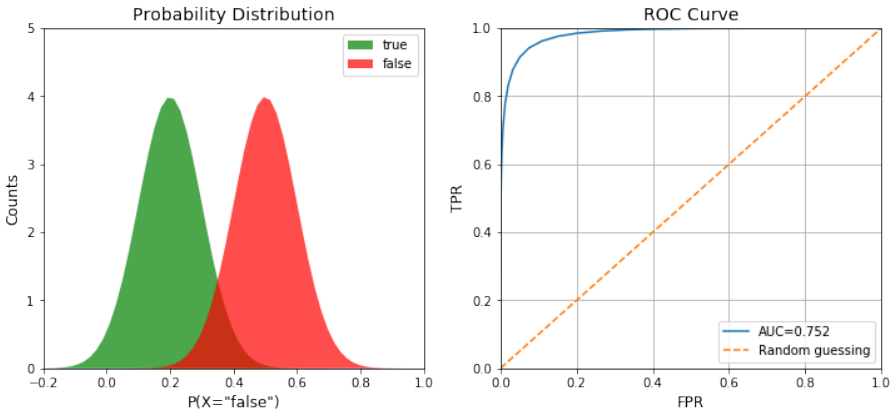


Figure 9 - ROC curve for classifier able to separate instances with a high degree of success

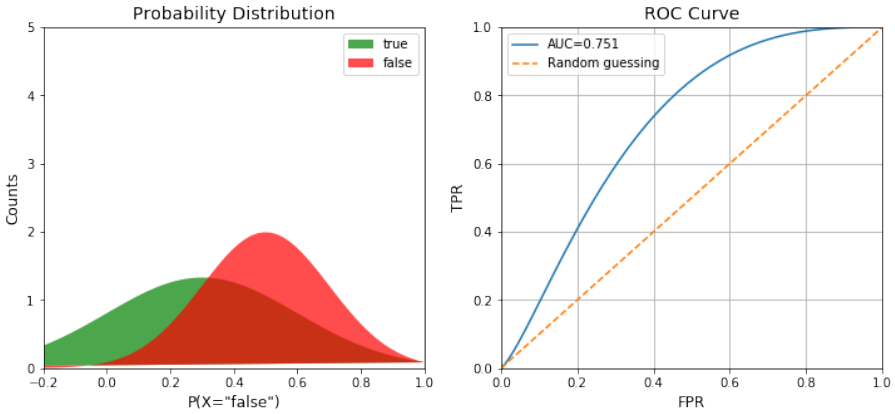


Figure 10 - ROC curve for classifier able to separate instances with a low degree of success

CHAPTER 4 - DATASET DEVELOPMENT AND FEATURE EXTRACTION

Introduction

Many machine learning algorithms require that the data input into the algorithm be in numerical format. Since peptides are a physical structure it is necessary to encode the peptide in some way that allows for a numerical representation while maintaining the information contained within the sequence and structure of the molecule. We propose a method using trigrams to encapsulate the sequence and secondary structure information and a set of simplexes to encode the tertiary structure information. In order to reduce the number of potential trigrams we also evaluate the use of five different alphabet reduction schemes.

Methods

In this binary classification setting the learning task is to predict if a peptide possesses an antimicrobial property. To achieve this goal, a dataset must be developed that is composed of features that numerically describe the peptide and a label. The label indicates if the peptide in question is antimicrobial or not and is considered the ground truth for the purposes of this experiment. The following section will describe the process of developing a data set which includes features that represent the primary, secondary and tertiary structure of each peptide.

Amp selection

While there are a number of publicly available dataset of antimicrobial peptides, unfortunately, there are no sample sets that also contain equivalent negative examples. In order to generate a complete dataset with both positive and negative samples a number of sources were utilized. Below we will describe the selection of both positive and negative AMPs included in the dataset. The protein sequences as well as the DSSP sequences for each peptide is provided at http://omics.gmu.edu/ssnaap/assets/supplemental.html.

Positive set selection

The information for known antimicrobial peptides was obtained from the cAMP database[15] The cAMP database contains over 8000 antimicrobial sequences, 757 of which have empirically derived structures available. Of these 757 sequences, 213 were excluded because the DSSP structure sequences were not available in the DSSP archive. Another 104 were excluded because they were determined to be either too long or too short. Any sequence less than 10 amino acids long were excluded because the information content was determined to be too small, while those with greater than 140 were also excluded. The longer sequences were excluded because it is believed that since the antimicrobial portion of a peptide is generally less than 50 amino acids the additional, non-antimicrobial, sequence would obscure the active region and confound the learning algorithm. An additional 14 instances were removed due to poor overlap between the antimicrobial sequence and the DSSP. The final positive set contains 426 known antimicrobial peptides with a variety of activity types.

Negative set selection

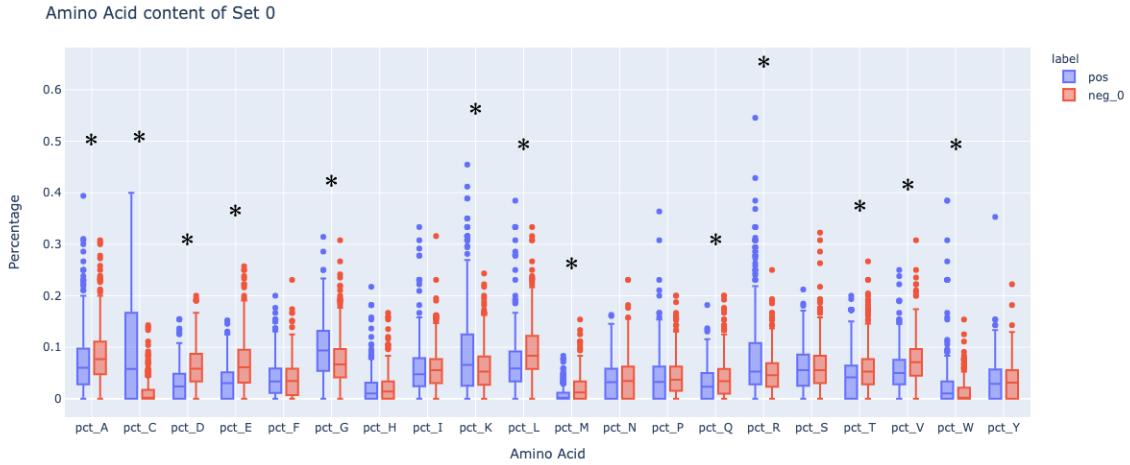
A total of 5 negative sets were produced. The decision to generate redundant negative data sets was made in order to compensate for the low probability that peptides with antimicrobial activity, but that have not been classified as such, may have been included in the negative set. By using multiple sets the rare addition of a false negative could be offset by the abundance of true negatives.

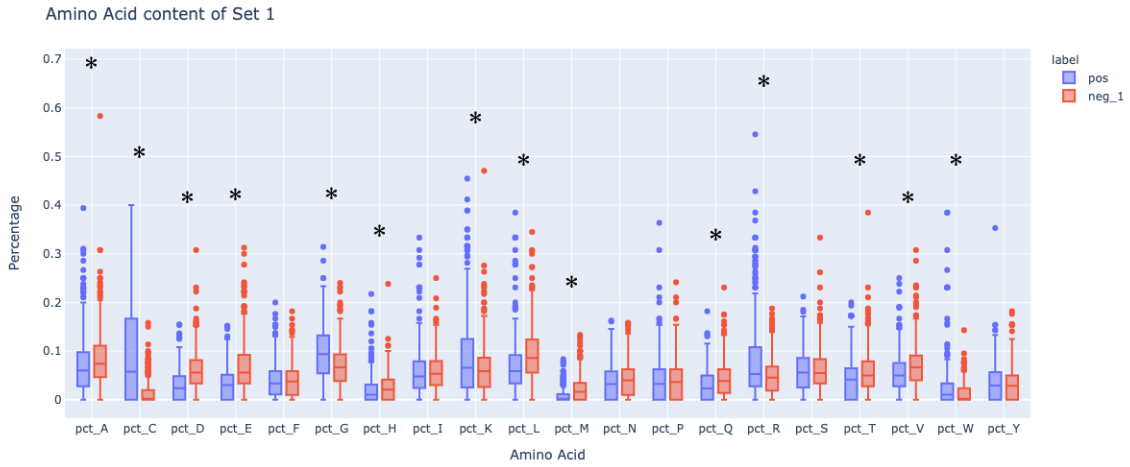
Each negative set was made to pair with the positive set by selecting peptide fragments with less than 50% sequence identity to any known AMP sequence. The negative set was generated using peptide fragments to match the length of its paired AMP. All five negative datasets can be found at http://omics.gmu.edu/ssnaap/assets/supplemental.html. Similar negative datasets were used by previous publications.[51,66,77,87]

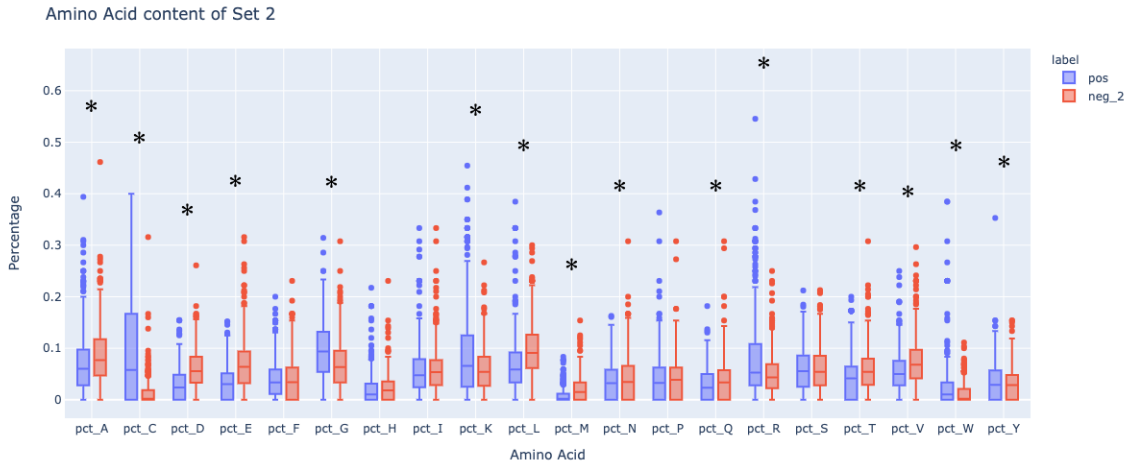
Feature Engineering

Amino Acid content

Amino acid content for each dataset was calculated using the protein analysis module contained within BioPython.[15] Amino acid composition of the original sequences can be found in Figure 11.


Amino Acid content of Set 0
label
pos
neg_0
Percentage
0
0.1
0.2
0.3
0.4
0.5
0.6
pct_A
pct_C
pct_D
pct_E
pct_F
pct_G
pct_H
pct_I
pct_K
pct_L
pct_M
pct_N
pct_P
pct_Q
pct_R
pct_S
pct_T
pct_V
pct_W
pct_Y
Amino Acid


Amino Acid content of Set 1
label
pos
neg_1
Percentage
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
pct_A
pct_C
pct_D
pct_E
pct_F
pct_G
pct_H
pct_I
pct_K
pct_L
pct_M
pct_N
pct_P
pct_Q
pct_R
pct_S
pct_T
pct_V
pct_W
pct_Y
Amino Acid


Amino Acid content of Set 2
label
pos
neg_2
Percentage
0
0.1
0.2
0.3
0.4
0.5
0.6
pct_A
pct_C
pct_D
pct_E
pct_F
pct_G
pct_H
pct_I
pct_K
pct_L
pct_M
pct_N
pct_P
pct_Q
pct_R
pct_S
pct_T
pct_V
pct_W
pct_Y
Amino Acid

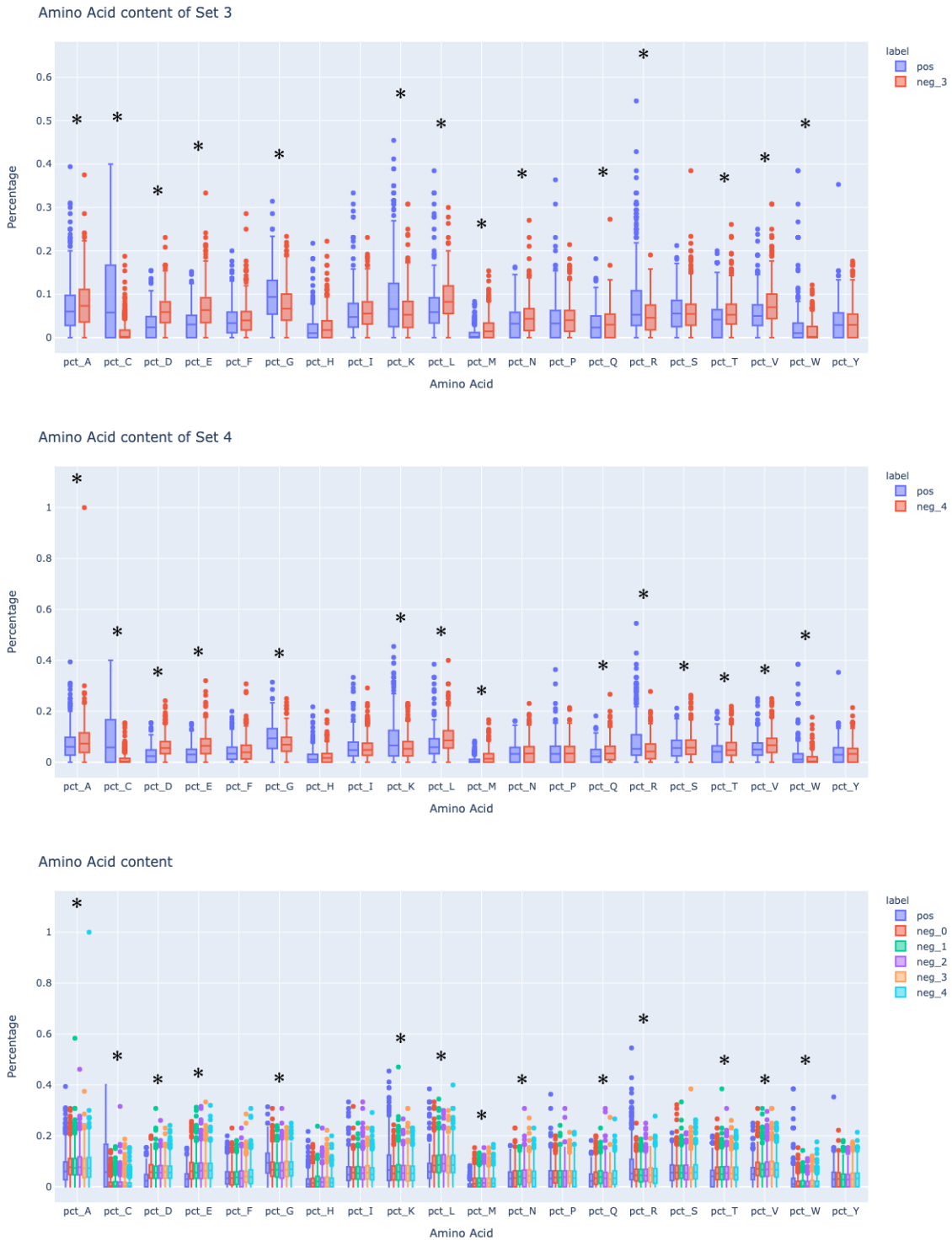
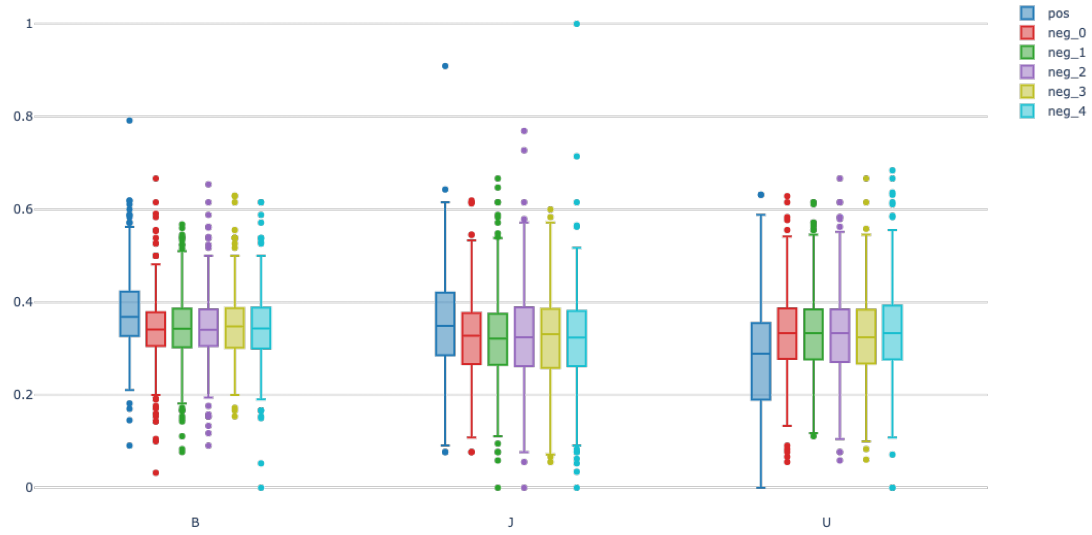
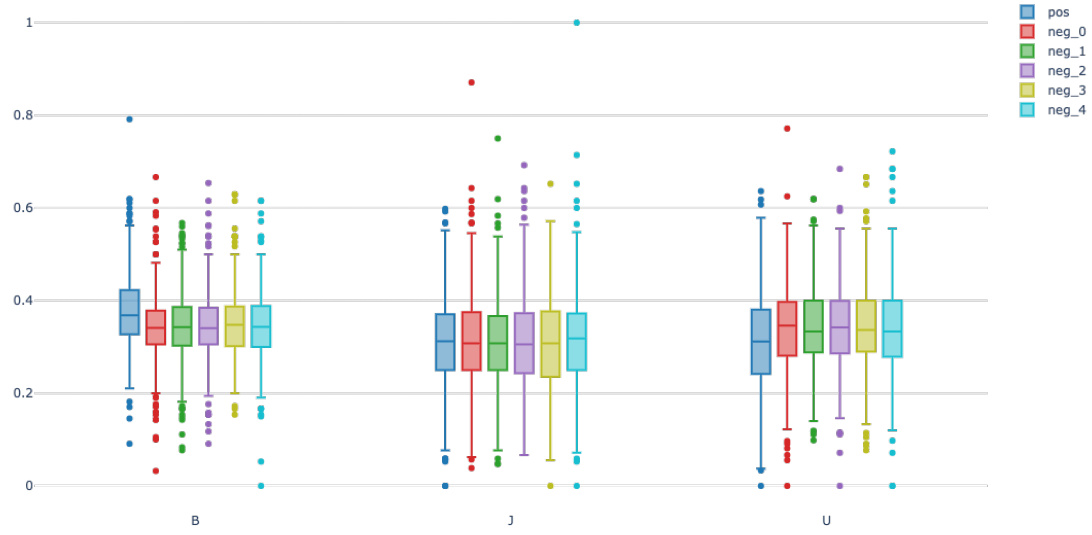


Figure 11 - Distribution of amino acids in positive and negative samples. .

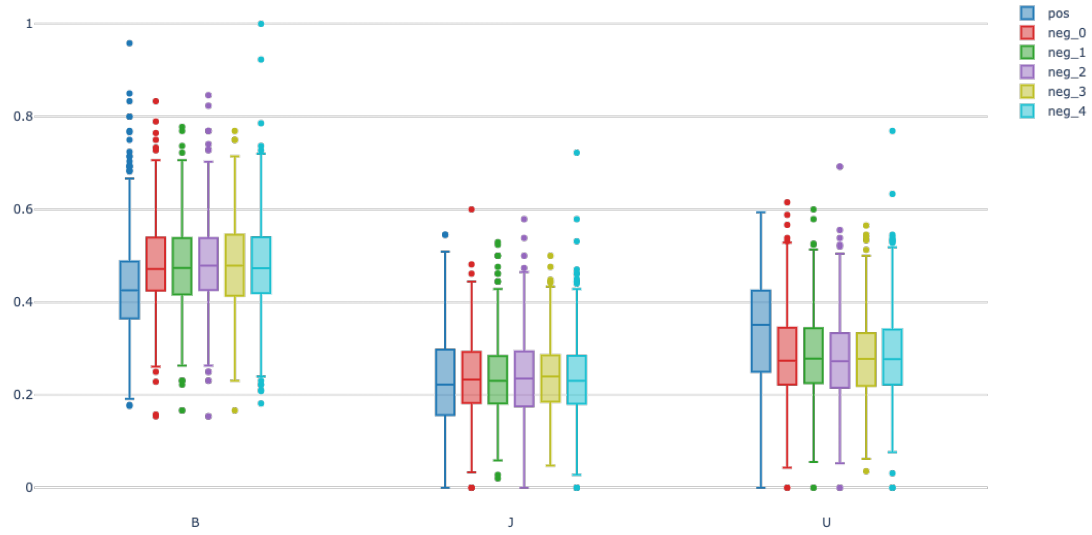
Reduced alphabet A amino acid distributions



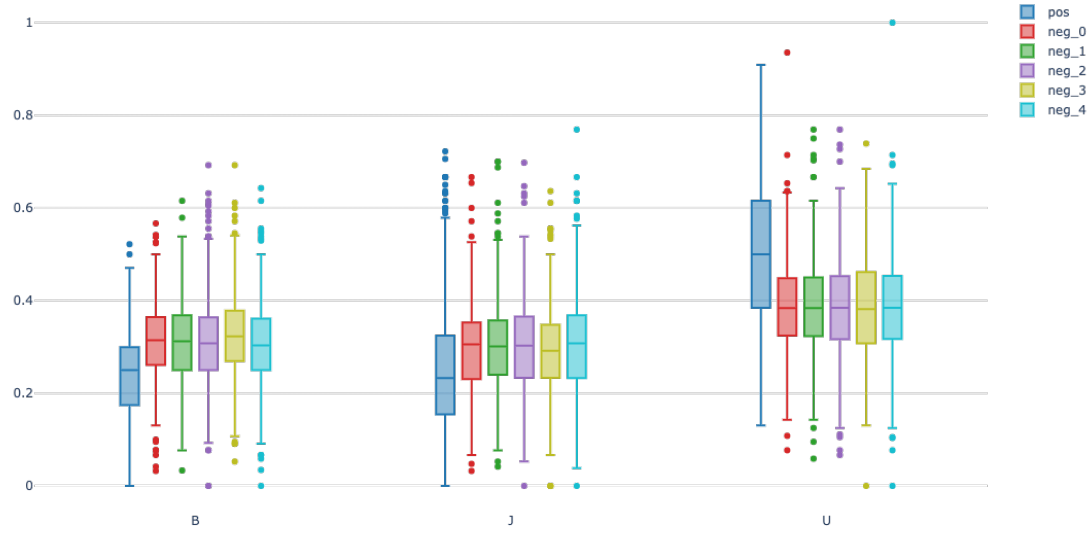
Reduced alphabet B amino acid distributions



Reduced alphabet C amino acid distributions



Reduced alphabet D amino acid distributions



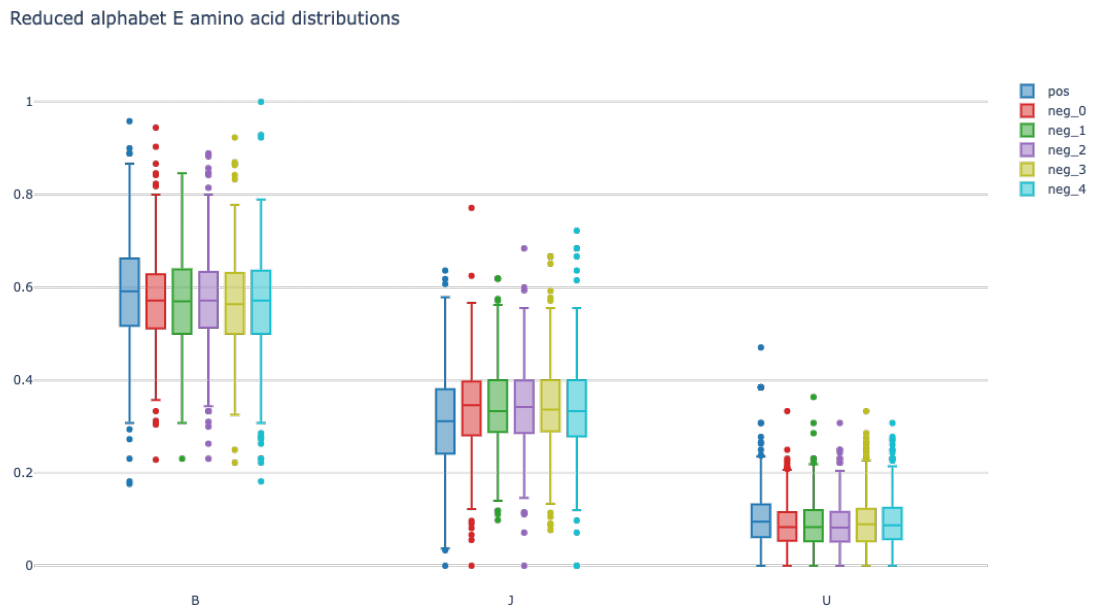


Figure 12 - Distribution of reduced alphabet amino acids

Alphabet reduction

Each amino acid sequence was reduced from the standard 20 letter alphabet to a three-letter alphabet. 5 different alphabets resulted in 5 sequence iterations. The alphabets chosen were previously shown to produce good results in the classification of AMP's.[60]

Table 2: Schemes for alphabet reduction

	Letter 1 – B	**Letter 2 – J**	**Letter 3 - U**	**Grouping strategy**
Reduced alphabet A[88]	CMFILVWY	ATHGPR	DESNQK	Minimized mismatch

Reduced alphabet B[87]	CMFILVWY	GPATS	EKRDNQH	BLOSUM & Monte Carlo method
Reduced alphabet C[89]	AVFILPMG	DEKR	STYCNQHW	Chemical properties
Reduced alphabet D[90]	MHVYNDI	QLEKF	WPRGSATC	Molecular recognition theory
Reduced alphabet E[91]	LASGVTIPMC	EKRDNQH	FYW	BLOSUM & deterministic reduction

Alphabet reduction grouping strategies

Strategies for amino acid alphabet reduction abound within the biological literature. The most simplistic reduction scheme, known as the HP model where H stands for hydrophobic and P for polar, consists of only two letters and uses hydrophobicity as the decision vector. However, previous research with the HP model has determined that more than two residues are required for successful modeling of protein structure.[92–94] As a result a more detailed pattern than that available from a two letter alphabet is required. In addition to those discussed below alphabets based on cassette mutation,[95] reduction of a binary dendrogram,[96] Finite Information Theory with respect to backbone structure,[61] Manhattan and Euclidian distances in the Miyazawa-Jernigan matrix (a matrix of pairwise interaction potentials between amino acids),[97] and others.[94,98–100]

With any reduction in alphabet complexity there is an inherent loss of information content. However, as the number of letters in the reduced alphabet decreases, the number of n-gram combinations decreases exponentially. Therefore, is in the best interest of this research, especially considering the limited dataset, to use as small an alphabet as

possible. In a case where there are a vast number of sparse features overfitting is highly likely. Due to the fact that only a finite amount of peptide sequence is available and the sparse nature of this data, reduction in feature number is necessary to increase the statistical significance of the results.

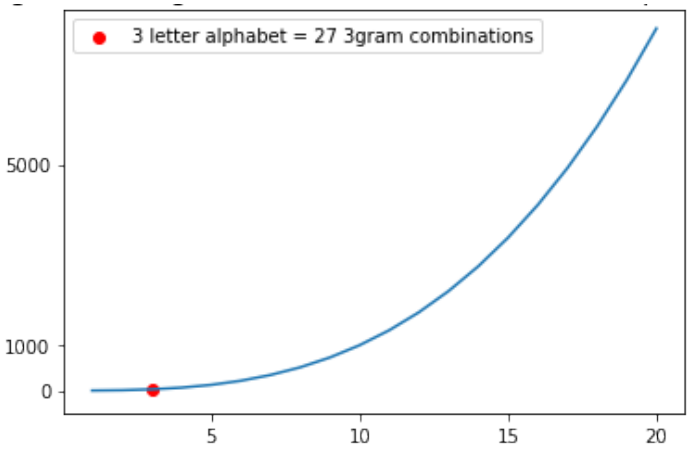


Figure 13 - Number of 3-gram combinations as a function of alphabet size

The general strategy for grouping residues into a single monomer rests on the assumption that each residue can be placed into a group with other residues with a similar chemical or physical characteristic. In order to reduce the number of groups, the individual residues in the two groups should interactions similar to each other, but distinct from those of residues in the other groups.

Five different 3 letter alphabet reduction grouping strategies are considered for this work. Each of these strategies exploits a different component of amino acid composition, structure or homology. These strategies will be more completely discussed below.

Reduced alphabet A - Minimized Mismatch

The minimized mismatch strategy[88] is a reduction strategy based on minimizing the mismatches between a reduced matrix and the statistical contact potentials of the Miyazawa and Jernigan (MJ) matrix versus the number of reduced residues. The MJ matrix measures inter-residue contact energies between pairs of amino acids.[101]

For this strategy, mismatch is defined as the discrepancy of properties between elements and blocks. A reduction algorithm rests on the idea that amino acids can be placed in a number of groups in which each group has different physical and chemical properties than the others. For example, assume that all residues can be placed in one of 5 blocks (A, B, C, D or E) based on physical and chemical properties. Elements in one group would should have different reactions with each other than they would with members outside of the group, but the interactions between, for example, groups A and B should be similar for all amino acids in those groups. Furthermore assume that the relative interactions between residues in group A and B are less than the interactions between residues in group C and D. if a single residue pair in block AB, A(i) and B(j), interact more strongly than a residue pair in CD, C(k) and D(l), then this interaction is described as a mismatch. The reduction algorithm would attempt to minimize these mismatches.

Using this technique Wang and Wang were able to describe a 3-letter alphabet with an average contact overlap of about 0.8. Contact overlap is defined as the number of common contacts in the native structure and the lowest energy structure, as determined

by a Monte Carlo method described in Xing et al.,[102] normalized by the maximal number of native contacts.

Reduced alphabet B – BLOSUM65 & Monte Carlo Reduction

The BLOSUM64/Monte Carlo Reduction strategy exploits the commonly used BLOSUM62 (blocks substitution matrix) originally proposed by Henikoff and Henikoff in 1992.[103] This matrix represents the log odds ratio of one amino acid substituting another in sequences with 62% similarity. A heuristic MC method is used to approach the optimal solution by attempting to maximize the similarity score between the reduced peptide sequence generated from an initially randomly generated reduction scheme. For each iteration of the MC method the groupings of two residues are switched based on a Metropolis criterion where the probability of accepting the move is:

$$P = \exp(S_{old} - S_{new}) / T_{MC} \qquad [20]$$

Where S_{old} is the total similarity score for simplified sequences prior to the switch and S_{new} is the total similarity score for the simplified sequences after the switch. If P is larger than a randomly generated number between 0 and 1 the move is accepted if not, the move is rejected.

Approximately 10^7 MC iterations were required to find the maximum score for each possible combination of amino acid groupings. Amino acid groupings consist of the number of sets that characterize the number of residues in each group. For example, given an alphabet of 3 characters there are 33 sets such as (1,1,18), (1,2,17), (1,3,16) etc.

The maximum similarity score was calculated for each set and the maximum of all sets was chosen as the final reduced alphabet.

The resulting 3-letter reduced alphabet produced a coverage of about .35 when normalized to the coverage provided by a 20 letter alphabet. Coverage is determined by comparing a reduced sequence to a known homolog using the BLASTP program. Coverage is defined as the number of protein sequence pairs with an aligned score larger than an expectation threshold (E-value) divided by the total number of homologous pairs (9044 in the SCOP40 database used).

Reduced alphabet C – Chemical Properties

Amino acids are grouped based on the charge of their side chain. The polar amino acids are asparagine, cystine, glutamine, histidine, serine, threonine, tryptophan and tyrosine. These residues are commonly found on the exterior of the peptide and may form hydrogen bonds with other amino acids or external ligands. The charged amino acids group is made up of arginine, aspartic acid, glutamic acid and lysine. Histidine is not considered to be a charged amino acid in this instance because with a side chain pKa of ~6.0 only the Henderson-Hasselbach equation would predict that only about 10% of the residues will be protonated at biological pH. Charged amino acids are most commonly found on the exterior of a peptide proven important to the peptide. These residues also form salt bridges that have proven important to the stabilization of three-dimensional peptide structure. Finally, the hydrophobic amino acids are alanine, isoleucine, leucine, methionine, phenylalanine, proline, and valine. These amino acids are most often found buried within the peptide structure or interacting with lipid layer of a membrane. Glycine

is included in the hydrophobic group as it is generally considered to be ambivalent, but is not polar or charged.

Reduced alphabet D – Molecular Recognition Theory

This reduction technique is based on work by LB Mekler and the Mekler-Idlis (M-I) pair theory[104] which states that amino acids may make specific pairwise interactions with the amino acid coded for by the reverse complement codon. Mekler proposed that the genetic code contained was able to specify-through space interactions between pairs of amino acid residues.

For example, given the codons for Glutamic Acid are GAA and GAG, their reverse complements would be UUC and UCU. These codons encode the amino acids phenylalanine and leucine. Mekler and Idlis identified all of the possible sense–antisense amino acid residue partnerships segregated them into three non-overlapping groups [Figure 14].

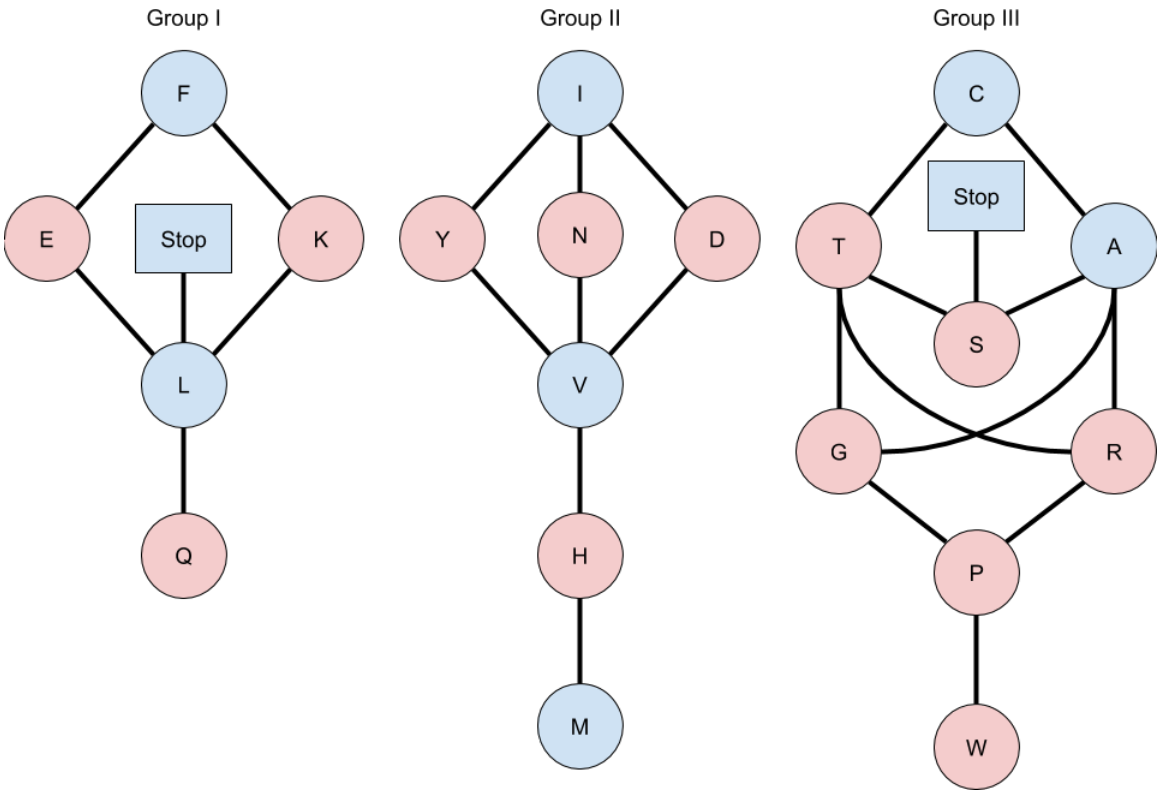


Figure 14 - Non-overlapping sets of Mekler-Idlis amino acid pairs. Amino acids are represented by their one letter designation. Pink circles indicate polar residues while blue circles indicate non-polar residues.

Reduced alphabet E – BLOSUM50 & deterministic reduction

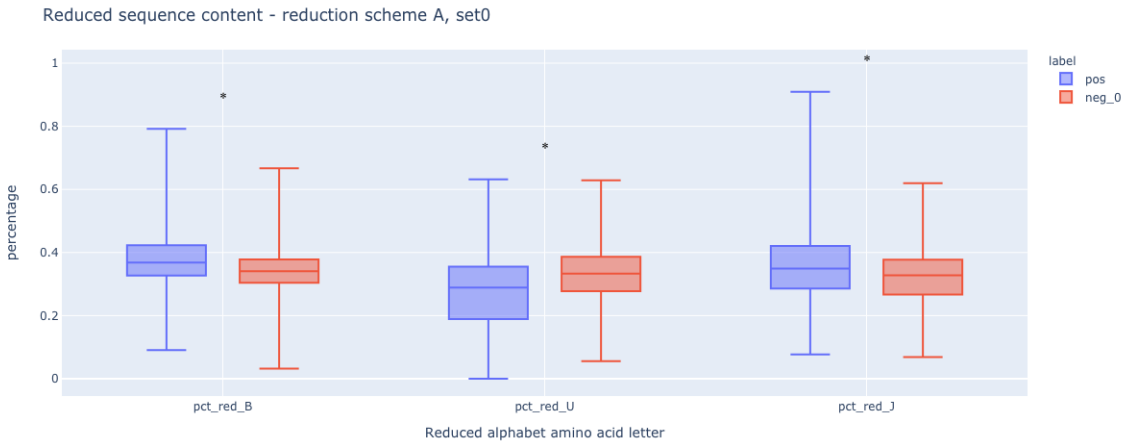
Similar to the BLOSUM similarity scoring strategy discussed above, this method clusters amino acids based on the BLOSUM50 similarity scores, but it does not make use of a Monte Carlo method. Instead, a deterministic method for clustering is devised by first calculating the correlation coefficients for all similarity matrix elements for all pairs of amino acids using the following equation:

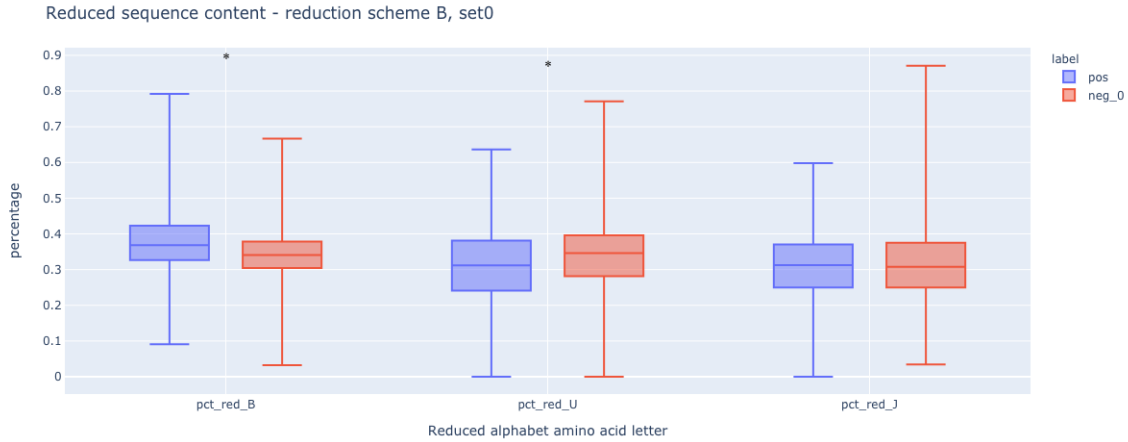
$$C_{A_1,A_2} = \frac{\sum_{i=1}^{20} M_{A_1,i} * M_{A_2,i}}{(\sum_{i=1}^{20} M_{A_1,i} * M_{A_1,i})(\sum_{i=1}^{20} M_{A_2,i} * M_{A_2,i})} \quad [21]$$

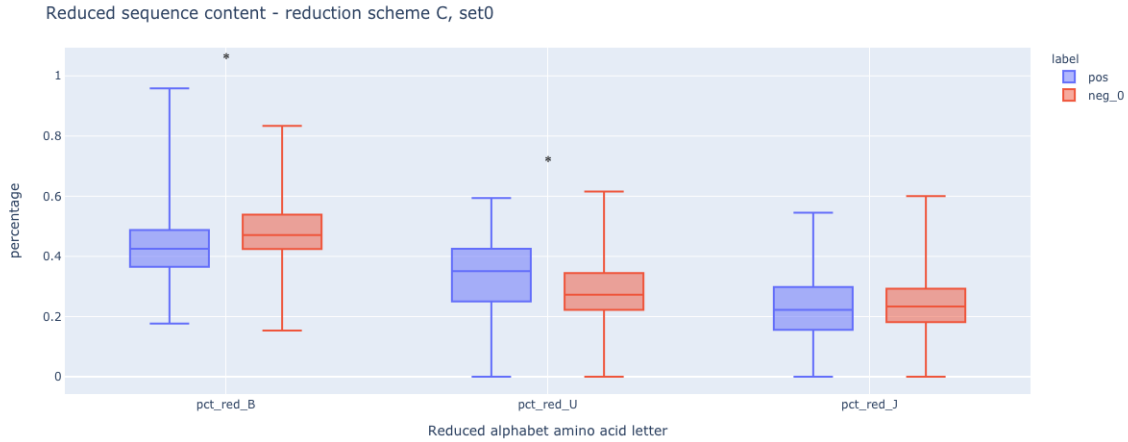
After all correlations have been calculated the pair with the highest correlation is placed in a group. The next highest correlated pair is either added to the first group, if one letter of the pair is in the first group, or separated into a new group. This grouping continues until the desired number of groups has been reached. Matrix values for grouped elements are calculated based on the average of the constituent matrix elements.

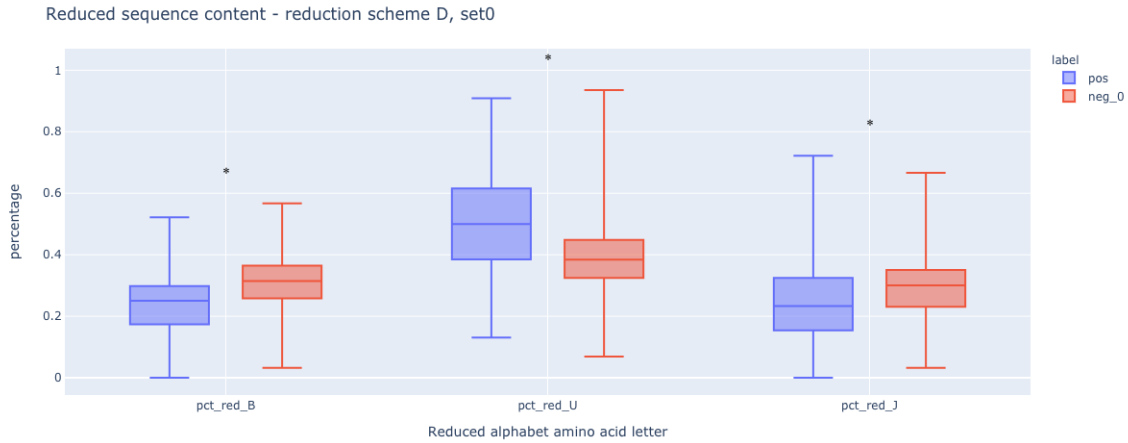
The resulting peptides with reduced complexity were evaluated much the same way that those developed using the BLOSUM64 and Monte Carlo method were evaluated using coverage of homologs in the SCOP40 database. The resulting coverage rates for this method were lower than those found for the BLOSUM64 method at small alphabets but increased rapidly for larger alphabets until the coverage exceeded that of the BLOSUM 64 method starting at alphabets of greater than 6 letters.

The sequences resulting from alphabet reduction produced amino acid analogs with distributions visualized in Figure 15




Reduced sequence content - reduction scheme B, set0
percentage
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
pct_red_B
pct_red_U
pct_red_J
Reduced alphabet amino acid letter
label
pos
neg_0


Reduced sequence content - reduction scheme C, set0
percentage
1
0.8
0.6
0.4
0.2
0
pct_red_B
pct_red_U
pct_red_J
Reduced alphabet amino acid letter
label
pos
neg_0


Reduced sequence content - reduction scheme D, set0
percentage
1
0.8
0.6
0.4
0.2
0
pct_red_B
pct_red_U
pct_red_J
Reduced alphabet amino acid letter
label
pos
neg_0

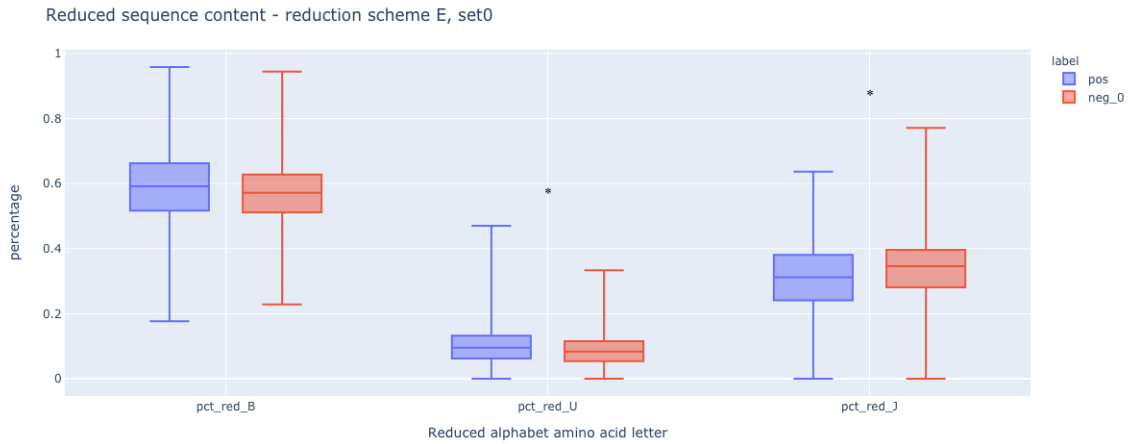


Figure 15 - Distribution of Amino Acid analogs

N-gram generation

Due to the small size of AMP's when compared to proteins the choice of a relatively small N was deemed prudent in order to avoid an unnecessarily high number of features when compared to the size of the dataset. Since it has been previously demonstrated that the use of trigrams results in a high level of accuracy in using both Naïve-Bayes and Decision tree algorithms for the classification of proteins at the family level[49] the N for this experiment was set to three.

Using a sliding window method, each full set of 3 consecutive residues was considered to be a tri-gram. The numbers of each of the possible 27 tri-grams were summed for each peptide. The frequency of each tri-gram was determined by dividing the total number of that tri-gram by the number of tri-grams possible in a given sequence (M).

$$M_{sequence} = Length - (n - 1) \quad [22]$$

$$F = \frac{I_{n-gram}}{M_{Sequence}} \quad [23]$$

Peptide frequencies were then normalized by dividing the frequency of the tri-gram by the frequency of each individual symbol.

$$F_{likelihood} = \frac{F_{ijk}}{f_i * f_j * f_k} \quad [24]$$

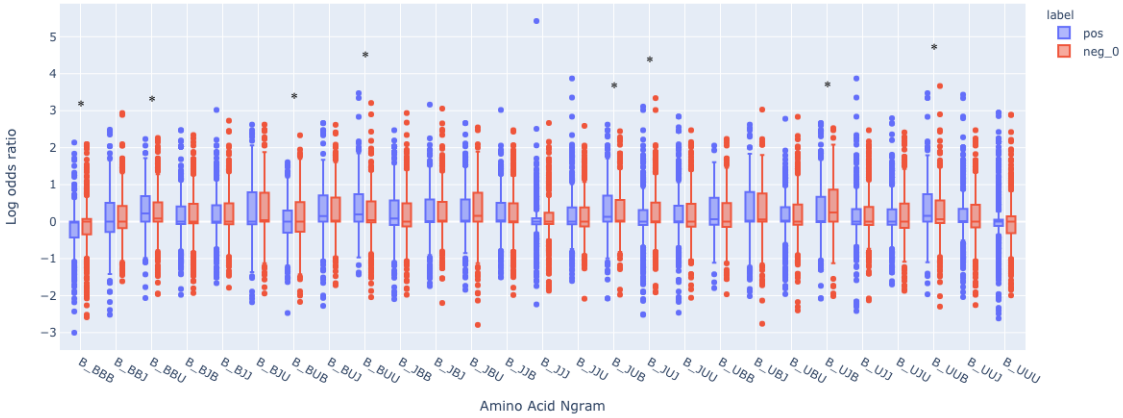
Finally the log of the likelihood for each tri-gram was taken and placed into a dataset for analysis.

$$logLikelihood = \log_2(F_{likelihood}) \quad [25]$$

In peptides where there were 0 instances of a particular tri-gram the log likelihood is artificially set to 0 to avoid null data.

Using a log of the likelihood results in a measure centered around 0 in which positive numbers represent frequencies greater than chance and negative numbers represent frequencies occurring less than would be expected by chance. Tri-grams were developed for each amino acid representation [Figure 16] and the structure sequence [Figure 18] in the same manner.


Amino Acid Ngram distribution of Set 0, reduction scheme A
label
pos
neg_0
Log odds ratio
Amino Acid Ngram


Amino Acid Ngram distribution of Set 0, reduction scheme B
label
pos
neg_0
Log odds ratio
Amino Acid Ngram


Amino Acid Ngram distribution of Set 0, reduction scheme C
label
pos
neg_0
Log odds ratio
Amino Acid Ngram

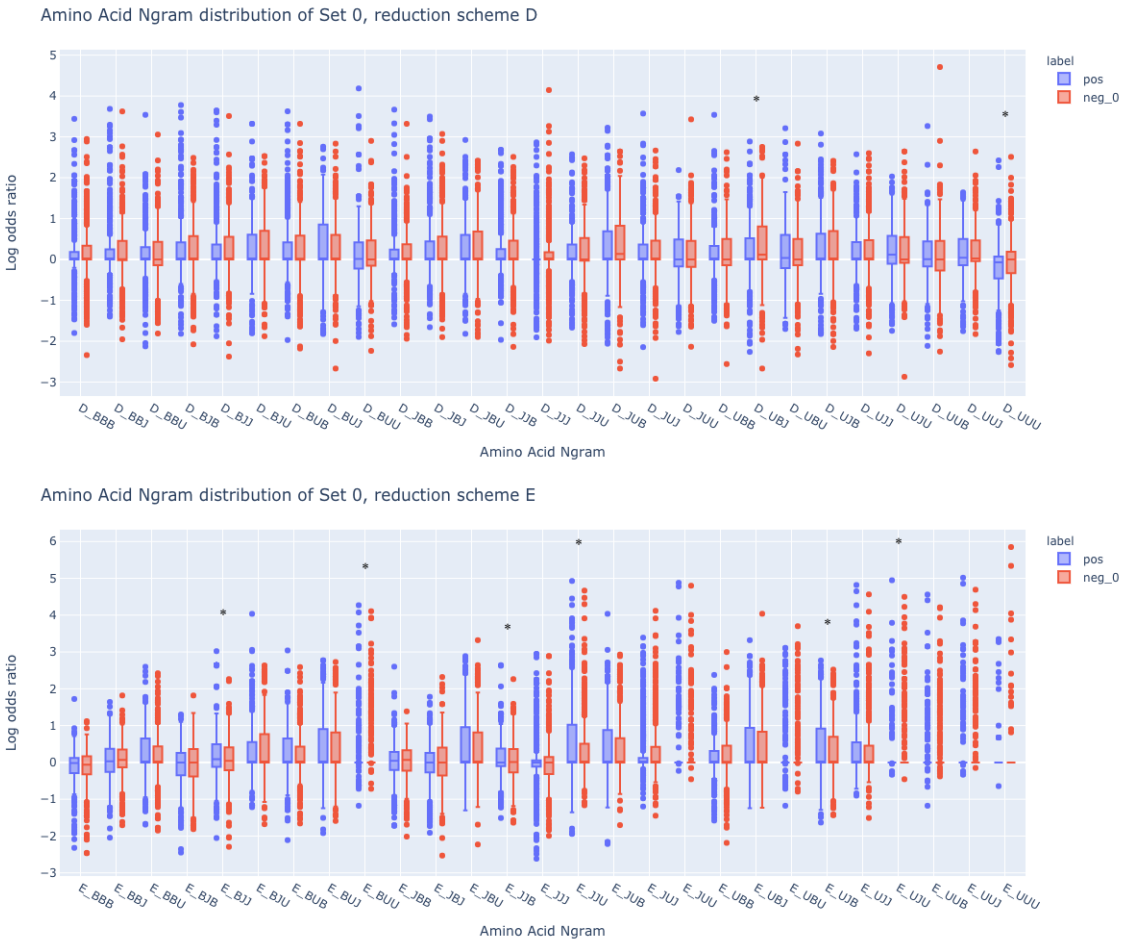


Figure 16 - Trigram log odds ratios for sample set 0.

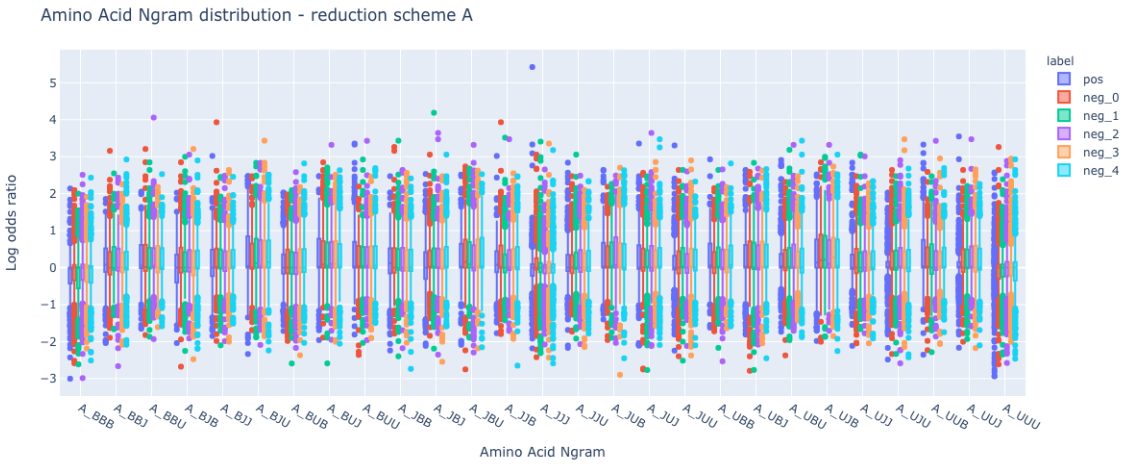
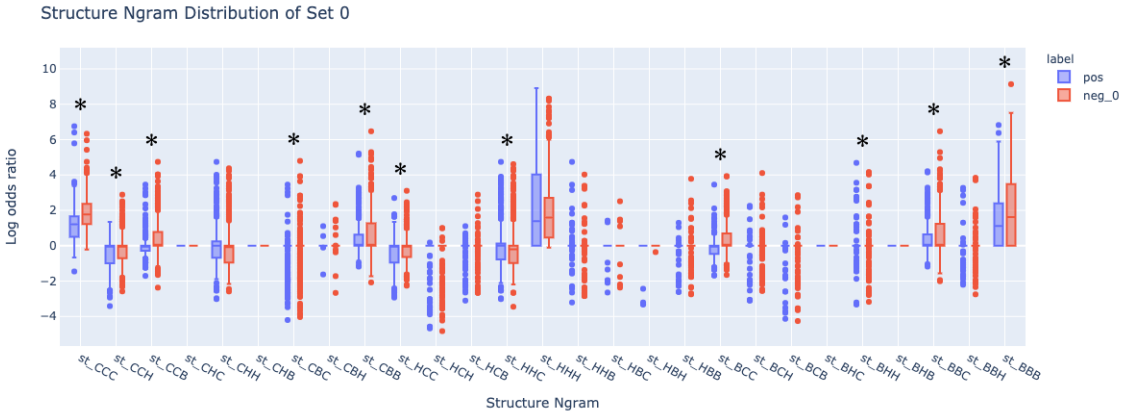
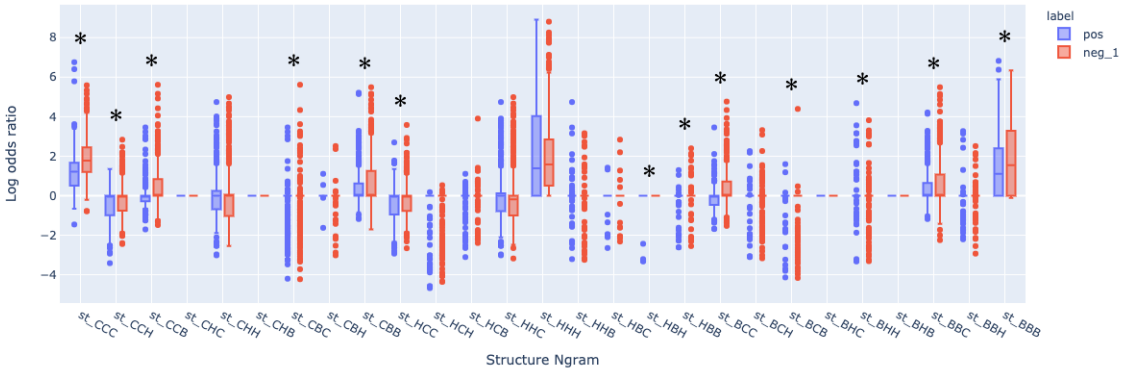


Figure 17 - Trigram log odds ratio of all datasets using reduction scheme A


Structure Ngram Distribution of Set 0
Log odds ratio
Structure Ngram
label
pos
neg_0
st_CCC st_CCH st_CCB st_CHC st_CHH st_CHB st_CBC st_CBH st_CBB st_HCC st_HCH st_HCB st_HHC st_HHH st_HHB st_HBC st_HBH st_HBB st_BCC st_BCH st_BCB st_BHC st_BHH st_BHB st_BBC st_BBH st_BBB


Structure Ngram Distribution of Set 1
Log odds ratio
Structure Ngram
label
pos
neg_1
st_CCC st_CCH st_CCB st_CHC st_CHH st_CHB st_CBC st_CBH st_CBB st_HCC st_HCH st_HCB st_HHC st_HHH st_HHB st_HBC st_HBH st_HBB st_BCC st_BCH st_BCB st_BHC st_BHH st_BHB st_BBC st_BBH st_BBB

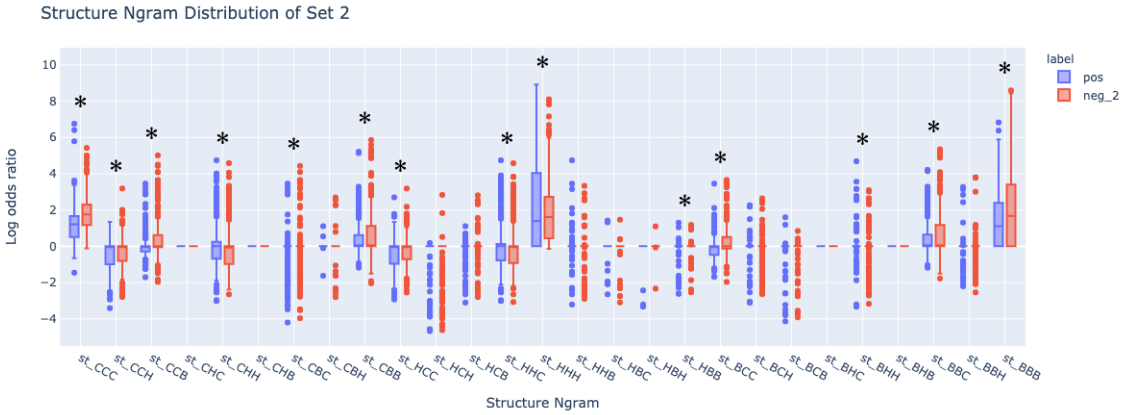

Structure Ngram Distribution of Set 2
Log odds ratio
Structure Ngram
label
pos
neg_2
st_CCC st_CCH st_CCB st_CHC st_CHH st_CHB st_CBC st_CBH st_CBB st_HCC st_HCH st_HCB st_HHC st_HHH st_HHB st_HBC st_HBH st_HBB st_BCC st_BCH st_BCB st_BHC st_BHH st_BHB st_BBC st_BBH st_BBB

Figure 18 – Trigram Log odds ratio of structure data

Simplex Generation

To capture the tertiary structure of the peptides in a tabulated form the choice was made to analyze the simplexes generated through the use of Delaunay triangulations.

Delaunay triangulation on a plane results from the generation of all triangles from a set of points in which the circumcircle of each triangle (the circle containing in each point) does not contain any other points. In a three-dimensional structure the triangles are replaced by polygons and the circumcircles become circumspheres. In the case of peptide triangulation, the vertices are represented by each C_α atom the resulting Delaunay simplex represents four nearest neighbors $C_\alpha$and therefore four nearest neighbor residues. Delaunay tessellations have been used in structural analysis of peptides in the past and exhibit good potential as a structural description method.[105,106] Peptide composition analysis of Delaunay simplexes has shown that residues associate in a highly non-random manner, with some simplexes appearing orders of magnitude more or less frequently than would be expected by chance alone.[107]

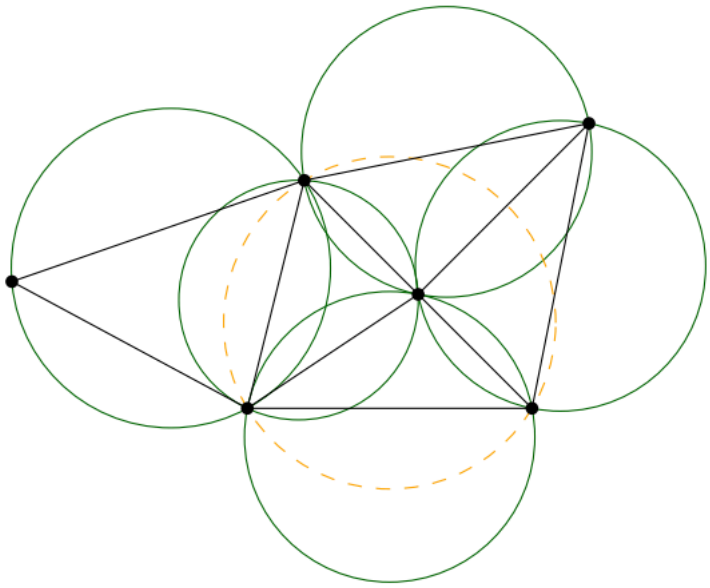
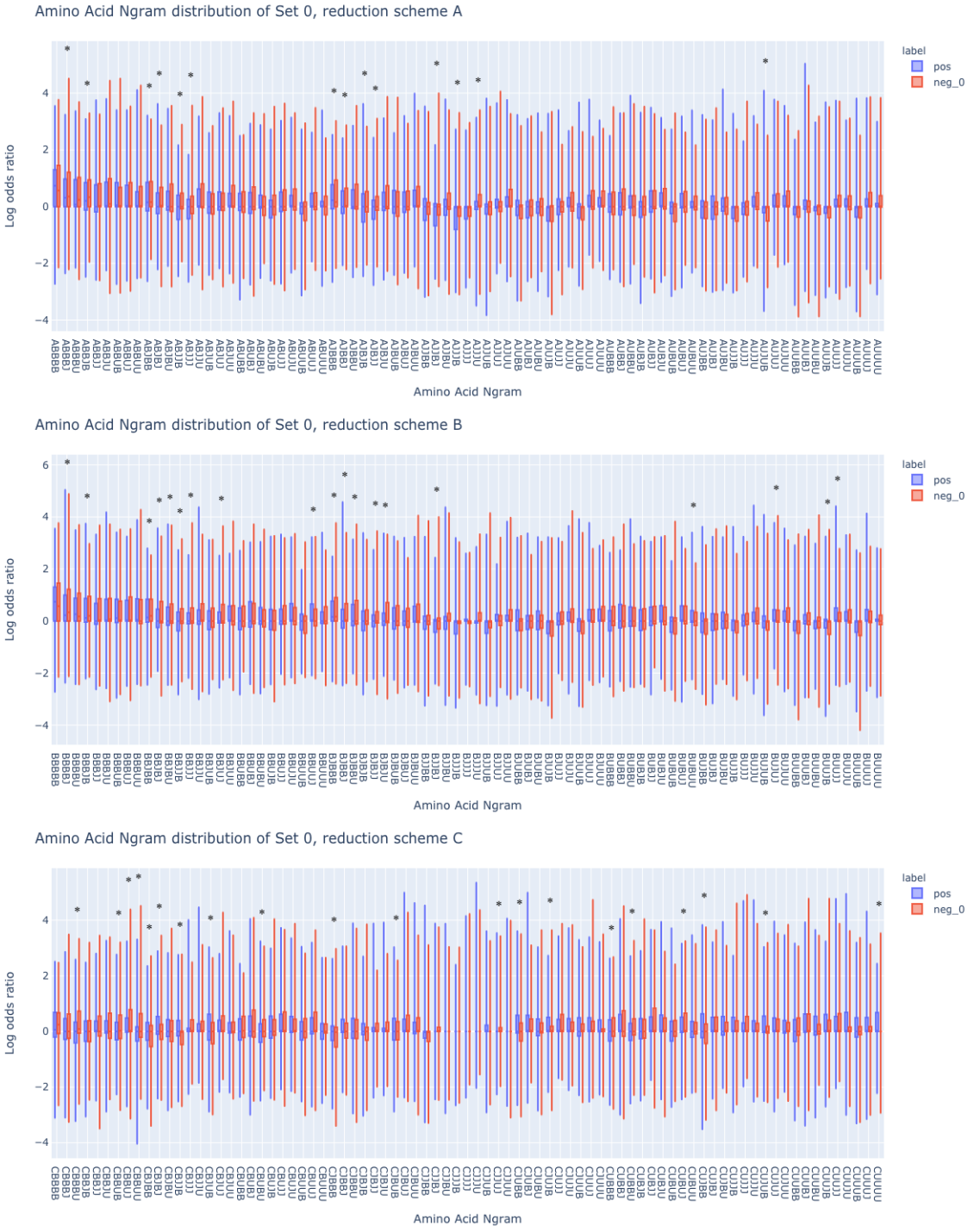


Figure 19 - Delaunay triangulates on a plane.

Simplexes were developed using the pyhull package, a python wrapper of qhull for the generation of Delaunay triangulations. The resulting 4-residue simplexes for each amino acid in the desired peptide were reduced using the designated alphabet and the log likelihood of each simplex was calculated in a manner similar to that used to calculate the log likelihoods for n-grams.[108] In the cases of peptides within the negative set, some residues contained within some simplexes may not be contained within the analyzed sequence. This is due to the use of sequence fragments for negative comparison. In this case, the likelihood of each simplex is calculated based on the entire peptide sequence, not just the section under analysis. Log odds ratios for each simplex in sample set 0 can be found in Figure 20. The data for all sample set is located at http://omics.gmu.edu/ssnaap/assets/supplemental.html


Amino Acid Ngram distribution of Set 0, reduction scheme A
Log odds ratio
Amino Acid Ngram
label
pos
neg_0
Amino Acid Ngram distribution of Set 0, reduction scheme B
Log odds ratio
Amino Acid Ngram
label
pos
neg_0
Amino Acid Ngram distribution of Set 0, reduction scheme C
Log odds ratio
Amino Acid Ngram
label
pos
neg_0

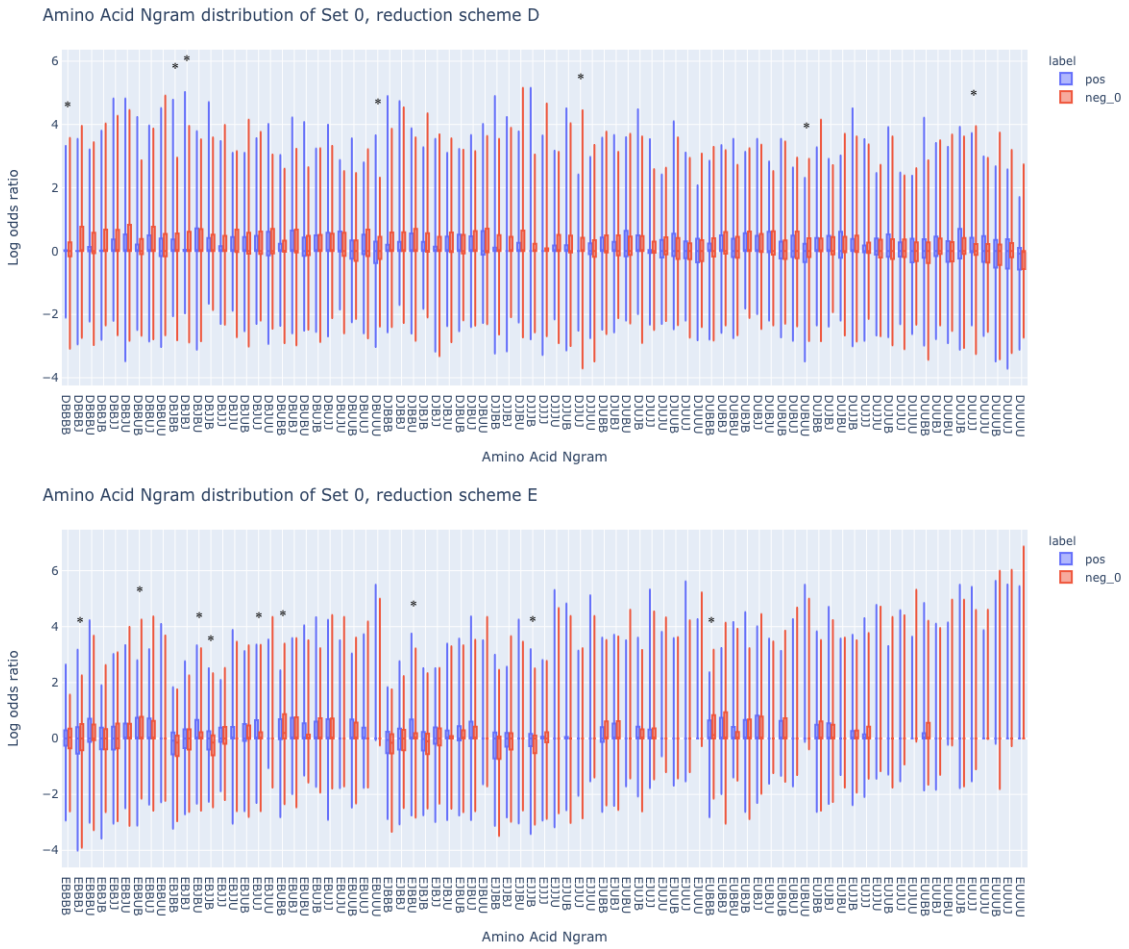


Figure 20 - Log odds ratios for Delaunay simplexes generated by each alphabet reduction, scheme sample set 0.

Results

Each of the resulting datasets is comprised of 5 reduced alphabets each alphabet contains 20 features describing the percentage of each amino acid in the analyzed sequence, 27 amino acid sequence features representing the 27 sequence based 3-grams, 27 secondary structure features representing the 27 3-grams based on the DSSP secondary structures and 81 tertiary structure features based on reduced sequence Delaunay simplexes

Amino acid variation

A preliminary analysis of the data shows that there are some significant differences between the positive and the negative data sets. For example, cystine is more prevalent in the positive peptides compared with the negative peptides, while both aspartic acid and glutamic acid are underrepresented in peptides with antimicrobial activity [Figure 21]. These results are similar to previously published results.[64,65]

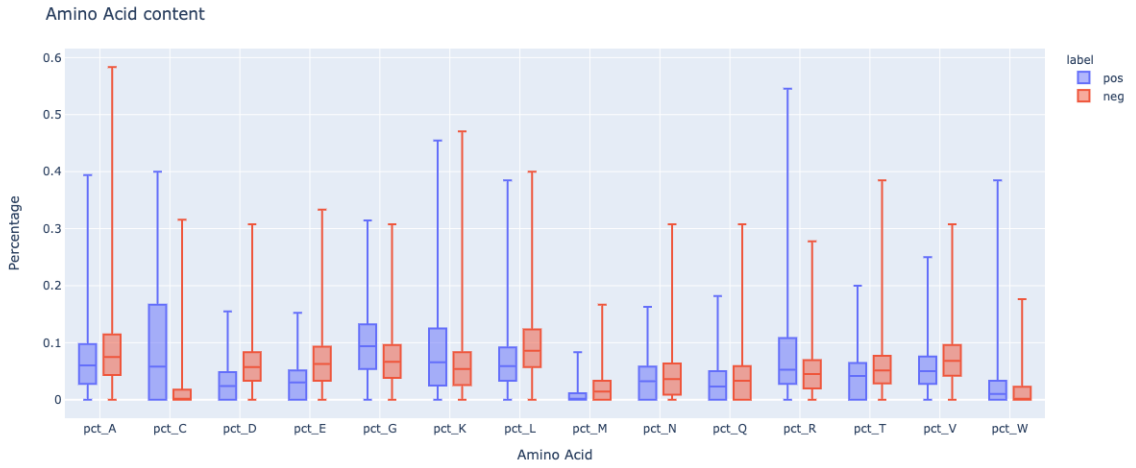


Figure 21 – Significant Amino acid variation in original sequences

This supports previous research that have also found high percentages of cystine in AMPs particularly plant peptides and anti-viral peptides.[109,110] The underrepresentation of Glutamic and Aspratic acid is likely due to the fact that most AMPs are positively charged and would, therefore, lack negatively charged amino acids.

Amino Acid n-gram Variation

Some differences were evident in the log odds ratios of n-grams [Figure 22].

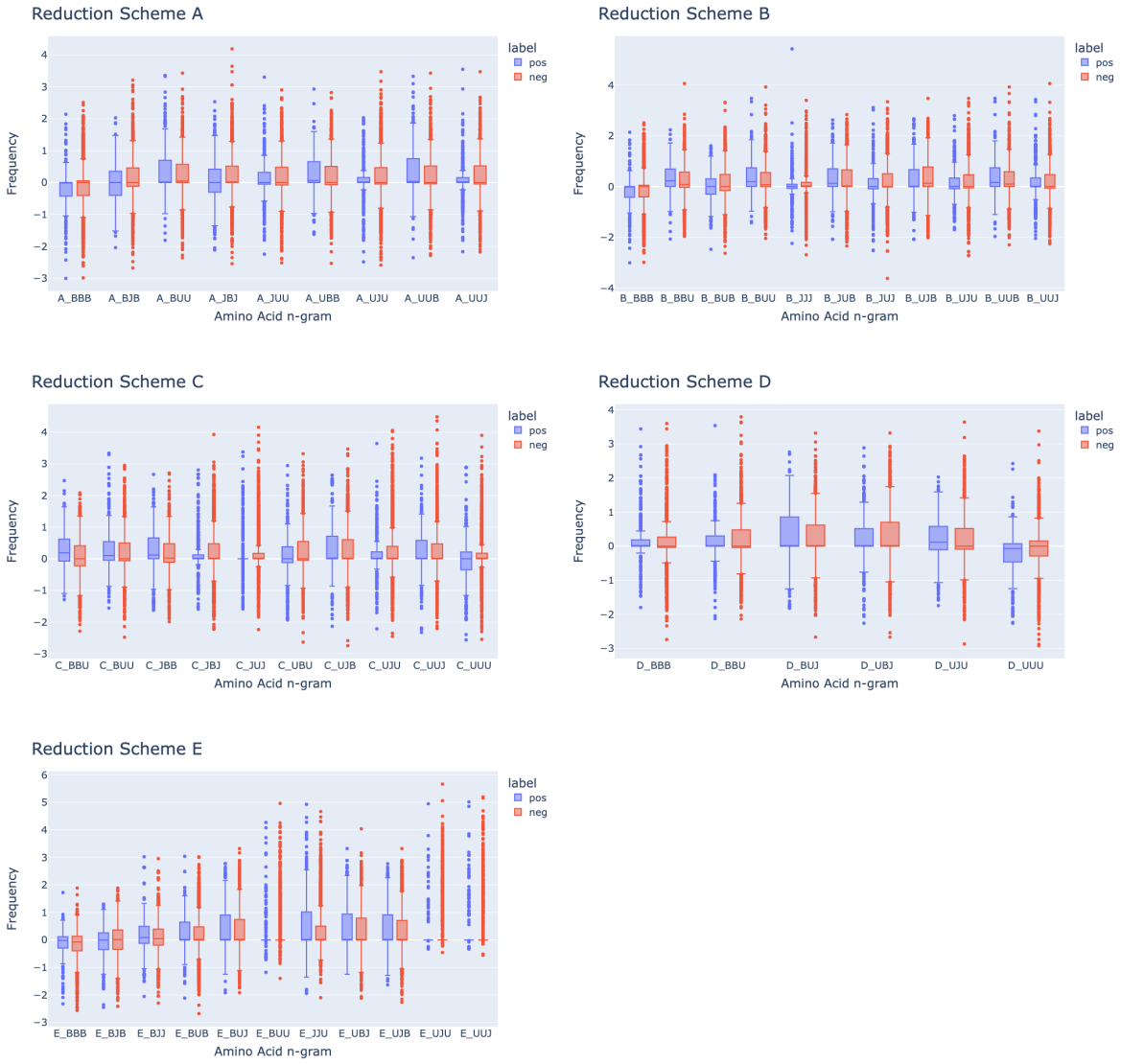


Figure 22 – Significantly different reduced alphabet n-gram sequences

Secondary Structure variation

Positive AMPs showed a slightly increased tendency toward coil structures as compared with negative peptides, which were comprised of more helical structures [Figure 23].

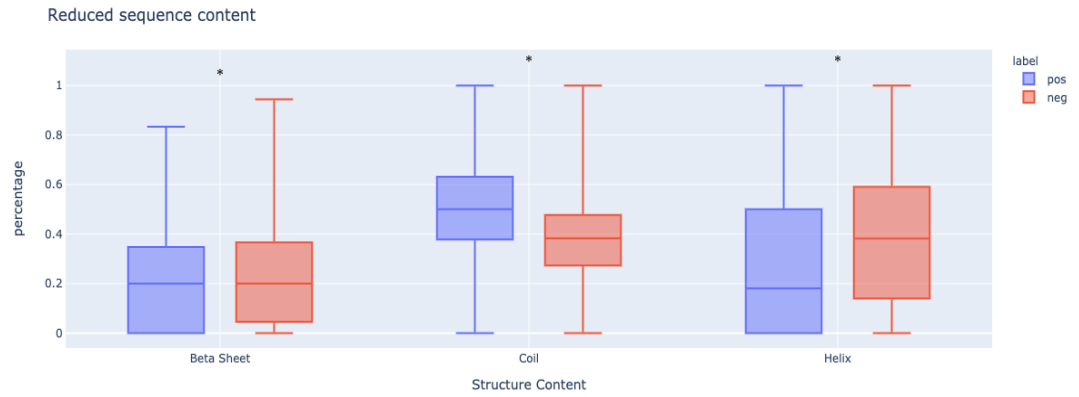


Figure 23 - Distribution of Structures

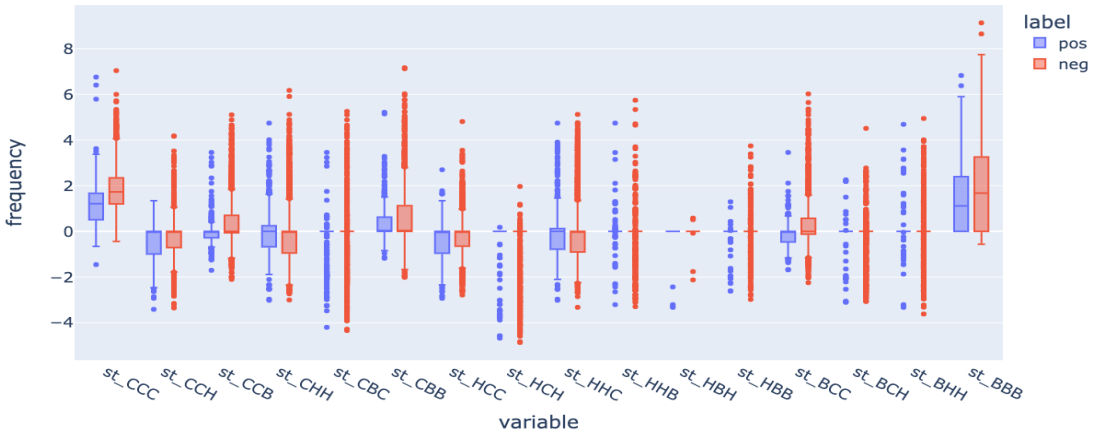


Figure 24 – Significantly different structure Ngrams of AMPs and non-AMP controls

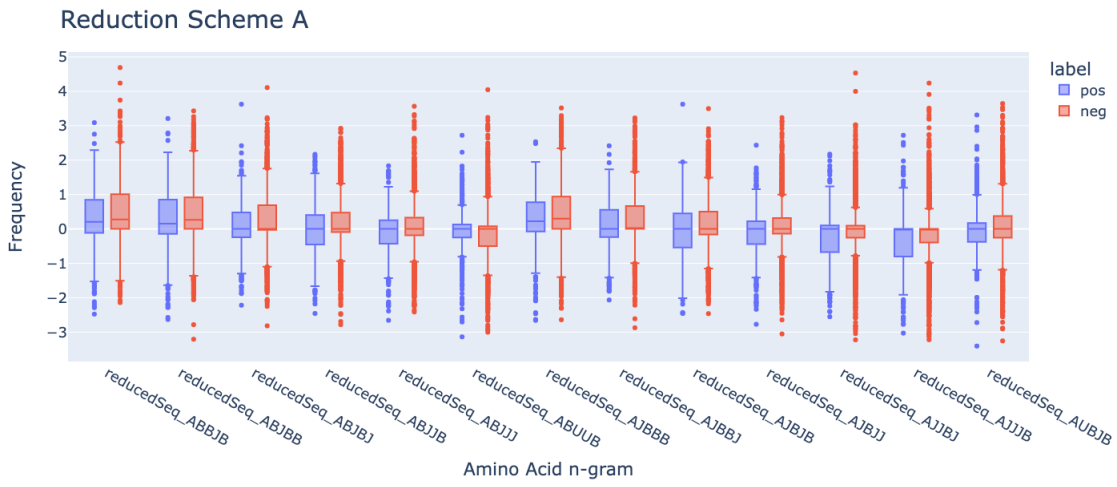
Structural n-gram variation

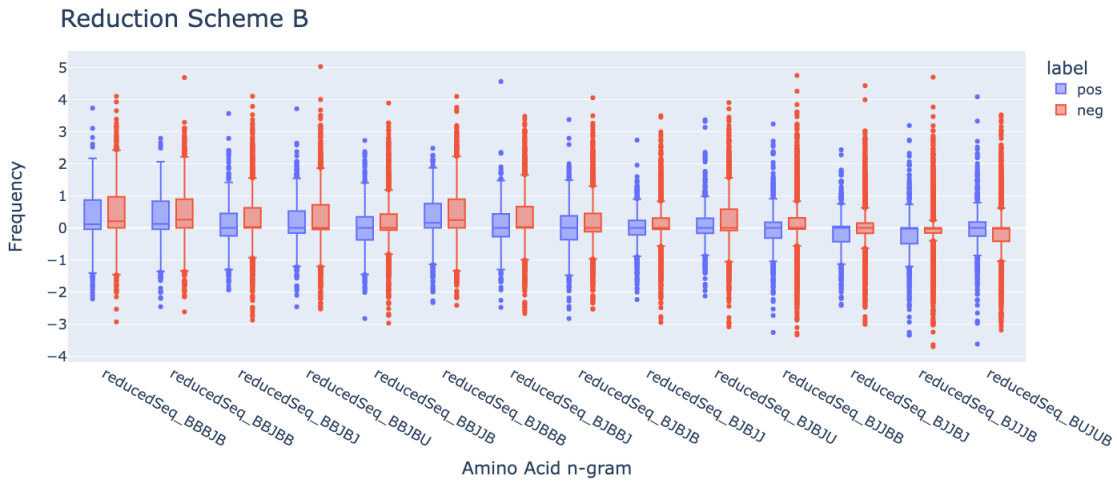
Structural n-grams show slightly more variation that a simple look at composition. Trigrams of all three structures appear more often that would be expect in both positive and negative samples, with negative samples presenting coil and beta sheet triplets with a significantly higher LOR than positive samples.

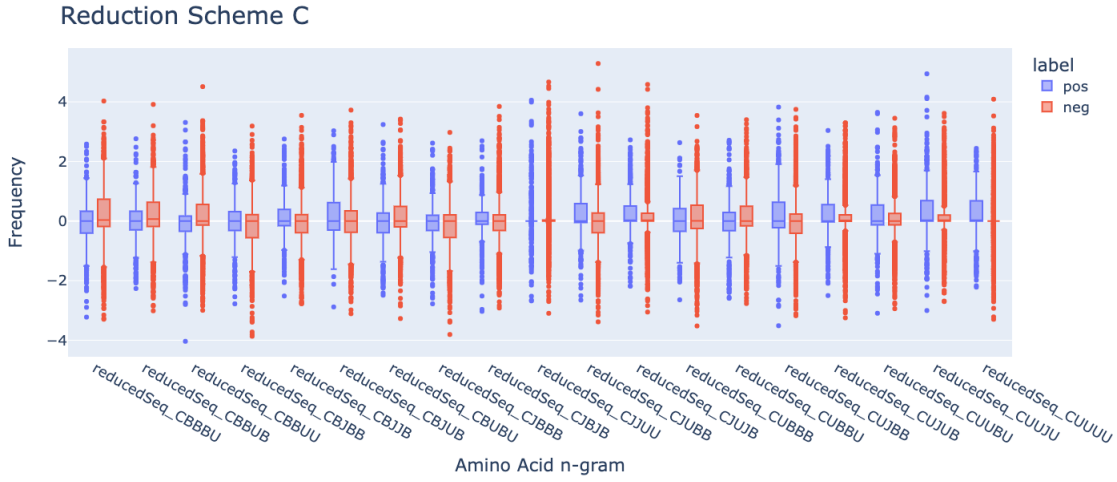
Tessellation Simplex Variations

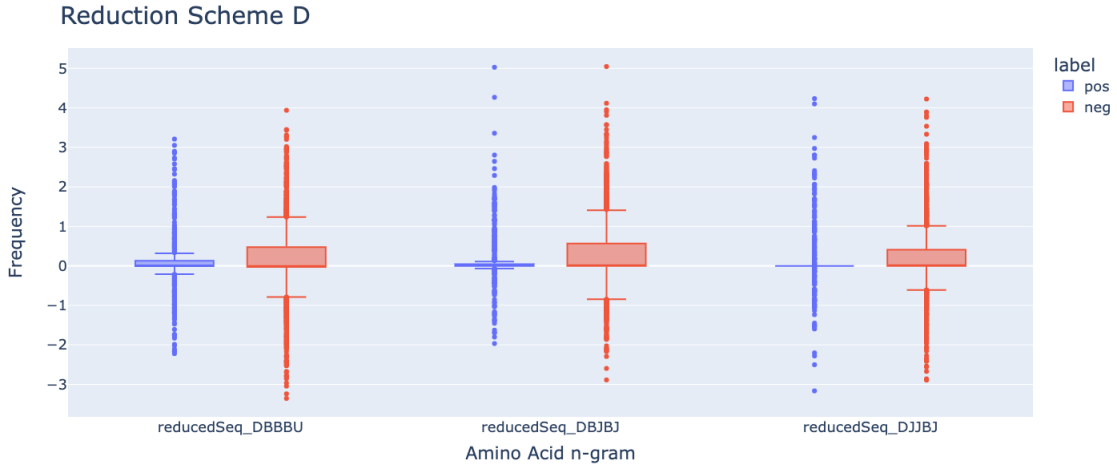
There was a great deal of variation between the different alphabet reduction schemes when it came to three dimensional structures represented by simplexes. Schemes A, B and C exposed a great deal of variation between the positive and the negative set with more than 10 simplexes being significantly different between the positive and negative samples [Figure 25].

Using both reduction scheme A and B we see a large difference in the number of simplexes composed of the B and J groups of amino acids, with AMPs generally exhibiting a higher LOR of these simplexes than the negative samples.




Reduction Scheme B
label
pos
neg
Frequency
5
4
3
2
1
0
−1
−2
−3
−4
reducedSeq_BBBJB
reducedSeq_BBJBB
reducedSeq_BBJBJ
reducedSeq_BBJBU
reducedSeq_BBJJB
reducedSeq_BJBBB
reducedSeq_BJBBJ
reducedSeq_BJBJB
reducedSeq_BJBJJ
reducedSeq_BJBJU
reducedSeq_BJJBB
reducedSeq_BJJBJ
reducedSeq_BJJJB
reducedSeq_BUJUB
Amino Acid n-gram


Reduction Scheme C
label
pos
neg
Frequency
4
2
0
−2
−4
reducedSeq_CBBBU
reducedSeq_CBBUB
reducedSeq_CBBUU
reducedSeq_CBJBB
reducedSeq_CBJJB
reducedSeq_CBJUB
reducedSeq_CBUBU
reducedSeq_CJBBB
reducedSeq_CJBJB
reducedSeq_CJJUU
reducedSeq_CJUBB
reducedSeq_CJUJB
reducedSeq_CUBBB
reducedSeq_CUBBU
reducedSeq_CUJBB
reducedSeq_CUJUB
reducedSeq_CUUBU
reducedSeq_CUUJU
reducedSeq_CUUUU
Amino Acid n-gram


Reduction Scheme D
label
pos
neg
Frequency
5
4
3
2
1
0
−1
−2
−3
reducedSeq_DBBBU
reducedSeq_DBJBJ
reducedSeq_DJJBJ
Amino Acid n-gram

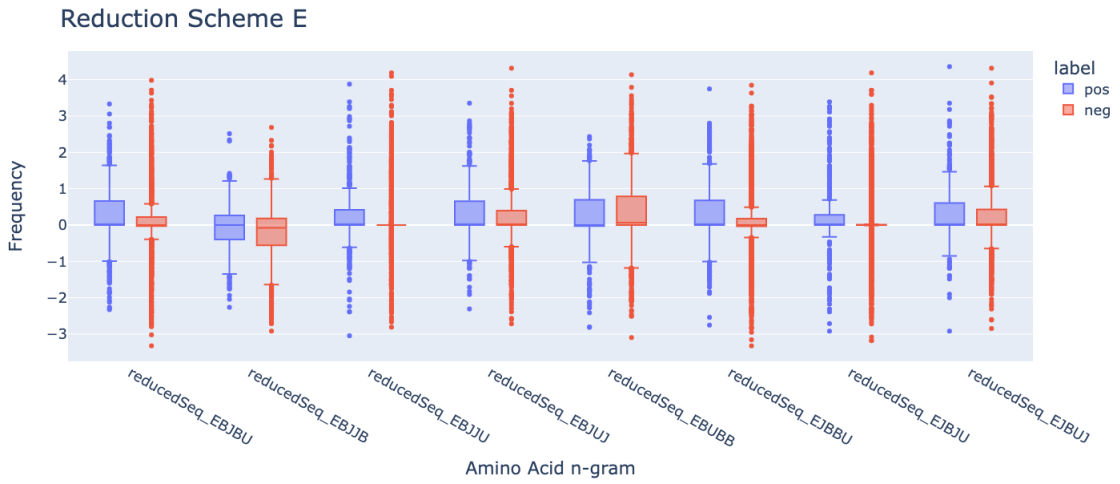


Figure 25 – Simplexes with a significant variation between positive and negative samples

Conclusions

The structural differences between peptides with and without antimicrobial activity indicate that there are some inherent differences between the tertiary structures of these peptides. In both reduction scheme A and B we see an increased likelihood of simplex composed primarily of the B type amino acid. The B group in both reduction schemes is composed of primarily hydrophobic residues. These pockets of hydrophobicity have been identified previously and are thought to increase the peptide's ability to interact with the bacterial cellular membrane.[111,112] However, a peptide with an overabundance of hydrophobic residues is more apt to for self-aggregation. This propensity is balanced by a distribution of positive charges at each terminus, limiting the aggregation to dimers.[113]

CHAPTER 5 – ALGORITHM DEVELOPMENT AND CLASSIFICATION PREDICTION

Introduction

This chapter will discuss the process of selecting and validating a machine learning algorithm with the intent of predicting amino acid sequences with a high probability of antimicrobial activity. Previous research has focused on a small number of potential algorithms such as Support Vector Machines, clustering algorithms and small neural networks (See **Error! Reference source not found.** for a complete listing) but have not taken more robust algorithms such as boosted ensemble algorithms and multilayer neural networks into consideration. In this work we will evaluate the algorithms listed above as well as several ensemble and deep learning algorithms.

Methods

Algorithm selection

Each algorithm was screened using the default optimization settings with a ten-fold cross validation. Accuracy was calculated by dividing the number of correct predictions by the total number of instances. Mean accuracies for algorithms across all reduction schemes can be seen in Figure 26. The ten algorithms with the highest accuracy are presented in **Error! Reference source not found.**.

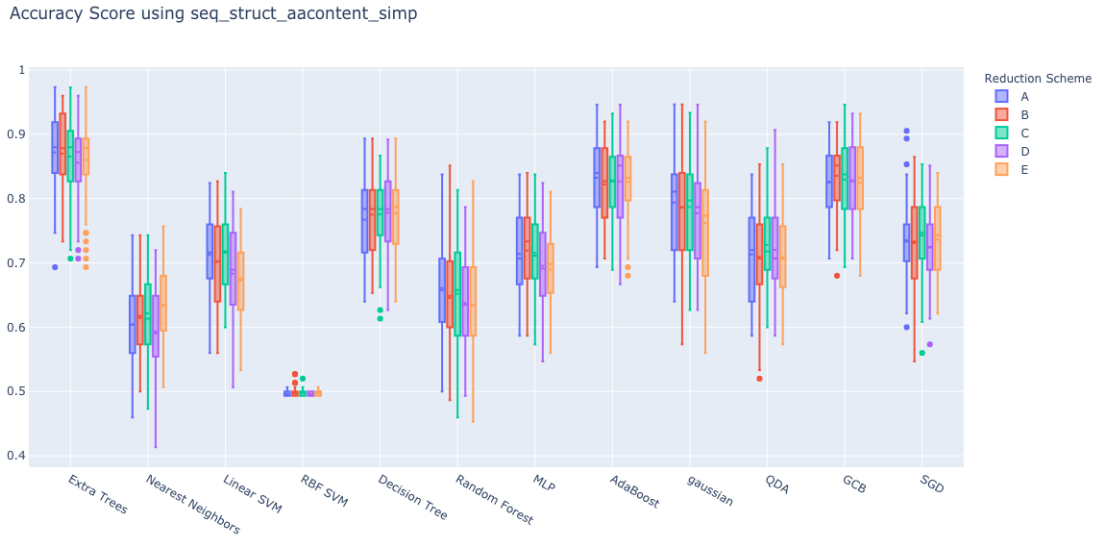


Figure 26 - Accuracy scores for each of the prediction algorithms tested. Algorithms were tested using the default optimization parameters.

Table 3 - Models and reduction schemes resulting in highest accuracy

Model Number	Classifier	Sequence Reduction Scheme	Mean Accuracy (%)	Median
1	**Extra Trees**	E	0.879 ± 0.07	0.88
2	**Extra Trees**	D	0.878 ± 0.07	0.88
3	**Extra Trees**	A	0.872 ± 0.06	0.87
4	**Extra Trees**	B	0.867 ± 0.06	0.87
5	**Extra Trees**	C	0.867 ± 0.08	0.87
6	**GBC**	E	0.851 ± 0.07	0.85
7	**GBC**	B	0.845 ± 0.05	0.84
8	**AdaBoost**	E	0.839 ± 0.06	0.84
9	**AdaBoost**	A	0.839 ± 0.05	0.84
10	**GBC**	A	0.833 ± 0.04	0.83

Model optimization

Each the hyperparameter for each model was optimized using a grid search. The hyperparameters and the search space for each parameter used in the grid search are

found in **Error! Reference source not found.**. A grid search systematically iterates over a multidimensional space of hyper parameters, fitting the model to each combination. The grid search only progresses for a set number of iterations, usually not enough to allow the model to find convergence. This allows the grid search to iterate over a very large search space in a reasonable time frame.

Table 4 - Hyperparameters and Search space for model optimization

Classifier	Hyperparameter	Search space
Extra Trees	N_estimators	[50,100,200],
	depth	[1,3,9]
GBC	N_estimators	[50,100,200]
	Learning_rate	[0.5, 1.0, 1.5]
	Max_depth	[1,3,9]
AdaBoost	N_estimators	[50, 100,200]
	Learning_rate	[0.5, 1.0, 1.5]
	Algorithm	['SAMME', SAMME.R']
Gaussian NB	Var_smoothing	[10^{-7}, 10^{-9}, 10^{-11}]

Once the optimal parameters have been found we can fully train a model using the identified parameters [**Error! Reference source not found.**]. These models do not appear to be significantly more accurate than their unoptimized counterparts however, this optimization may prove beneficial for feature reduction [

Figure 27].

Table 5 - Optimized Hyperparameters for each Model

Model Number	Classifier	Sequence Reduction Scheme	Optimized Hypterparameters	Mean Accuracy(%)	Median Score
1	**Extra Trees**	E	N_estimators: 100	0.86 ± 0.06	0.88

			Max_depth: 9		
2	**Extra Trees**	D	N_estimators: 200 Max_depth: 9	0.85 ± 0.07	0.86
3	**Extra Trees**	A	N_estimators: 200 Max_depth: 3	0.84 ± 0.07	0.85
4	**Extra Trees**	B	N_estimators: 100 Max_depth: 9	0.86 ± 0.06	0.88
5	**Extra Trees**	C	N_estimators: 100 Max_depth: 9	0.85 ± 0.06	0.86
6	**GBC**	E	N_estimators: 50 Max_depth: 1 Learning_rate:1.0	0.82 ± 0.06	0.83
7	**GBC**	B	N_estimators: 50 Max_depth:3 Learning_rate:0.5	0.86 ± 0.05	0.86
8	**AdaBoost**	E	N_estimators: 50 Learning_rate:1.0 Algorithm: Samme.r	0.80 ± 0.07	0.81
9	**AdaBoost**	A	N_estimators: 210 Learning_rate:1.0 Algorithm: Samme.r	0.84 ± 0.06	0.84
10	**GBC**	A	N_estimators: 50 Max_depth: 3 Learning_rate:1.5	0.83 ± 0.06	0.83

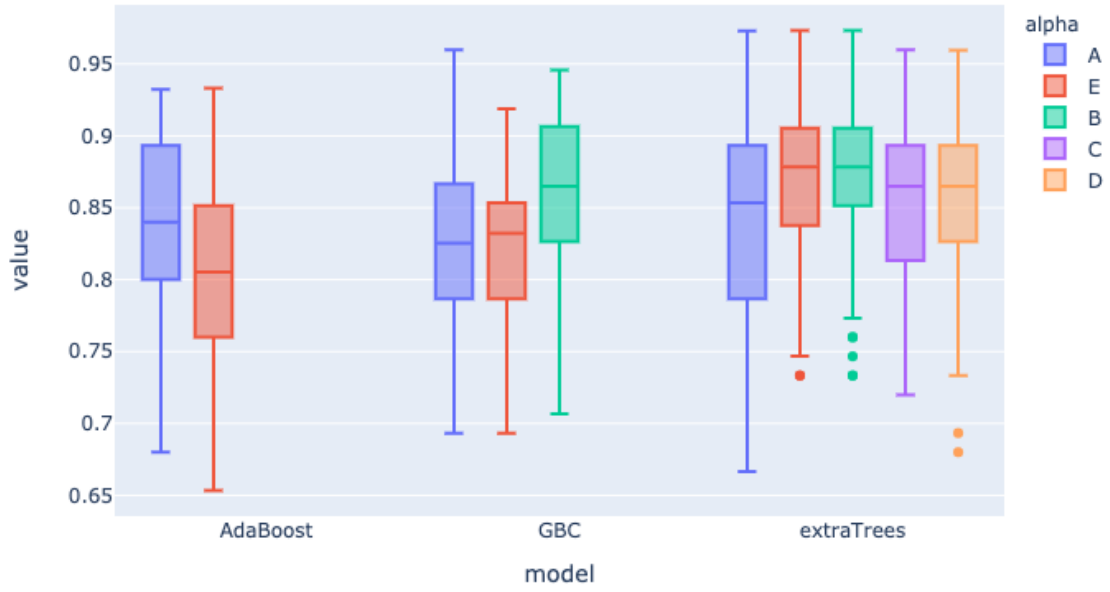
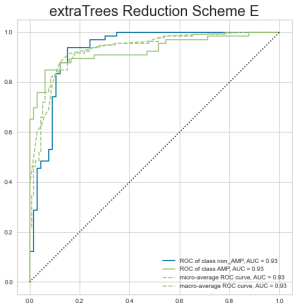
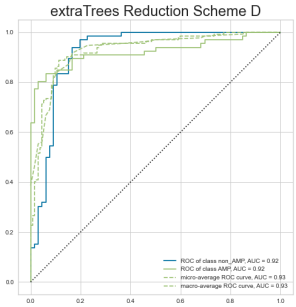
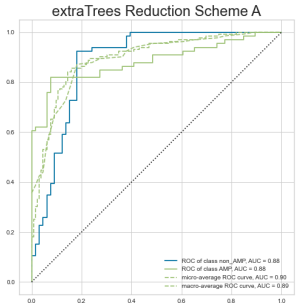
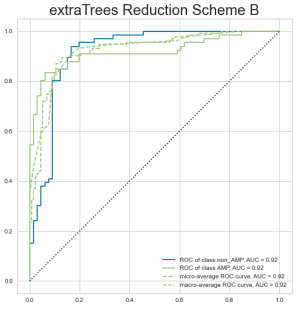


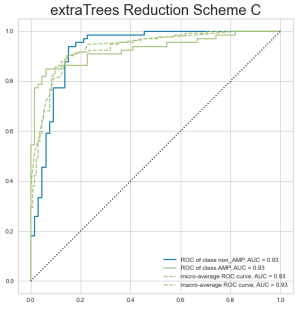
Figure 27 - Accuracy of optimized models

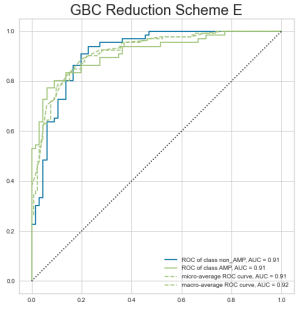

extraTrees Reduction Scheme E
ROC of class non_AMP, AUC = 0.93
ROC of class AMP, AUC = 0.93
micro-average ROC curve, AUC = 0.93
macro-average ROC curve, AUC = 0.93

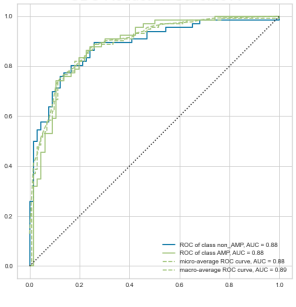

extraTrees Reduction Scheme D
ROC of class non_AMP, AUC = 0.92
ROC of class AMP, AUC = 0.92
micro-average ROC curve, AUC = 0.93
macro-average ROC curve, AUC = 0.93

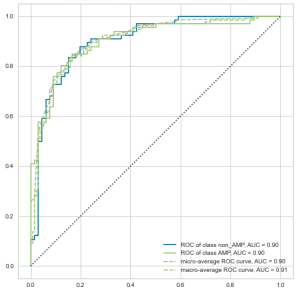

extraTrees Reduction Scheme A
ROC of class non_AMP, AUC = 0.88
ROC of class AMP, AUC = 0.88
micro-average ROC curve, AUC = 0.90
macro-average ROC curve, AUC = 0.89

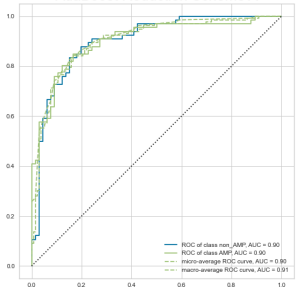

extraTrees Reduction Scheme B
ROC of class non_AMP, AUC = 0.92
ROC of class AMP, AUC = 0.92
micro-average ROC curve, AUC = 0.92
macro-average ROC curve, AUC = 0.92


extraTrees Reduction Scheme C
ROC of class non_AMP, AUC = 0.93
ROC of class AMP, AUC = 0.93
micro-average ROC curve, AUC = 0.93
macro-average ROC curve, AUC = 0.93


GBC Reduction Scheme E
ROC of class non_AMP, AUC = 0.91
ROC of class AMP, AUC = 0.91
micro-average ROC curve, AUC = 0.91
macro-average ROC curve, AUC = 0.92


GBC Reduction Scheme A
ROC of class non_AMP, AUC = 0.88
ROC of class AMP, AUC = 0.88
micro-average ROC curve, AUC = 0.88
macro-average ROC curve, AUC = 0.89


AdaBoost Reduction Scheme A
ROC of class non_AMP, AUC = 0.90
ROC of class AMP, AUC = 0.90
micro-average ROC curve, AUC = 0.90
macro-average ROC curve, AUC = 0.91


AdaBoost Reduction Scheme A
ROC of class non_AMP, AUC = 0.90
ROC of class AMP, AUC = 0.90
micro-average ROC curve, AUC = 0.90
macro-average ROC curve, AUC = 0.91

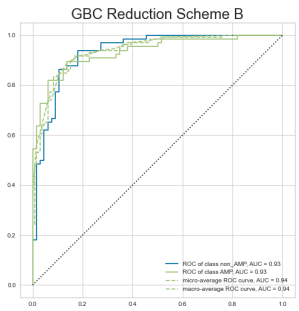


Figure 28 - AUC graphs for each of the optimized models

Conclusion

Previous classification efforts have focused on individual classifiers such as SVM and a number of simple neural networks. I have found that the use of an ensemble, boosted algorithm performs with a higher rate of accuracy than many of these previous algorithms with the ExtraTrees Classifier algorithm performing with an accuracy in the range of 86 - 88% and an AUC greater than 0.96 across all AMP types. This is comparable or superior to most previous work, especially considering that the negative set was paired for sequence and structure similarity and that no attempt was made to pre-define the AMPs by their target organisms. Proceeding chapters will describe efforts to refine the algorithms as well as to identify feature within the AMP that may lead to the identified activity. Further work has been done to identify algorithms that accurately identify specific classes of AMPs such as those with anti-bacterial or anti-parasitic activity.

CHAPTER 6 – IDENTIFICATION OF RELEVENT FEATURES

Introduction

All peptides are composed of amino acids in a specific sequence and spatial conformation. It is this sequence and three-dimensional structure that determine the functionality of the resulting peptide. One of the goals of this research is to identify specific features that are likely to result in antimicrobial activity. The identification of such features can lead to more successful identification of naturally occurring antimicrobial peptides as well as inform the development of future, synthetic AMPs. Initial optimization was done using all of the available features. However, this does little to elucidate the importance of individual features for antimicrobial activity. In an attempt to identify the features most responsible for antimicrobial activity an attempt at feature selection was made.

Feature selection should not be confused with dimensionality reduction, though they do both endeavor to reduce the number of attributes in a dataset. Dimensionality reduction, however, is the process of recombining attributes into new features through process such as Principle Component Analysis (PCA), which takes advantage of the co-linearity of features to reduce dimensionality. While often successful, dimensionality reduction techniques to not preserve the information inherent within the features

themselves rendering it less useful if the goal is to understand the structure of a peptide, rather than simply predict its classification.

Feature selection, also known as variable or attribute selection, is the process of selecting a subset of the data that is most relevant to the prediction problem at hand. Feature selection can be used to remove unneeded or redundant features from the dataset. The remaining features will be those that are most useful in distinguishing an AMP from a non-AMP and by extension, presumably the features that endow the peptide with antimicrobial activity.

There are additional benefits to feature selection when it pertains to the ability of a classifier to make an accurate prediction. A large number of irrelevant features has the effect of forcing the classifier to work harder to identify feature that discriminate between the positive and the negative set. This increases computation time and can obscure important variations making predictions less accurate and can increase overfitting because the classifier makes more decisions based on irrelevant background noise.

There are a number of feature extraction techniques commonly used in machine learning. Wrapper methods treat the feature reduction problem as a search problem. Different combinations of features are generated, evaluated and compared to other combinations to identify the most accurate subset. A rudimentary wrapper type search is undertaken in this research using combinations of the features based on the source of the data. The Recursive Feature Elimination (RFE) algorithm is another type of wrapper method in which the feature with the lowest rank based on a weighted ranking critera is removed over each iteration. However, the resulting feature set (F_m) does not necessarily

represent the individually most relevent features, but an optimal subset when taken together.[114]

Methods

Isolation of feature sets

The first, rudimentary, attempt was done by using predefined subsets of the input data to train the ExtraTrees, GBC and Adaboost algorithms. The subsets were defined based on the method of data extraction. Amino acid percentages, sequence n-grams, structure n-grams and simplex n-grams were each identified as a 'subset'. All possible combinations of the subsets were considered. Each subset combination was used to train and test one of the three most successful classification algorithms (AdaBoost, GBC, and ExtraTrees classifiers). It was assumed that if a reduced feature set lead to increased accuracy in AMP prediction that the features contained within where, in some way, relevant to antimicrobial activity.

Univariant selection

Univariant selection uses a function to select the features most closely related to the desired output. Before feature selection, nonvariant features were removed. Using the SelectKBest function contained within the sklearn feature selection module a series of K features with the highest ANOVA f-value between the label and the feature were selected. Four K values [15, 25, 35, and 50] were evaluated for optimal accuracy. Each of the resulting features sets were used as the training set for one of the three previously defined classification algorithms.

Recursive Feature Elimination

In recursive feature elimination a prediction model is built using the desired classifier algorithm. The features are then ranked by their importance to the model. The least predictive feature is dropped, and the model is rebuilt using the remaining features. This process continues in an iterative manner until the desired number of features remains. The resulting features are assumed to be those with the greatest relevance to the prediction.

Top Algorithm selection

The most accurate algorithm was selected and fully trained using the identified feature set. This model was then used for subsequent feature analysis.

Results

Feature subsets

The mean accuracies for each combination of subsets for three classification algorithms is shown in Figure 29.

Figure 29 - Accuracy distribution of models with features selected by feature subsets

The models resulting in the top five accuracies are detailed in **Error! Reference source not found.**. The score distributions for the top 5 models are show in Figure 30

Table 6 - Accuracies for feature selection using subset Feature Selection

Model	Reduction Scheme	Features	Mean	Median
ExtraTrees	E	Sequence, Structure & Amino Acid Content	0.88±0.06	0.89
ExtraTrees	D	Sequence, Structure & Amino Acid Content	0.87±0.04	0.89
ExtraTrees	A	Sequence, Structure & Amino Acid Content	0.88±0.06	0.90
ExtraTrees	B	Amino Acid Content	0.88±0.05	0.89
ExtraTrees	C	Structure & Amino Acid Content	0.88±0.05	0.89
GBC	E	Sequence, Structure & Amino Acid Content	0.84±0.06	0.85
GBC	B	All Features	0.84±0.05	0.85
AdaBoost	E	Structure & Amino Acid Content	0.84±0.05	0.85
AdaBoost	A	Structure & Amino Acid Content	0.84±0.05	0.85
GBC	A	Structure & Amino Acid Content	0.84±0.05	0.84

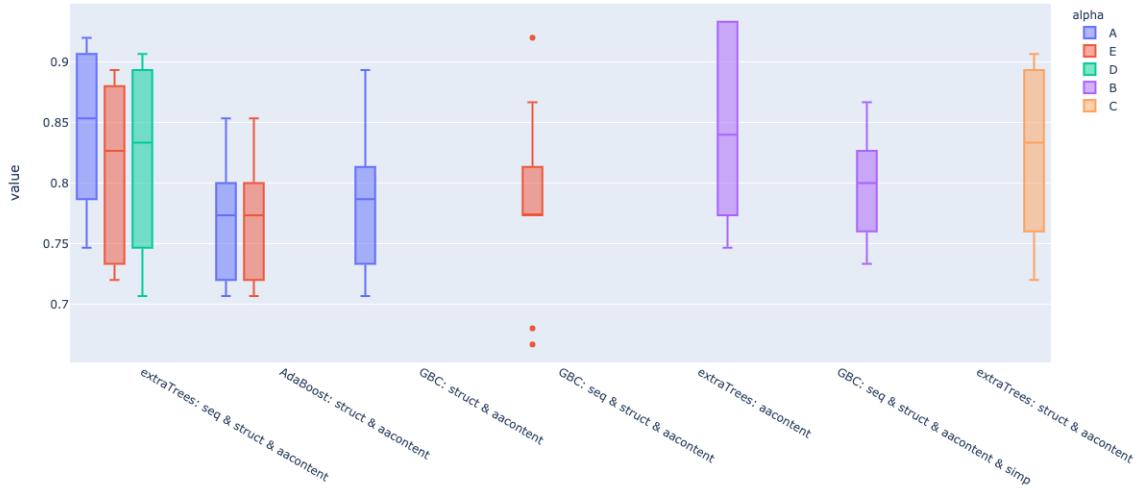
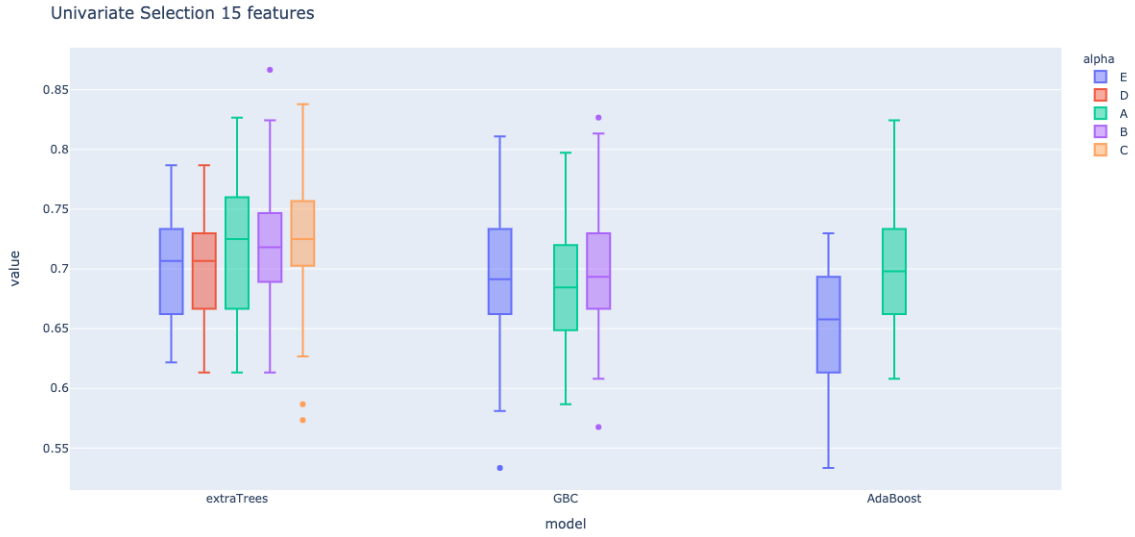
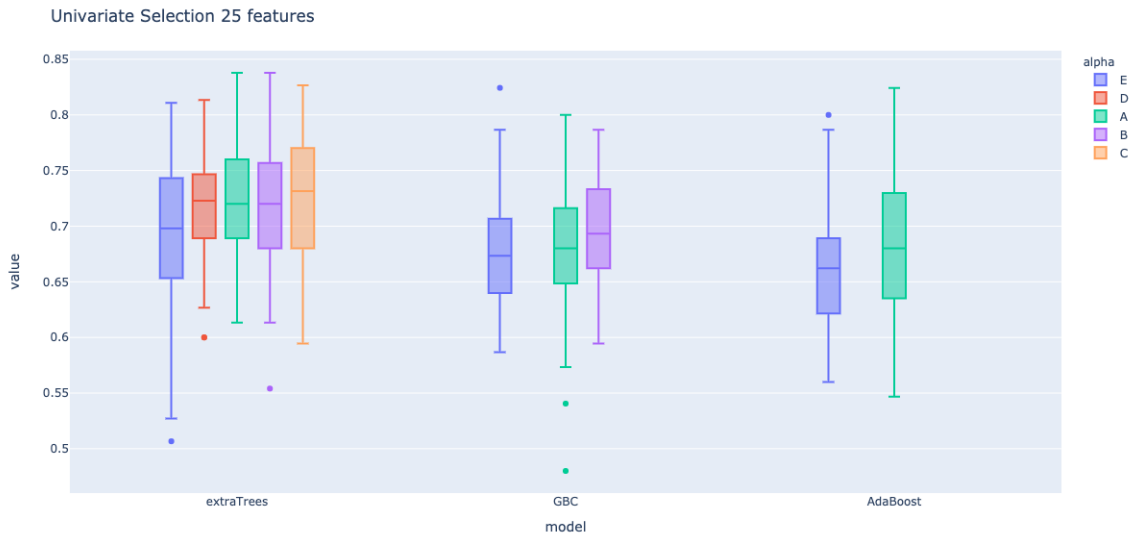


Figure 30 – Score distribution for the most accurate models

Univariant selection

The accuracy distributions for each model with either 15, 25, 35 or 50 selected features can be found in Figure 31 and detailed in **Error! Reference source not found.**.


Univariate Selection 15 features
alpha
E
D
A
B
C
value
0.85
0.8
0.75
0.7
0.65
0.6
0.55
extraTrees
GBC
AdaBoost
model


Univariate Selection 25 features
alpha
E
D
A
B
C
value
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
extraTrees
GBC
AdaBoost
model

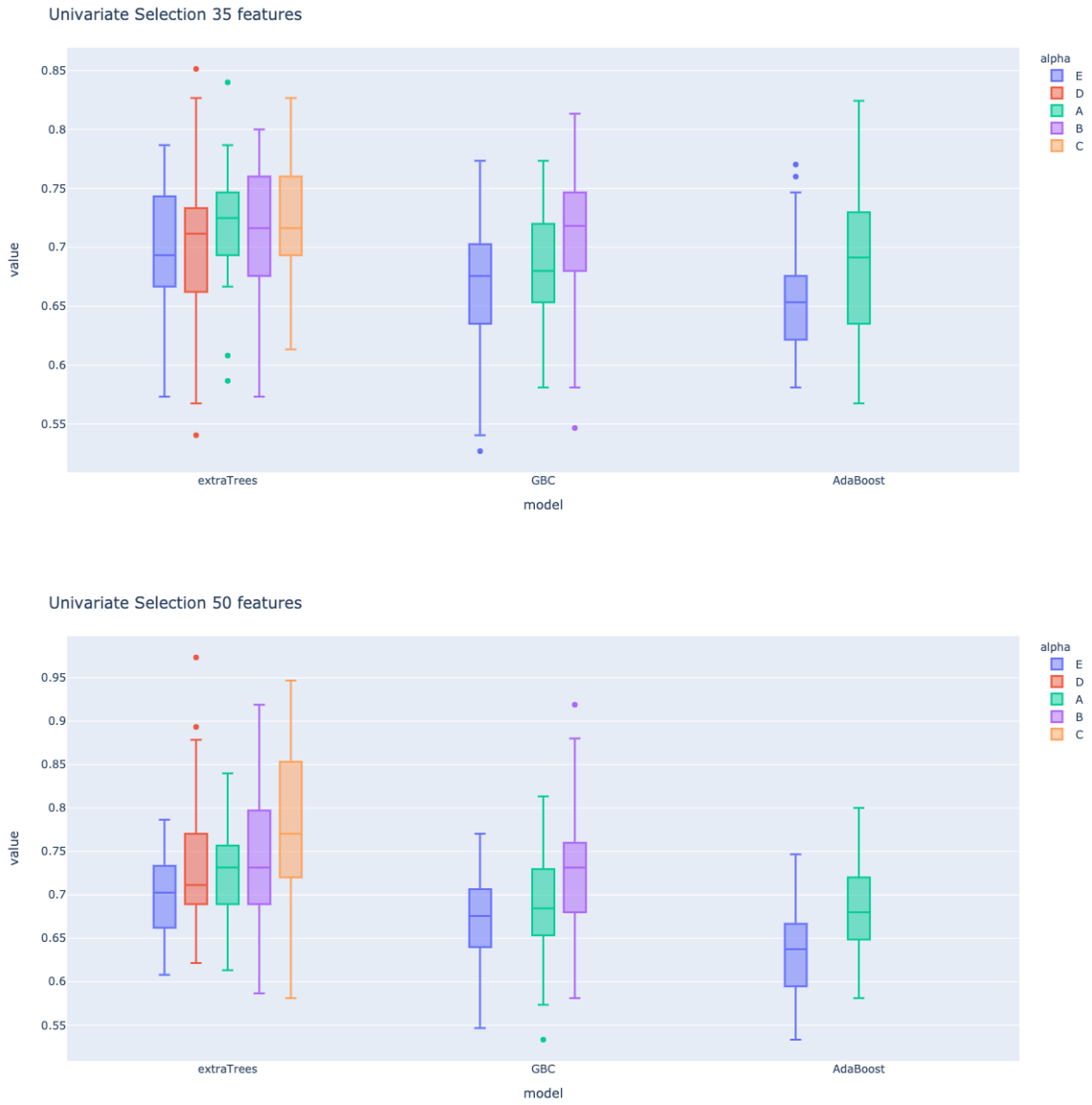


Figure 31 - Accuracy distribution of models with features selected by univariant selection

Table 7 - Accuracies for feature selection using Univariant Feature Selection

Model	Reduction Scheme	15 features		25 features		35 features		50 features	
		Mean	Median	Mean	Median	Mean	Median	Mean	Median
ExtraTrees	E	0.70±0.04	0.71	0.70±0.07	0.70	0.70±0.05	0.69	0.70±0.05	0.70
ExtraTrees	D	0.70±0.04	0.71	0.71±0.04	0.72	0.71±0.06	0.71	0.73±0.07	0.71
ExtraTrees	A	0.72±0.06	0.72	0.72±0.05	0.72	0.72±0.04	0.72	0.73±0.05	0.73
ExtraTrees	B	0.72±0.05	0.72	0.72±0.05	0.72	0.71±0.06	0.72	0.75±0.08	0.73
ExtraTrees	C	0.72±0.05	0.72	0.73±0.06	0.73	0.72±0.05	0.72	0.78±0.09	0.77
GBC	E	0.69±0.06	0.69	0.68±0.05	0.67	0.67 ± 0.06	0.68	0.67±0.05	0.68
GBC	B	0.70±0.05	0.69	0.70±0.05	0.69	0.72 ± 0.05	0.71	0.74±0.08	0.73
AdaBoost	E	0.70±0.05	0.69	0.70±0.05	0.69	0.72 ± 0.05	0.71	0.74±0.08	0.73
AdaBoost	A	0.65±0.05	0.66	0.66±0.05	0.66	0.65 ± 0.05	0.66	0.63±0.05	0.64
GBC	A	0.69±0.05	0.68	0.68±0.08	0.68	0.68 ± 0.05	0.68	0.69±0.05	0.68

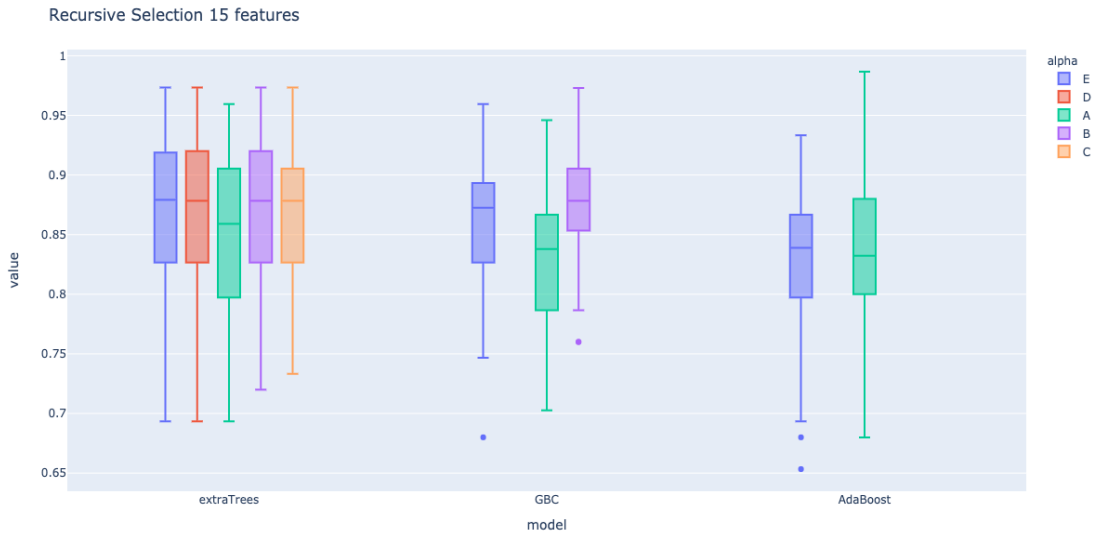
There were 4 features with a standard deviation of 0 for all reduction schemes. These features were removed as they added no discriminatory information to the dataset. (these features were all secondary structure features and therefore did not vary from one alphabet reduction scheme to the next. [sCHC, sCHB, sBHC, sBHB]).

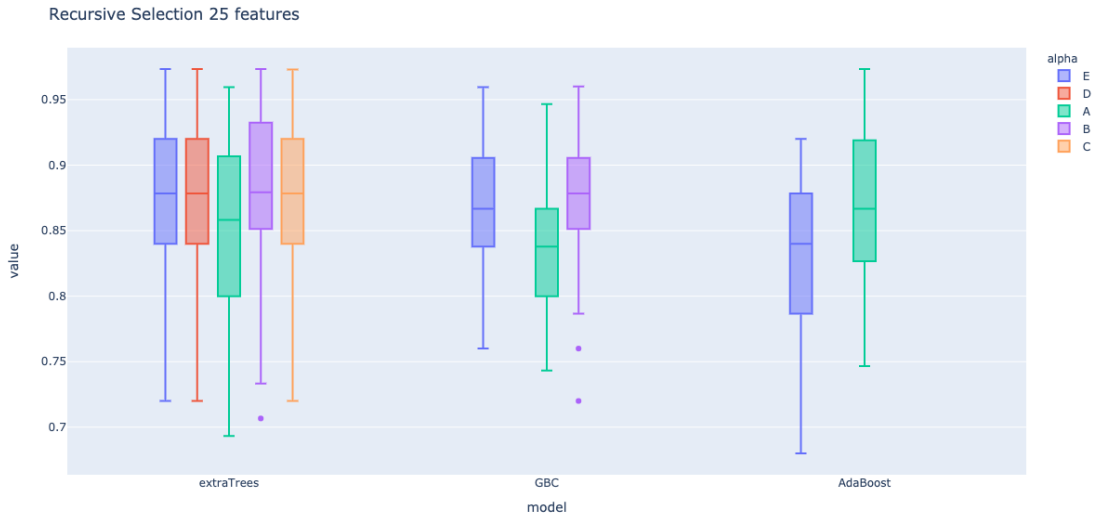
While the features selected varied between the different alphabet reduction schemes, several amino acid content features were selected for each reduction scheme. Amino acid percentages for C, E Q, M, and D were repeatedly within the features selected. A complete listing of the features for each feature reduction scheme can be found at http://omics.gmu.edu/ssnaap/assets/supplemental.html. If we consider only the data used for the 5 most accurate prediction models we find that this technique does not produce results that are statistically more accurate than either using the entire data set or selecting basic subsets [**Error! Reference source not found.**].

Recursive Feature Selection

Recursive feature selection resulted in the highest accuracies of all of the feature selection schemes. The top ten models are detailed in **Error! Reference source not**

found.. Recursive elimination produced the most accurate results. Once again, the amino acid composition features were well represented among the selected features, as well as features representing the three-dimensional structure. A complete listing of the features selected is found at http://omics.gmu.edu/ssnaap/assets/supplemental.html





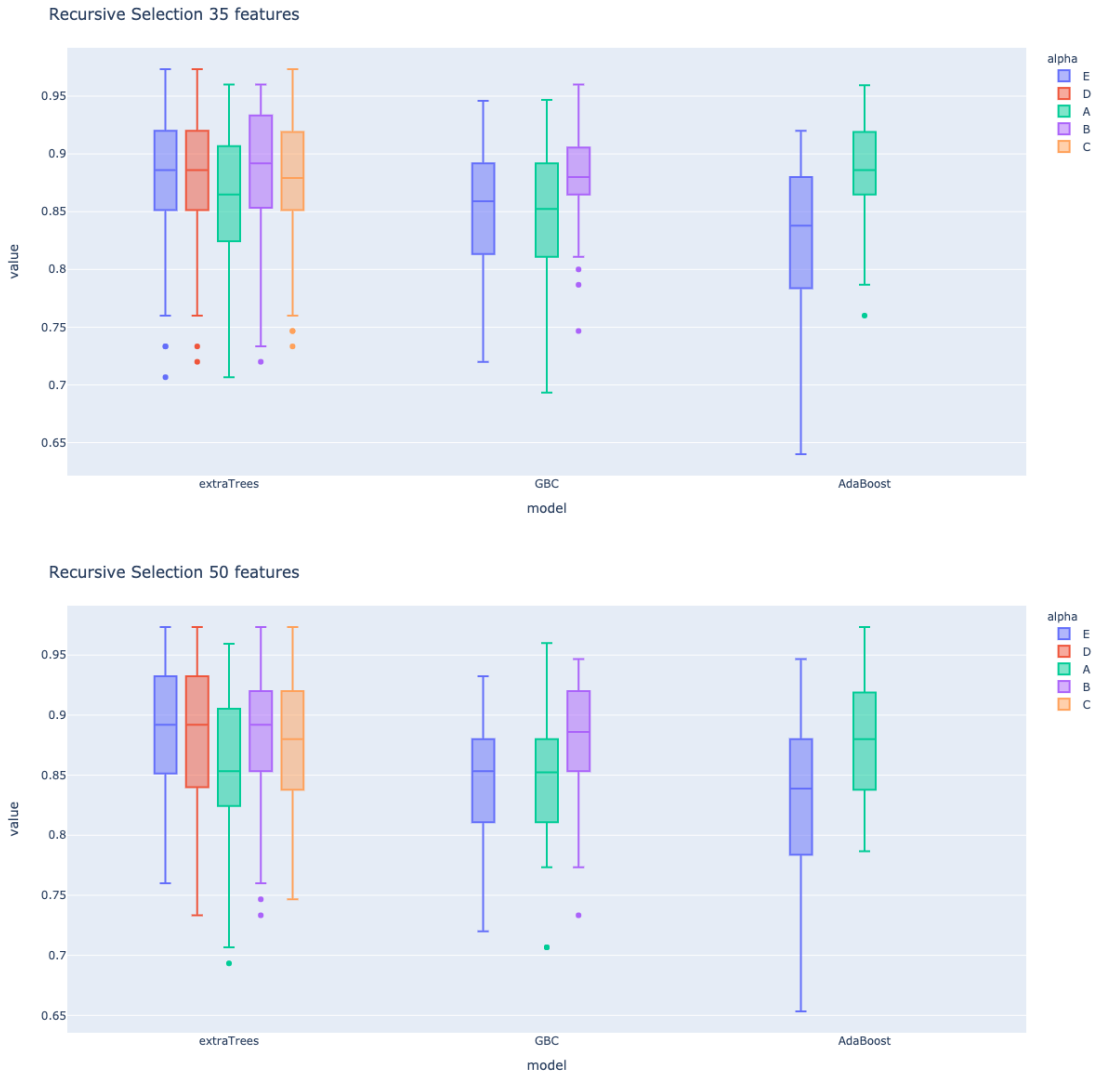


Figure 32 - Accuracy distribution of models with features selected by recursive selection

Table 8 - Accuracies for feature selection using Recursive Feature Elimination

Model	Reduction Scheme	15 features		25 features		35 features		50 features	
		Mean	Median	Mean	Median	Mean	Median	Mean	Median
ExtraTrees	E	0.86±0.07	0.88	0.87±0.06	0.88	0.87±0.06	0.89	0.88±0.06	0.89
ExtraTrees	D	0.86±0.07	0.88	0.87±0.06	0.88	0.88±0.06	0.89	0.88±0.06	0.89
ExtraTrees	A	0.84±0.07	0.86	0.85±0.07	0.86	0.85±0.07	0.86	0.85±0.07	0.85
ExtraTrees	B	0.86±0.07	0.88	0.88±0.08	0.88	0.88±0.07	0.89	0.88±0.06	0.89
ExtraTrees	C	0.87±0.07	0.88	0.88±0.06	0.88	0.87±0.06	0.88	0.88±0.06	0.88
GBC	E	0.86±0.06	0.87	0.68±0.05	0.87	0.85 ± 0.06	0.86	0.84±0.06	0.85
GBC	B	0.88±0.04	0.88	0.87±0.05	0.88	0.88 ± 0.04	0.88	0.88±0.05	0.89
AdaBoost	E	0.83±0.07	0.84	0.83±0.06	0.84	0.83 ± 0.07	0.84	0.83±0.07	0.84
AdaBoost	A	0.84±0.06	0.83	0.87±0.05	0.87	0.88 ± 0.05	0.89	0.88±0.05	0.88
GBC	A	0.83±0.06	0.84	0.84±0.05	0.84	0.84 ± 0.05	0.85	0.85±0.06	0.85

Conclusions

Feature subsets

All 4 of the feature subsets appear in one of the most accurate models. However, amino acid content appears in every model, while tertiary structure based simplexes only appear in a single model. The importance of the amino acid composition of peptides will be further reinforced in the subsequent discussion of other feature selection methods. Secondary structure elements also make a strong showing, appearing in 9 of the ten top models. This lends to the conclusion that amino acid content, and by extension physiochemical properties, and secondary structures are most relevant to AMP activity.

Univariate Selection

396 unique features appear in the ten models of interest when features are selected using univariate selection. This unexpectedly large number is due to the fact that some sample sets may have different features selected for the same model. Many of the features most frequently selected result from one of the reduction schemes with schemes C and B being highly represented in the most frequent features. It should be noted, however, that only one model made use of the C and B reduction schemes, so their high rate of appearance should be weighted against the fact that a single appearance one model set would result in 100% of the possible appearances. The relatively low accuracy rates of models arising from univariate selection lead to the assumption that this feature selection technique is not well suited to this machine learning problem. As this algorithm disproportionally selects features with larger deviation it will select those features where larger deviation is more likely, such as those from the simplex datasets where there are 81

potential features as opposed to amino acid content where there are only 20 potential features.

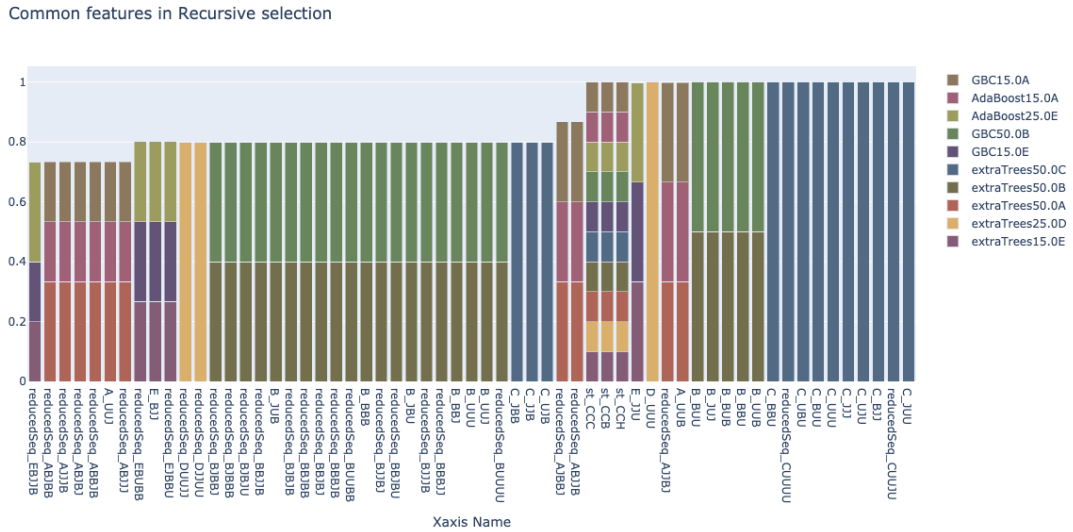


Figure 33:Distribution of occurrences of the most common features across the 10 models using univariate selection

Recursive Selection

There are 351 unique features selected using recursive selection. The features most commonly selected in this set represent a greater reliance on structural and amino acid content features, consistent with the results from the feature subsets above. Recursive feature selection resulted in models with greater accuracy indicating that this may be a more applicable feature selection technique for AMP prediction. Once again, features from reduction schemes B and C are highly represented in the feature set, though this may be an artificially inflated representation due to the small number of models utilizing those schema.

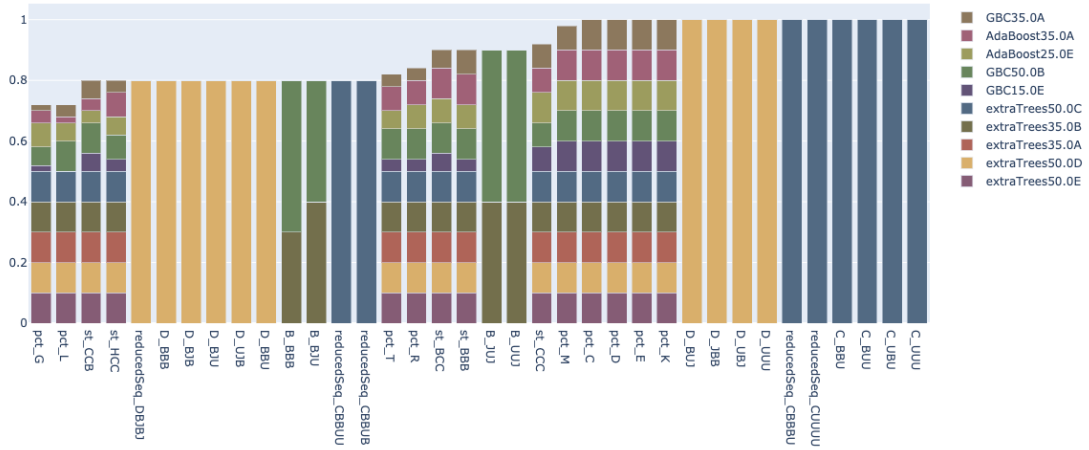


Figure 34: Distribution of occurrences of the most common features across the 10 models using recursive selection

Algorithm accuracy

Based on the results from feature selection it is not possible to select a specific model with the highest accuracy. We can see, however, that the ExtraTrees model seems to perform with the highest accuracy regardless of the features chosen indicating that this algorithm is robust enough to detect AMP's versus non-amps with approximately 89% accuracy. Considering that this algorithm accomplished similar accuracy using only structural and amino acid content columns it can be deduced that these small secondary structures and physiological characteristics are important factors in AMP activity.

CHAPTER 7 – LEARNING MODEL ANALYSIS

Introduction

While it is not possible to select a single 'best' model, it is still valuable to analyze the high performing models. The ExtraTrees models are approximately 89% accurate in AMP prediction, indicating that they have managed to identify some important factors in AMP structure. By taking a deeper look at the methods and features weights of several of these models a greater understanding of AMP structure can be found.

In this section several physiological features have been added to the dataset. Historically, AMP prediction has been accomplished using these physiochemical properties.[50,52,53,62,71,75,115] The most commonly cited properties are hydrophobic moment,[52,62] charge,[53,62,75] isoelectric point,[50,53,71] aromaticity, aliphatic index, hydrophobicity,[50,53,71,76,115] and charge density.[52] They have been added at this point to determine if they can supplement the previously developed sequence and structure based models.

Methods

Addition of physiochemical properties

Six physiochemical features were added to the dataset using the ModlAMP library[116]. The features selected were hydrophobic moment, charge, isoelectric point,

aromaticity, aliphatic index, and hydrophobic ratio. The models previously optimized for the original dataset were retrained using the dataset with these physiochemical properties. Feature reduction was done as described above.

Accuracy testing

An additional test set was developed using AMPs added to the CAMP database since the original datasets were developed. This second test set was developed and processed in the same manner as the original test set and shows statistically similar feature distribution. Each model was used to predict the label for each set. The accuracy, ROC and MCC for each prediction was recorded.

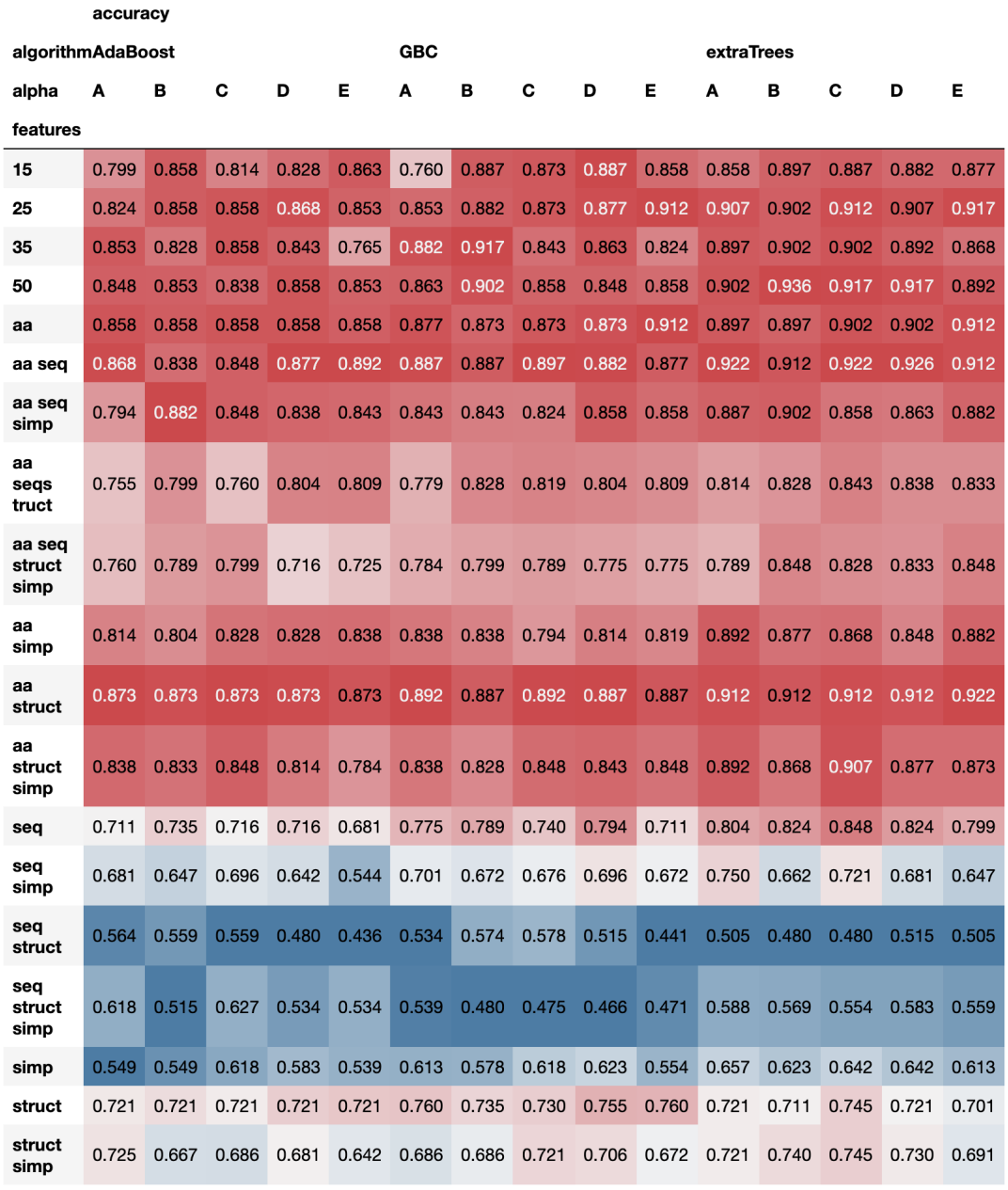
Results

Accuracy testing

Average accuracies for the initial test sets can be found in Figure 35: Average accuracies of each model for the initial test set and the accuracies for the new test set can be found in Figure 37. Accuracies ranged from 0.47 to 0.94 for the original test set and from 0.43 to 0.93 for the new test set.

accuracy															
algorithm	AdaBoost					GBC					extraTrees				
alpha	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
features															
15	0.861	0.853	0.867	0.868	0.876	0.888	0.908	0.902	0.900	0.894	0.892	0.918	0.895	0.906	0.897
25	0.892	0.879	0.880	0.873	0.888	0.906	0.885	0.894	0.903	0.906	0.918	0.927	0.924	0.915	0.923
35	0.868	0.876	0.894	0.902	0.856	0.897	0.900	0.921	0.898	0.903	0.909	0.930	0.918	0.932	0.912
50	0.880	0.886	0.885	0.908	0.882	0.914	0.911	0.911	0.902	0.898	0.920	0.932	0.927	0.921	0.929
aa	0.856	0.856	0.856	0.856	0.856	0.902	0.900	0.897	0.898	0.902	0.902	0.897	0.915	0.908	0.909
aa seq	0.867	0.859	0.882	0.880	0.870	0.880	0.905	0.912	0.914	0.905	0.927	0.924	0.930	0.923	0.918
aa seq simp	0.852	0.871	0.868	0.839	0.871	0.883	0.905	0.876	0.895	0.894	0.926	0.929	0.921	0.932	0.914
aa seqs truct	0.782	0.771	0.765	0.783	0.805	0.797	0.802	0.794	0.795	0.780	0.811	0.842	0.844	0.817	0.824
aa seq struct simp	0.798	0.814	0.811	0.780	0.806	0.826	0.859	0.798	0.852	0.833	0.871	0.864	0.892	0.858	0.848
aa simp	0.873	0.833	0.877	0.862	0.871	0.902	0.867	0.882	0.889	0.897	0.926	0.918	0.923	0.914	0.917
aa struct	0.886	0.886	0.886	0.886	0.886	0.912	0.914	0.914	0.912	0.912	0.918	0.924	0.924	0.911	0.918
aa struct simp	0.882	0.870	0.886	0.873	0.891	0.912	0.891	0.885	0.902	0.892	0.936	0.932	0.927	0.941	0.918
seq	0.715	0.673	0.700	0.700	0.650	0.753	0.730	0.745	0.752	0.733	0.783	0.756	0.779	0.789	0.777
seq simp	0.671	0.714	0.714	0.741	0.648	0.744	0.700	0.785	0.744	0.703	0.768	0.755	0.782	0.803	0.773
seq struct	0.541	0.535	0.544	0.515	0.494	0.521	0.473	0.600	0.520	0.474	0.508	0.479	0.485	0.527	0.485
seq struct simp	0.620	0.550	0.586	0.608	0.544	0.592	0.542	0.538	0.553	0.526	0.621	0.618	0.618	0.682	0.567
simp	0.659	0.650	0.720	0.723	0.667	0.715	0.733	0.764	0.753	0.692	0.732	0.723	0.755	0.771	0.695
struct	0.771	0.771	0.771	0.771	0.771	0.795	0.800	0.802	0.797	0.800	0.783	0.779	0.798	0.788	0.783
struct simp	0.774	0.776	0.785	0.789	0.739	0.791	0.771	0.808	0.795	0.782	0.806	0.806	0.791	0.817	0.773

Figure 35: Average accuracies of each model for the initial test set

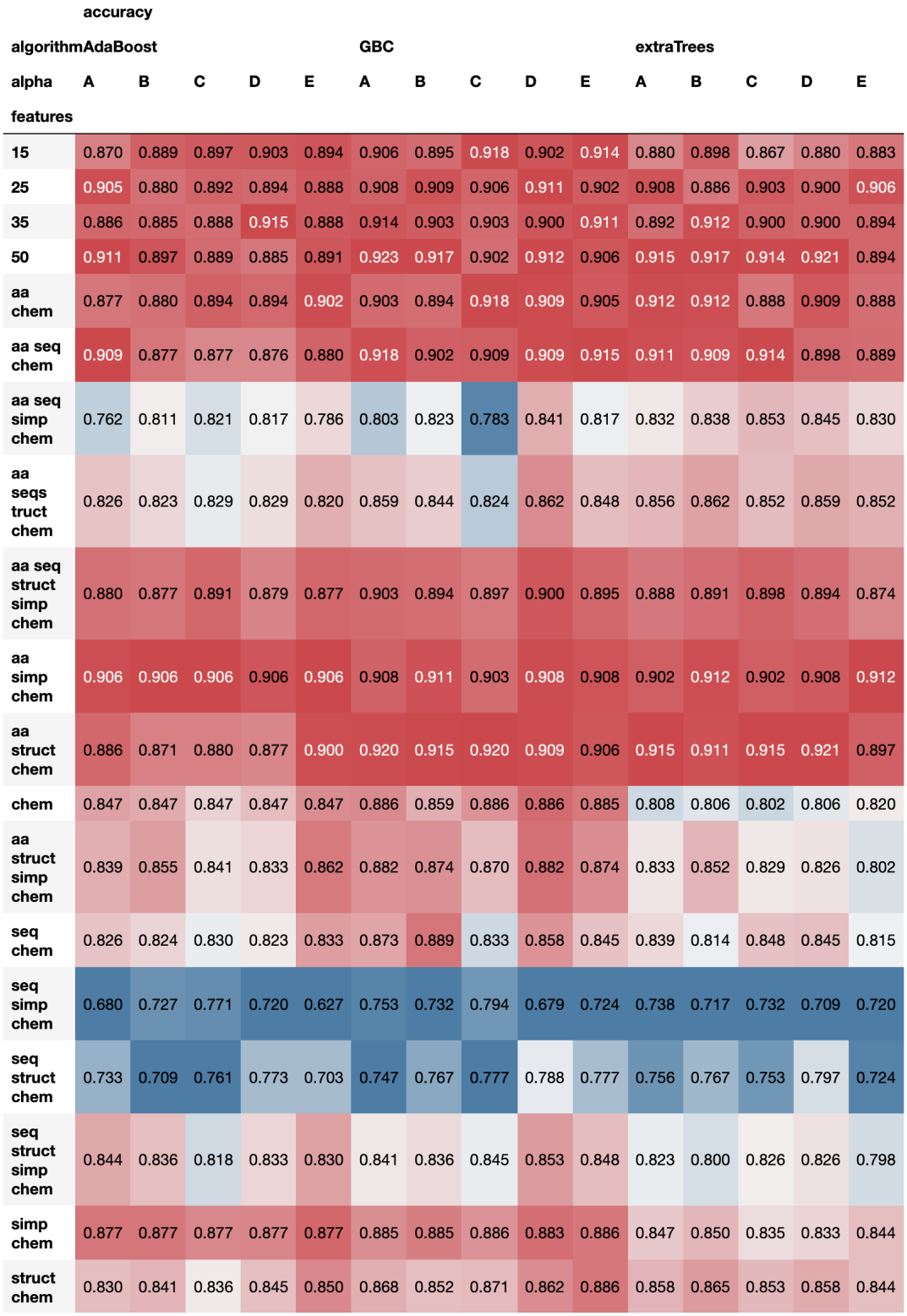


accuracy															
algorithm	AdaBoost					GBC					extraTrees				
alpha	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
features															
15	0.799	0.858	0.814	0.828	0.863	0.760	0.887	0.873	0.887	0.858	0.858	0.897	0.887	0.882	0.877
25	0.824	0.858	0.858	0.868	0.853	0.853	0.882	0.873	0.877	0.912	0.907	0.902	0.912	0.907	0.917
35	0.853	0.828	0.858	0.843	0.765	0.882	0.917	0.843	0.863	0.824	0.897	0.902	0.902	0.892	0.868
50	0.848	0.853	0.838	0.858	0.853	0.863	0.902	0.858	0.848	0.858	0.902	0.936	0.917	0.917	0.892
aa	0.858	0.858	0.858	0.858	0.858	0.877	0.873	0.873	0.873	0.912	0.897	0.897	0.902	0.902	0.912
aa seq	0.868	0.838	0.848	0.877	0.892	0.887	0.887	0.897	0.882	0.877	0.922	0.912	0.922	0.926	0.912
aa seq simp	0.794	0.882	0.848	0.838	0.843	0.843	0.843	0.824	0.858	0.858	0.887	0.902	0.858	0.863	0.882
aa seqs truct	0.755	0.799	0.760	0.804	0.809	0.779	0.828	0.819	0.804	0.809	0.814	0.828	0.843	0.838	0.833
aa seq struct simp	0.760	0.789	0.799	0.716	0.725	0.784	0.799	0.789	0.775	0.775	0.789	0.848	0.828	0.833	0.848
aa simp	0.814	0.804	0.828	0.828	0.838	0.838	0.838	0.794	0.814	0.819	0.892	0.877	0.868	0.848	0.882
aa struct	0.873	0.873	0.873	0.873	0.873	0.892	0.887	0.892	0.887	0.887	0.912	0.912	0.912	0.912	0.922
aa struct simp	0.838	0.833	0.848	0.814	0.784	0.838	0.828	0.848	0.843	0.848	0.892	0.868	0.907	0.877	0.873
seq	0.711	0.735	0.716	0.716	0.681	0.775	0.789	0.740	0.794	0.711	0.804	0.824	0.848	0.824	0.799
seq simp	0.681	0.647	0.696	0.642	0.544	0.701	0.672	0.676	0.696	0.672	0.750	0.662	0.721	0.681	0.647
seq struct	0.564	0.559	0.559	0.480	0.436	0.534	0.574	0.578	0.515	0.441	0.505	0.480	0.480	0.515	0.505
seq struct simp	0.618	0.515	0.627	0.534	0.534	0.539	0.480	0.475	0.466	0.471	0.588	0.569	0.554	0.583	0.559
simp	0.549	0.549	0.618	0.583	0.539	0.613	0.578	0.618	0.623	0.554	0.657	0.623	0.642	0.642	0.613
struct	0.721	0.721	0.721	0.721	0.721	0.760	0.735	0.730	0.755	0.760	0.721	0.711	0.745	0.721	0.701
struct simp	0.725	0.667	0.686	0.681	0.642	0.686	0.686	0.721	0.706	0.672	0.721	0.740	0.745	0.730	0.691

Figure 36:Average accuracies of each model for the new test set

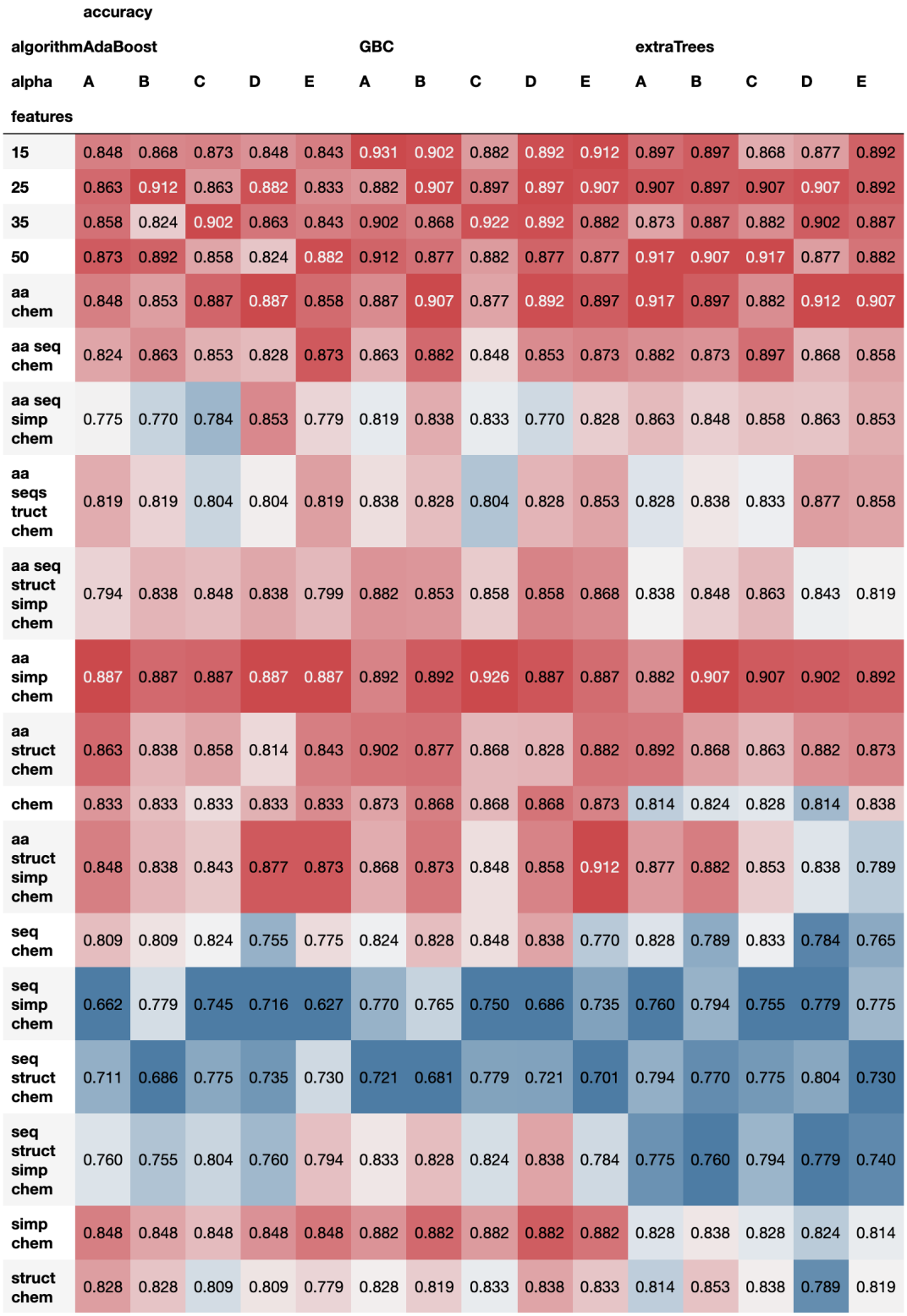
Accuracy testing with physiochemical features

Average accuracies for the initial test sets can be found in Figure 37 and the accuracies for the new test set can be found in Figure 38. Accuracies ranged from 0.62 to 0.92 for the original test set and from 0.62 to 0.93 for the new test set.



accuracy															
algorithm	AdaBoost					GBC					extraTrees				
alpha	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
features															
15	0.870	0.889	0.897	0.903	0.894	0.906	0.895	0.918	0.902	0.914	0.880	0.898	0.867	0.880	0.883
25	0.905	0.880	0.892	0.894	0.888	0.908	0.909	0.906	0.911	0.902	0.908	0.886	0.903	0.900	0.906
35	0.886	0.885	0.888	0.915	0.888	0.914	0.903	0.903	0.900	0.911	0.892	0.912	0.900	0.900	0.894
50	0.911	0.897	0.889	0.885	0.891	0.923	0.917	0.902	0.912	0.906	0.915	0.917	0.914	0.921	0.894
aa chem	0.877	0.880	0.894	0.894	0.902	0.903	0.894	0.918	0.909	0.905	0.912	0.912	0.888	0.909	0.888
aa seq chem	0.909	0.877	0.877	0.876	0.880	0.918	0.902	0.909	0.909	0.915	0.911	0.909	0.914	0.898	0.889
aa seq simp chem	0.762	0.811	0.821	0.817	0.786	0.803	0.823	0.783	0.841	0.817	0.832	0.838	0.853	0.845	0.830
aa seqs tructchem	0.826	0.823	0.829	0.829	0.820	0.859	0.844	0.824	0.862	0.848	0.856	0.862	0.852	0.859	0.852
aa seq struct simp chem	0.880	0.877	0.891	0.879	0.877	0.903	0.894	0.897	0.900	0.895	0.888	0.891	0.898	0.894	0.874
aa simp chem	0.906	0.906	0.906	0.906	0.906	0.908	0.911	0.903	0.908	0.908	0.902	0.912	0.902	0.908	0.912
aa struct chem	0.886	0.871	0.880	0.877	0.900	0.920	0.915	0.920	0.909	0.906	0.915	0.911	0.915	0.921	0.897
chem	0.847	0.847	0.847	0.847	0.847	0.886	0.859	0.886	0.886	0.885	0.808	0.806	0.802	0.806	0.820
aa struct simp chem	0.839	0.855	0.841	0.833	0.862	0.882	0.874	0.870	0.882	0.874	0.833	0.852	0.829	0.826	0.802
seq chem	0.826	0.824	0.830	0.823	0.833	0.873	0.889	0.833	0.858	0.845	0.839	0.814	0.848	0.845	0.815
seq simp chem	0.680	0.727	0.771	0.720	0.627	0.753	0.732	0.794	0.679	0.724	0.738	0.717	0.732	0.709	0.720
seq struct chem	0.733	0.709	0.761	0.773	0.703	0.747	0.767	0.777	0.788	0.777	0.756	0.767	0.753	0.797	0.724
seq struct simp chem	0.844	0.836	0.818	0.833	0.830	0.841	0.836	0.845	0.853	0.848	0.823	0.800	0.826	0.826	0.798
simp chem	0.877	0.877	0.877	0.877	0.877	0.885	0.885	0.886	0.883	0.886	0.847	0.850	0.835	0.833	0.844
struct chem	0.830	0.841	0.836	0.845	0.850	0.868	0.852	0.871	0.862	0.886	0.858	0.865	0.853	0.858	0.844

Figure 37 - Average accuracies of each model for the initial test set with physiochemical data



accuracy															
algorithm	AdaBoost					GBC					extraTrees				
alpha	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
features															
15	0.848	0.868	0.873	0.848	0.843	0.931	0.902	0.882	0.892	0.912	0.897	0.897	0.868	0.877	0.892
25	0.863	0.912	0.863	0.882	0.833	0.882	0.907	0.897	0.897	0.907	0.907	0.897	0.907	0.907	0.892
35	0.858	0.824	0.902	0.863	0.843	0.902	0.868	0.922	0.892	0.882	0.873	0.887	0.882	0.902	0.887
50	0.873	0.892	0.858	0.824	0.882	0.912	0.877	0.882	0.877	0.877	0.917	0.907	0.917	0.877	0.882
aa chem	0.848	0.853	0.887	0.887	0.858	0.887	0.907	0.877	0.892	0.897	0.917	0.897	0.882	0.912	0.907
aa seq chem	0.824	0.863	0.853	0.828	0.873	0.863	0.882	0.848	0.853	0.873	0.882	0.873	0.897	0.868	0.858
aa seq simp chem	0.775	0.770	0.784	0.853	0.779	0.819	0.838	0.833	0.770	0.828	0.863	0.848	0.858	0.863	0.853
aa seqs tructchem	0.819	0.819	0.804	0.804	0.819	0.838	0.828	0.804	0.828	0.853	0.828	0.838	0.833	0.877	0.858
aa seq struct simp chem	0.794	0.838	0.848	0.838	0.799	0.882	0.853	0.858	0.858	0.868	0.838	0.848	0.863	0.843	0.819
aa simp chem	0.887	0.887	0.887	0.887	0.887	0.892	0.892	0.926	0.887	0.887	0.882	0.907	0.907	0.902	0.892
aa struct chem	0.863	0.838	0.858	0.814	0.843	0.902	0.877	0.868	0.828	0.882	0.892	0.868	0.863	0.882	0.873
chem	0.833	0.833	0.833	0.833	0.833	0.873	0.868	0.868	0.868	0.873	0.814	0.824	0.828	0.814	0.838
aa struct simp chem	0.848	0.838	0.843	0.877	0.873	0.868	0.873	0.848	0.858	0.912	0.877	0.882	0.853	0.838	0.789
seq chem	0.809	0.809	0.824	0.755	0.775	0.824	0.828	0.848	0.838	0.770	0.828	0.789	0.833	0.784	0.765
seq simp chem	0.662	0.779	0.745	0.716	0.627	0.770	0.765	0.750	0.686	0.735	0.760	0.794	0.755	0.779	0.775
seq struct chem	0.711	0.686	0.775	0.735	0.730	0.721	0.681	0.779	0.721	0.701	0.794	0.770	0.775	0.804	0.730
seq struct simp chem	0.760	0.755	0.804	0.760	0.794	0.833	0.828	0.824	0.838	0.784	0.775	0.760	0.794	0.779	0.740
simp chem	0.848	0.848	0.848	0.848	0.848	0.882	0.882	0.882	0.882	0.882	0.828	0.838	0.828	0.824	0.814
struct chem	0.828	0.828	0.809	0.809	0.779	0.828	0.819	0.833	0.838	0.833	0.814	0.853	0.838	0.789	0.819

Figure 38 - Average accuracies of each model for the new test set with physiochemical data

Discussion

When selecting for entire subsets of features, tertiary structure alone seemed to provide the least discriminatory information, while amino acid content alone produced the best results of any single subcategory with accuracies around 90%. This is likely due to the consistently higher than average cystine content found in AMPs. We did, however see increased accuracy as additional feature subsets were incorporated into the training set. Amino acid content coupled with structure or sequence n-grams acihieved accuracies of up to 92% rivaling the accuracies achieved by the optimized feature sets.. Despite the fact that physiochemical properties have been traditionally used in AMP prediction, the addition of these features did not appear to significantly improve the predictive abilities of the machine learning algorithms over those developed with only sequence and structure characteristics. This may be due to the fact that the learning algorithm is able to infer the physiochemical properties from the amino acid content features.

With more robust feature selection methods a variety features are combined to produce a dataset that reduces computational complexity while increasing predictive ability. In these models, the percentages of amino acids continue to be highly weighted. Percentages of cytosine, aspartic acid and glutamic acid were the most heavily weighted features in the most accurate model. High percentages of cytosine contribute to a prediction of positive antimicrobial activity while high percentages of glutamic acid and aspartic acid contribute to a negative prediction [Figure 39]. Cystine residues allow for disulfide bonds which are believed to be integral in the three-dimensional confirmation of

the peptide molecule. Disulfide bridges are also involved in the formation of loops of the backbone structure. These loops can be exposed on the surface of the peptide and allow for interaction with external factors. Factors containing secondary looping structures are among the more highly weighted factors in the model [Figure 39], indicating that looping structures are important to AMP prediction.

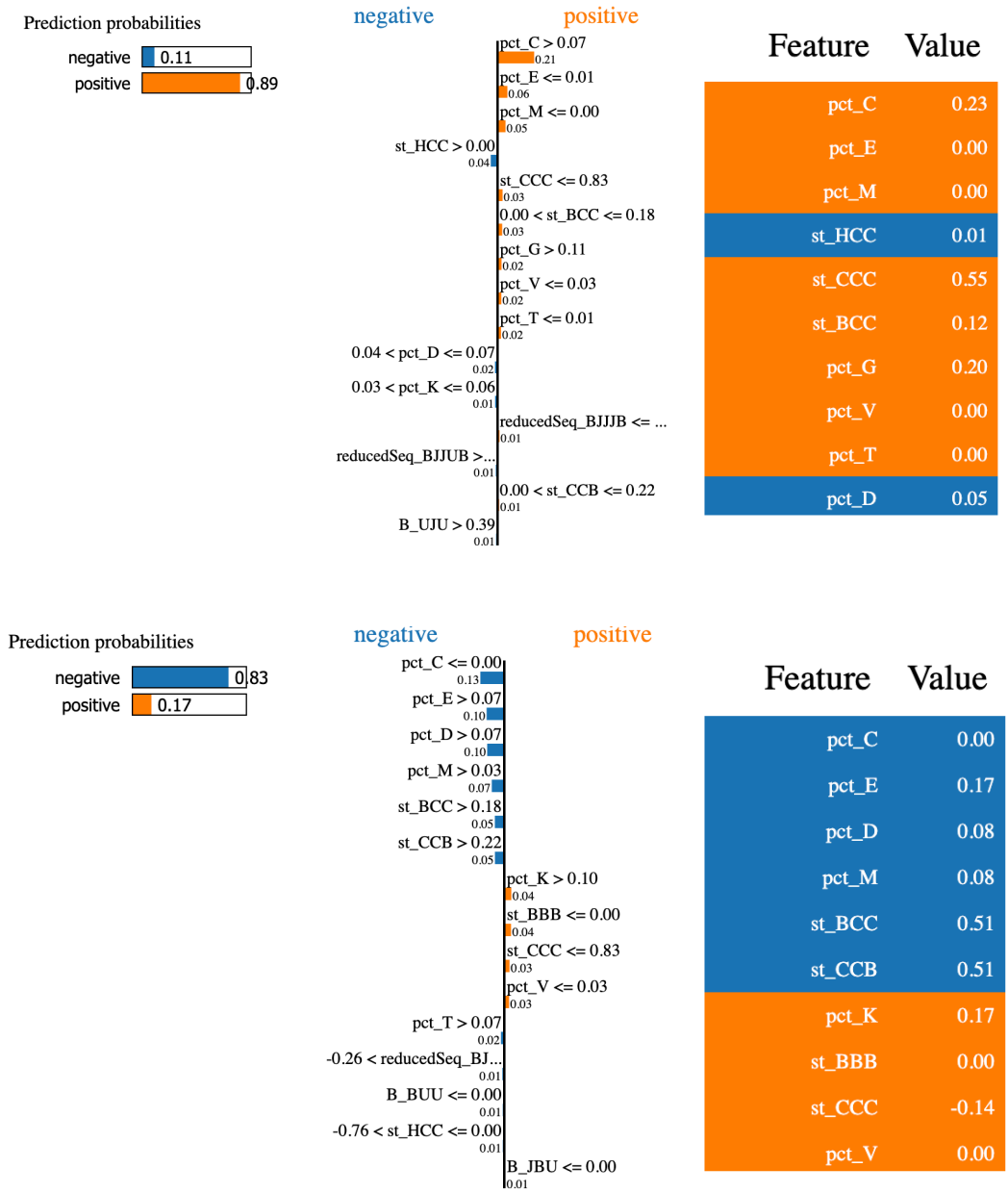


Figure 39 - Feature weights for positive and negative predictions

Conversely, the presence of higher percentages of aspartic and glutamic acid relate to negative predictions for antimicrobial activity. These amino acids are negatively charged at physiological pH and contribute to a negative or less positive charge for the

peptide as a whole. The lower prevalence of negative charged amino acids in AMPs would explain the fact that they are often cationic. Negatively charged peptides would have more difficulty associating with the negatively charged prokaryotic cellular membrane.

We also find that the prevalence of structures with high numbers of beta sheet structures are predicted to lack antimicrobial activity. This may indicate the importance of helical and looping structures in peptides with antimicrobial activity. Once again, this makes sense given that we know that the ability to form amphipathic structures are important for antimicrobial activity and the association between α-helical peptides and amphipathic character is well established.

CHAPTER 8 : AMP-CLASSIFICATION WEB APPLICATION

Introduction

To make the previously described encoding and learning algorithms more easily accessible to the general biological research community a web based application has been developed and is available at http://omics.gmu.edu/ssnaap [Figure 40].

Operation

Data Entry

The peptide for analysis may be entered in one of 3 ways.

1. PBD ID

   If the peptide in question is accessible through the protein data bank (https://www.rcsb.org/) a PDB id can be directly entered into the application. SSNAAP will request the PDB file directly from the API and process the data. Users may optially select the start and end residues or may indicate the chain to analyze (in the absence of explicit chain identification the A chain will be used.

2. Direct Upload

   In the case where a user would like to analyze a peptide that is not available directly from the PDB, a file in PDB format may be directly uploaded into the application. Data processing will proceed in the same manner as above with the same options to refine the peptide to be analyzed.

3. Sequence and structure input

Users may directly enter an amino acid sequence and, optionally, a secondary structure sequence. In this case the models that make use of any missing data, tertiary structure and potentially secondary structure, and will not be available.

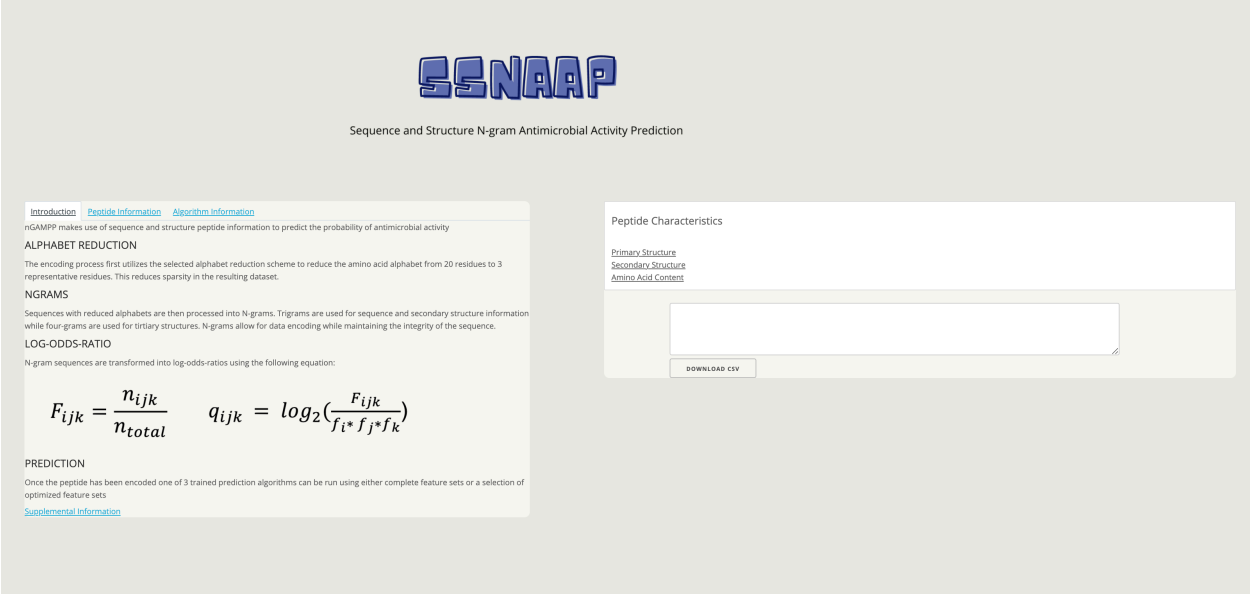


Figure 40: Landing page for the SSNAAP web application

Upon submit the application will perform initial data validation and processing. This initial validation will ensure that the peptide does not contain any non-standard amino acids. If this is the case the user will be alerted and asked to specify a start or end position to rectify the situation. After processing the app will display a graphic indicating the amino acid content of the selected peptide. At this point, the user may progress to algorithm selection.

Algorithm Selection

There are three algorithms to choose from within the algorithm selection tab, ExtraTrees, GradientBoostedClassifier and AdaBoost. These algorithms have been shown to be successful in AMP prediction [citation here to publication]. Users will also select from 5 alphabet reduction schemes. These five schemes are used to reduce the amino acid alphabet from 20 residues to 3 residues while maintaining the basic characteristics of each residue. This reduces data sparsity in the final dataset. Finally, the user will select the features to be used in construting the data set. There are 4 feature sets that have been optimized for AMP prediction using Scikit-learn's feature selection algorithms. These feature sets consist of features derived from all four feature groups. Users may also manually select one or more of the feature groups. These groups are based on the manner in which the data was derived. Amino Acid composition is simply the percentage of each amino acid in the peptide. Sequence n-grams are trigrams of the peptide after alphabet reduction. Secondary structure n-grams are trigrams of the DSSP annotated secondary structure. Finally, Tertiary simplexes are 4 residue simplexes derived from the three dimetional structure of the peptide. These simplexes were developed using the pyhull package.

Users may also choose to use a prediction window. This feature will allow for prediction of a defined number of residues across the entire peptide

Upon submitting the application will perform the required data processing and prediction. A prediction of high or low AMP probability will be returned if a single prediction is requested. In the case of a window prediction, a scatterplot of AMP probabilities will be

returned. The user then has the option of downloading the dataset developed by the app for further analysis.

CHAPTER 9: CONCLUSION AND FUTURE WORK

This work has developed a method for encoding peptides based on primary, secondary and tertiary structures for machine learning and has presented several machine learning models to predict generalized antimicrobial activity that rival those available in the current literature.

Chapter 4 details the development of positive and negative datasets as well as the methodology for encoding these datasets into features applicable for machine learning. This encoding takes advantage of five alphabet reduction techniques to reduce data sparsity. N-grams were utilized to maintain amino acid sequence composition while still enabling the development of numerical features. N-grams were also used to encode the secondary structure of peptides with similar results. Finally, amino acid simplices derived from Delaunay tessellations were similarly simplified using reduced alphabets and encoded into numeric log odds ratios. It is believed that this technique for encoding the complex structures of peptides can be extended to other computational biological questions not limited to AMP identification.

Chapter 5 explored the use of multiple machine learning algorithms for antimicrobial activity prediction. It was found that ensemble models such as Extra Trees, Gradient Boosted Classifiers and AdaBoost performed with the highest levels of

accuracy. These models were optimized and were found to result in high levels of accuracy (86-88%) when used to predict a test set.

In chapter 6 the learning models were used to determine relevant features for antimicrobial activity. Using a variety of feature reduction techniques feature sets consisting of 15, 25, 35 and 50 relevant features for each alphabet reduction technique were generated. These algorithms have been shown successful enough to serve as a screening mechanism used in conjunction with wet lab techniques to identify AMPs with the potential to supplement the currently available antibiotic and lessen the severity of the pending antibiotic crisis.

Chapters 7 and 8 detail the development and use of the SSNAAP web site for easy application of the leaning models developed in this work. This simple to use web based application makes the entire suite of learning algorithms available to researchers worldwide.

REFERENCES

1. Roberts, R. R. *et al.* Hospital and societal costs of antimicrobial-resistant infections in a Chicago teaching hospital: implications for antibiotic stewardship. *Clin. Infect. Dis.* **49**, 1175–84 (2009).
2. Zaidi, A. K. M. *et al.* Hospital-acquired neonatal infections in developing countries. *Lancet* **365**, 1175–1188 (2005).
3. Laxminarayan, R. & Heymann, D. L. Challenges of drug resistance in the developing world. *BMJ* **344**, 2–6 (2012).
4. Antibiotics Currently in Global Clinical Development | The Pew Charitable Trusts.
5. Kinch, M. S., Patridge, E., Plummer, M. & Hoyer, D. An analysis of FDA-approved drugs for infectious disease: Antibacterial agents. *Drug Discovery Today* **19**, 1283–1287 (2014).
6. Guiyoule, A. *et al.* Transferable plasmid-mediated resistance to streptomycin in a clinical isolate of Yersinia pestis. *Emerg. Infect. Dis.* **7**, 43–48 (2001).
7. Galimand, M. *et al.* Multidrug resistance in Yersinia pestis mediated by a transferable plasmid. *N. Engl. J. Med.* **337**, 677–680 (1997).
8. Alibek, K. & Handelman, S. *Biohazard: The Chilling True Story of the Largest Covert Biological Weapons Program in the World: Told from the inside by the Man Who Ran ItBiological Weapons: Limiting the Threat*. *Foreign Affairs* (1999).

doi:10.2307/20049476

9. Koplan, J. P. *et al. Centers for Disease Control and Prevention The production of this report as an MMWR serial publication was coordinated in Epidemiology Program Office Visual Information Specialist*. (2000).

10. Hotchkiss, R. D. & Dubos, R. J. FRACTIONATION OF THE BACTERICIDAL AGENT FROM CULTURES OF A SOIL BACILLUS. *J. Biol. Chem.* (1940).

11. Matsuzaki, K., Sugishita, K. I., Harada, M., Fujii, N. & Miyajima, K. Interactions of an antimicrobial peptide, magainin 2, with outer and inner membranes of Gram-negative bacteria. *Biochim. Biophys. Acta - Biomembr.* (1997). doi:10.1016/S0005-2736(97)00051-5

12. Van Epps, H. L. René dubos: Unearthing antibiotics. *J. Exp. Med.* (2006). doi:10.1084/jem.2032fta

13. Wang, Z. APD: the Antimicrobial Peptide Database. *Nucleic Acids Res.* **32**, 590D – 592 (2004).

14. Piotto, S. P., Sessa, L., Concilio, S. & Iannelli, P. YADAMP: yet another database of antimicrobial peptides. *Int. J. Antimicrob. Agents* **39**, 346–51 (2012).

15. Waghu, F. H., Barai, R. S., Gurung, P. & Idicula-Thomas, S. CAMPR3: a database on sequences, structures and signatures of antimicrobial peptides. *Nucleic Acids Res.* **44**, D1094-7 (2016).

16. Burger, L. & Van Nimwegen, E. Accurate prediction of protein-protein interactions from sequence alignments using a Bayesian method. *Mol. Syst. Biol.* (2008). doi:10.1038/msb4100203

17. Glazer, D. S., Radmer, R. J. & Altman, R. B. Combining molecular dynamics and machine learning to improve protein function recognition. in *Pacific Symposium on Biocomputing 2008, PSB 2008* (2008).

18. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V. & Fotiadis, D. I. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal* (2015). doi:10.1016/j.csbj.2014.11.005

19. Woolhouse, M. E. J., Webster, J. P., Domingo, E., Charlesworth, B. & Levin, B. R. Biological and biomedical implications of the co-evolution of pathogens and their hosts. *Nature Genetics* (2002). doi:10.1038/ng1202-569

20. Peschel, A. & Sahl, H. G. The co-evolution of host cationic antimicrobial peptides and microbial resistance. *Nature Reviews Microbiology* (2006). doi:10.1038/nrmicro1441

21. Pootoolal, J., Neu, J. & Wright, G. D. Glycopeptide antibiotic resistance. *Annu. Rev. Pharmacol. Toxicol.* (2002).

22. Gazit, E., Miller, I. R., Biggin, P. C., Sansom, M. S. P. & Shai, Y. Structure and orientation of the mammalian antibacterial peptide cecropin P1 within phospholipid membranes. *J. Mol. Biol.* (1996). doi:10.1006/jmbi.1996.0293

23. Fernandez, D. I. *et al.* The antimicrobial peptide aurein 1.2 disrupts model membranes via the carpet mechanism. *Phys. Chem. Chem. Phys.* (2012). doi:10.1039/c2cp43099a

24. Rapaport, D. & Shai, Y. Aggregation and organization of pardaxin in phospholipid membranes: A fluorescence energy transfer study. *J. Biol. Chem.* (1992).

25. Chen, F. Y., Lee, M. T. & Huang, H. W. Sigmoidal concentration dependence of antimicrobial peptide activities: A case study on alamethicin. *Biophys. J.* (2002). doi:10.1016/S0006-3495(02)75452-0

26. Cruciani, R. A. *et al.* Magainin 2, a natural antibiotic from frog skin, forms ion channels in lipid bilayer membranes. *Eur. J. Pharmacol. Mol. Pharmacol.* (1992). doi:10.1016/0922-4106(92)90045-W

27. Matsuzaki, K., Yoneyama, S. & Miyajima, K. Pore formation and translocation of melittin. *Biophys. J.* (1997). doi:10.1016/S0006-3495(97)78115-3

28. Silva, P. M., Gonçalves, S. & Santos, N. C. Defensins: Antifungal lessons from eukaryotes. *Frontiers in Microbiology* (2014). doi:10.3389/fmicb.2014.00097

29. Patrzykat, A., Friedrich, C. L., Zhang, L., Mendoza, V. & Hancock, R. E. W. Sublethal concentrations of pleurocidin-derived antimicrobial peptides inhibit macromolecular synthesis in Escherichia coli. *Antimicrob. Agents Chemother.* (2002). doi:10.1128/AAC.46.3.605-614.2002

30. Brogden, K. A., De Lucca, A. J., Bland, J. & Elliott, S. Isolation of an ovine pulmonary surfactant-associated anionic peptide bactericidal for Pasteurella haemolytica. *Proc. Natl. Acad. Sci. U. S. A.* (1996). doi:10.1073/pnas.93.1.412

31. Chongsiriwatana, N. P. *et al.* Intracellular biomass flocculation as a key mechanism of rapid bacterial killing by cationic, amphipathic antimicrobial peptides and peptoids. *Sci. Rep.* (2017). doi:10.1038/s41598-017-16180-0

32. Park, C. B., Kim, H. S. & Kim, S. C. Mechanism of action of the antimicrobial peptide buforin II: Buforin II kills microorganisms by penetrating the cell

membrane and inhibiting cellular functions. *Biochem. Biophys. Res. Commun.* (1998). doi:10.1006/bbrc.1998.8159

33. Del Castillo, F. J., Del Castillo, I. & Moreno, F. Construction and characterization of mutations at codon 751 of the Escherichia coli gyrB gene that confer resistance to the antimicrobial peptide microcin B17 and alter the activity of DNA gyrase. *J. Bacteriol.* (2001). doi:10.1128/JB.183.6.2137-2140.2001

34. Nan, Y. H. *et al.* Investigating the effects of positive charge and hydrophobicity on the cell selectivity, mechanism of action and anti-inflammatory activity of a Trp-rich antimicrobial peptide indolicidin. *FEMS Microbiol. Lett.* (2009). doi:10.1111/j.1574-6968.2008.01484.x

35. Verkleij, A. J. *et al.* The Asymmetric Distribution of Phospholipids in the Human Red Cell Membrane. *BBA - Biomembr.* (1973). doi:10.1016/0005-2736(73)90143-0

36. Bessalle, R., Kapitkovsky, A., Gorea, A., Shalit, I. & Fridkin, M. All-D-magainin: chirality, antimicrobial activity and proteolytic resistance. *FEBS Lett.* (1990). doi:10.1016/0014-5793(90)81351-N

37. Merrifield, R. B. *et al.* Retro and retroenantio analogs of cecropin-melittin hybrids. *Proc. Natl. Acad. Sci. U. S. A.* (1995). doi:10.1073/pnas.92.8.3449

38. Wade, D. *et al.* All-D amino acid-containing channel-forming antibiotic peptides. *Proc. Natl. Acad. Sci. U. S. A.* (1990). doi:10.1073/pnas.87.12.4761

39. Chen, H. C., Brown, J. H., Morell, J. L. & Huang, C. M. Synthetic magainin analogues with improved antimicrobial activity. *FEBS Lett.* (1988).

doi:10.1016/0014-5793(88)80077-2

40. Morgan, D. J., Okeke, I. N., Laxminarayan, R., Perencevich, E. N. & Weisenberg, S. Non-prescription antimicrobial use worldwide: a systematic review. doi:10.1016/S1473-3099(11)70054-8

41. Rost, B. & Sander, C. Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* (1993). doi:10.1006/jmbi.1993.1413

42. Wu, S. & Zhang, Y. A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics* (2008). doi:10.1093/bioinformatics/btn069

43. Kosciolek, T. & Jones, D. T. De novo structure prediction of globular proteins aided by sequence variation-derived contacts. *PLoS One* (2014). doi:10.1371/journal.pone.0092197

44. Nakashima, H., Nishikawa, K. & Ooi, T. The Folding Acid Type of a Protein Is Relevant to the Amino Composition. *J. Biochem.* **99**, 153–162 (1986).

45. Cedano, J., Aloy, P., Perez-Pons, J. A. & Querol, E. Relation between amino acid composition and cellular location of proteins. *J. Mol. Biol.* **266**, 594–600 (1997).

46. Reinhardt, A. & Hubbard, T. Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res.* **26**, 2230–2236 (1998).

47. Cheng, B. Y. M., Carbonell, J. G. & Klein-Seetharaman, J. Protein classification based on text document classification techniques. *Proteins Struct. Funct. Genet.* **58**, 955–970 (2005).

48. Chou, K. C. Prediction of protein cellular attributes using pseudo-amino acid

composition. *Proteins Struct. Funct. Genet.* **43**, 246–255 (2001).

49. Meher, P. K., Sahu, T. K., Saini, V. & Rao, A. R. Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC. *Sci. Rep.* **7**, 1–12 (2017).

50. Xiao, X., Wang, P., Lin, W. Z., Jia, J. H. & Chou, K. C. IAMP-2L: A two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Anal. Biochem.* **436**, 168–177 (2013).

51. Thomas, S., Karnik, S., Barai, R. S., Jayaraman, V. K. & Idicula-Thomas, S. CAMP: A useful resource for research on antimicrobial peptides. *Nucleic Acids Res.* **38**, 774–780 (2009).

52. Vishnepolsky, B. & Pirtskhalava, M. Prediction of linear cationic antimicrobial peptides based on characteristics responsible for their interaction with the membranes. *J. Chem. Inf. Model.* (2014). doi:10.1021/ci4007003

53. Khosravian, M., Kazemi Faramarzi, F., Mohammad Beigi, M., Behbahani, M. & Mohabatkar, H. Predicting Antibacterial Peptides by the Concept of Chou's Pseudo-amino Acid Composition and Machine Learning Methods. *Protein Pept. Lett.* **20**, 180–186 (2013).

54. Cavnar, W. B., Trenkle, J. M. & Mi, A. A. N-Gram-Based Text Categorization. *Proc. SDAIR-94, 3rd Annu. Symp. Doc. Anal. Inf. Retr.* (1994). doi:10.1.1.53.9367

55. Kešelj, V., Peng, F., Cercone, N. & Thomas, C. N-GRAM-BASED AUTHOR PROFILES FOR AUTHORSHIP ATTRIBUTION. *Pacific Assoc. Comput. Linguist.* (2003).

56. Agarwal, A., Xie, B., Vovsha, I., Rambow, O. & Passonneau, R. Sentiment analysis of twitter data. In Proceedings of the Workshop on Language in Social Media (LSM 2011). *Proc. Work. Lang. Soc. Media (LSM 2011)* (2011). doi:10.4018/ijhisi.2019040101

57. Skourikhine, A. N. & Burr, T. Linguistic analysis of the nucleoprotein gene of influenza A virus. in *Proceedings - IEEE International Symposium on Bio-Informatics and Biomedical Engineering, BIBE 2000* (2000). doi:10.1109/BIBE.2000.889607

58. Tomović, A., Janičić, P. & Kešelj, V. n-Gram-based classification and unsupervised hierarchical clustering of genome sequences. *Comput. Methods Programs Biomed.* (2006). doi:10.1016/j.cmpb.2005.11.007

59. Hamid, M. N. & Friedberg, I. Identifying antimicrobial peptides using word embedding with deep recurrent neural networks. *Bioinformatics* (2019). doi:10.1093/bioinformatics/bty937

60. Othman, M. *et al.* Classification and Prediction of Antimicrobial Peptides Using N-gram Representation and Machine Learning. in (2017). doi:10.1145/3107411.3108215

61. Solis, A. D. & Rackovsky, S. Optimized representations and maximal information in proteins. *Proteins Struct. Funct. Genet.* (2000). doi:10.1002/(sici)1097-0134(20000201)38:2<149::aid-prot4>3.0.co;2-%23

62. Porto, W. F., Pires, Á. S. & Franco, O. L. CS-AMPPred: An Updated SVM Model for Antimicrobial Activity Prediction in Cysteine-Stabilized Peptides. *PLoS One*

(2012). doi:10.1371/journal.pone.0051444

63. Xu, C., Ge, L., Zhang, Y., Dehmer, M. & Gutman, I. Computational prediction of therapeutic peptides based on graph index. *J. Biomed. Inform.* (2017). doi:10.1016/j.jbi.2017.09.011

64. Lata, S., Sharma, B. K. & Raghava, G. P. S. Analysis and prediction of antibacterial peptides. *BMC Bioinformatics* (2007). doi:10.1186/1471-2105-8-263

65. Lata, S., Mishra, N. K. & Raghava, G. P. S. AntiBP2: Improved version of antibacterial peptide prediction. *BMC Bioinformatics* (2010). doi:10.1186/1471-2105-11-S1-S19

66. Torrent, M. *et al.* AMPA: An automated web server for prediction of protein antimicrobial regions. *Bioinformatics* (2012). doi:10.1093/bioinformatics/btr604

67. Fjell, C. D. *et al.* Identification of novel antibacterial peptides by chemoinformatics and machine learning. *J. Med. Chem.* (2009). doi:10.1021/jm8015365

68. Jenssen, H., Fjell, C. D., Cherkasov, A. & Hancock, R. E. W. QSAR modeling and computer-aided design of antimicrobial peptides. *J. Pept. Sci.* (2008). doi:10.1002/psc.908

69. Frecer, V. QSAR analysis of antimicrobial and haemolytic effects of cyclic cationic antimicrobial peptides derived from protegrin-1. *Bioorganic Med. Chem.* (2006). doi:10.1016/j.bmc.2006.05.005

70. Cherkasov, A. & Jankovic, B. Application of ‘inductive’ QSAR descriptors for quantification of antibacterial activity of cationic polypeptides. *Molecules* (2004).

doi:10.3390/91201034

71. Fernandes, F. C., Rigden, D. J. & Franco, O. L. Prediction of antimicrobial peptides based on the adaptive neuro-fuzzy inference system application. *Biopolymers* (2012). doi:10.1002/bip.22066

72. Fjell, C. D., Hancock, R. E. W. & Cherkasov, A. AMPer: A database and an automated discovery tool for antimicrobial peptides. *Bioinformatics* **23**, 1148–1155 (2007).

73. Breiman, L. Random forests. *Mach. Learn.* (2001). doi:10.1023/A:1010933404324

74. James, G., Witten, D. & Hastie, T. *Introduction to Statistical Learning with Applications in R*. *Synthesis Lectures on Mathematics and Statistics* (2019). doi:10.2200/S00899ED1V01Y201902MAS024

75. Bhadra, P., Yan, J., Li, J., Fong, S. & Siu, S. W. I. AmPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest. *Sci. Rep.* (2018). doi:10.1038/s41598-018-19752-w

76. Joseph, S., Karnik, S., Nilawe, P., Jayaraman, V. K. & Idicula-Thomas, S. ClassAMP: A prediction tool for classification of antimicrobial peptides. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* (2012). doi:10.1109/TCBB.2012.89

77. Cortes, C. & Vapnik, V. Support-Vector Networks. *Mach. Learn.* (1995). doi:10.1023/A:1022627411411

78. Porto, W. F., Fernandes, F. C. & Franco, O. L. An SVM model based on physicochemical properties to predict antimicrobial activity from protein sequences with cysteine knot motifs. in *Lecture Notes in Computer Science*

(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2010). doi:10.1007/978-3-642-15060-9_6

79. Freund, Y. & Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* (1997). doi:10.1006/jcss.1997.1504

80. Abdo, A., Leclère, V., Jacques, P., Salim, N. & Pupin, M. Prediction of new bioactive molecules using a Bayesian belief network. *J. Chem. Inf. Model.* (2014). doi:10.1021/ci4004909

81. Chen, W. & Luo, L. Classification of antimicrobial peptide using diversity measure with quadratic discriminant analysis. *J. Microbiol. Methods* (2009). doi:10.1016/j.mimet.2009.03.013

82. Zare, M., Mohabatkar, H., Faramarzi, F. K., Beigi, M. M. & Behbahani, M. Using Chou's Pseudo Amino Acid Composition and Machine Learning Method to Predict the Antiviral Peptides. *Open Bioinforma. J.* (2015). doi:10.2174/1875036201509010013

83. Kozak, M. Erratum: A dendrite method for cluster analysis" by caliski and harabasz: AA classical work that is far too often incorrectly cited. *Communications in Statistics - Theory and Methods* (2012). doi:10.1080/03610926.2011.560741

84. Iso. Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions. *Measurement* (1994).

85. YERUSHALMY, J. Statistical problems in assessing methods of medical diagnosis, with special reference to X-ray techniques. *Public Health Rep.* (1947).

doi:10.2307/4586294

86. Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *BBA - Protein Struct.* (1975). doi:10.1016/0005-2795(75)90109-9

87. Li, T., Fan, K., Wang, J. & Wang, W. Reduction of protein sequence complexity by residue grouping. *Protein Eng.* (2003). doi:10.1093/protein/gzg044

88. Wang, J. & Wang, W. A computational approach to simplifying the protein folding alphabet. *Nat. Struct. Biol.* (1999). doi:10.1038/14918

89. Branden, C. I. & Tooze, J. *Introduction to Protein Structure*. *Introduction to Protein Structure* (2012). doi:10.1201/9781136969898

90. Mekler, L. B. Specific selective interaction between amino acid residues of the polypeptide chains. *Biophysics (Oxf).* (1969).

91. Murphy, L. R., Wallqvist, A. & Levy, R. M. Simplified amino acid alphabets for protein fold recognition and implications for folding. *Protein Eng.* (2000). doi:10.1093/protein/13.3.149

92. Regan, L. & Degrado, W. F. Characterization of a helical protein designed from first principles. *Science (80-.).* (1988). doi:10.1126/science.3043666

93. Kamtekar, S., Schiffer, J. M., Xiong, H., Babik, J. M. & Hecht, M. H. Protein design by binary patterning of polar and nonpolar amino acids. *Science (80-.).* (1993). doi:10.1126/science.8259512

94. Davidson, A. R., Lumb, K. J. & Sauer, R. T. Cooperatively folded proteins in random sequence libraries. *Nat. Struct. Biol.* (1995). doi:10.1038/nsb1095-856

95. Reidhaar-Olson, J. F. & Sauer, R. T. Combinatorial cassette mutagenesis as a probe of the informational content of protein sequences. *Science (80-.).* (1988). doi:10.1126/science.3388019

96. Smith, R. F. & Smith, T. F. Automatic generation of primary sequence patterns from sets of related protein sequences. *Proc. Natl. Acad. Sci. U. S. A.* (1990). doi:10.1073/pnas.87.1.118

97. Cieplak, M., Holter, N. S., Maritan, A. & Banavar, J. R. Amino acid classes and the protein folding problem. *J. Chem. Phys.* (2001). doi:10.1063/1.1333025

98. Bacardit, J. *et al.* Automated alphabet reduction for protein datasets. *BMC Bioinformatics* (2009). doi:10.1186/1471-2105-10-6

99. Liu, X., Liu, D., Qi, J. & Zheng, W. M. Simplified amino acid alphabets based on deviation of conditional probability from random background. *Phys. Rev. E. Stat. Nonlin. Soft Matter Phys.* (2002). doi:10.1103/PhysRevE.66.021906

100. Nanni, L. & Lumini, A. An ensemble of reduced alphabets with protein encoding based on grouped weight for predicting DNA-binding proteins. *Amino Acids* (2009). doi:10.1007/s00726-008-0044-7

101. Miyazawa, S. & Jernigan, R. Residue–Residue Potentials with a Favorable Contact Pair Term and an Unfavorable High Packing …. *Journal of Molecular Biology* (1996).

102. Xing, Z., Wang, J. & Wang, W. Effect of interaction energy fluctuation on the folding of a proteinlike model. *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.* (1998). doi:10.1103/PhysRevE.58.3552

103. Henikoff, S. & Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.* (1992). doi:10.1073/pnas.89.22.10915

104. Mekler, L. B. & Idlis, R. Construction of models of three-dimensional biological polypeptide and nucleoprotein molecules in agreement with a general code which determines specific linear recognition and binding of amino acid residues of polypeptides to each other and to the trinuc. *Depos. Doc. VINITI* **1476**, 81 (1981).

105. Anikeenko, A. V., Gavrilova, M. L. & Medvedev, N. N. Shapes of delaunay simplexes and structural analysis of hard sphere packings. *Stud. Comput. Intell.* (2009). doi:10.1007/978-3-540-85126-4_2

106. Singh, R. K. Delaunay tessellation of proteins: Four body nearest-neighbor propensities of amino acid residues. *J. Comput. Biol.* (1996). doi:10.1089/cmb.1996.3.213

107. Tropsha, A., Singh, R. K., Vaisman, I. I. & Zheng, W. Statistical geometry analysis of proteins: implications for inverted structure prediction. *Pac. Symp. Biocomput.* (1996).

108. Barber, C. B., Dobkin, D. P. & Huhdanpaa, H. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.* (1996). doi:10.1145/235815.235821

109. Wang, G., Li, X. & Wang, Z. APD2: The updated antimicrobial peptide database and its application in peptide design. *Nucleic Acids Res.* (2009). doi:10.1093/nar/gkn823

110. Mishra, B. & Wang, G. The importance of amino acid composition in natural amps: An evolutional, structural, and functional perspective. *Front. Immunol.*

(2012). doi:10.3389/fimmu.2012.00221

111. Wang, G. Human antimicrobial peptides and proteins. *Pharmaceuticals* (2014). doi:10.3390/ph7050545

112. Yin, L. M., Edwards, M. A., Li, J., Yip, C. M. & Deber, C. M. Roles of hydrophobicity and charge distribution of cationic antimicrobial peptides in peptide-membrane interactions. *J. Biol. Chem.* (2012). doi:10.1074/jbc.M111.303602

113. Chen, Y. *et al.* Role of peptide hydrophobicity in the mechanism of action of α-helical antimicrobial peptides. *Antimicrob. Agents Chemother.* (2007). doi:10.1128/AAC.00925-06

114. Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* (2002). doi:10.1023/A:1012487302797

115. James, G., Witten, D., Hastie, T. & Tibshirani, R. Tree-Based Methods. in (2013). doi:10.1007/978-1-4614-7138-7_8

116. Müller, A. T., Gabernet, G., Hiss, J. A. & Schneider, G. modlAMP: Python for antimicrobial peptides. *Bioinformatics* (2017). doi:10.1093/bioinformatics/btx285

BIOGRAPHY

Krista Smith graduated from Woodland Hills High School, Churchill, Pennsylvania, in 1993. She received her Bachelor of Science in Biology from Carnegie Mellon University in 1997 and her Masters of Science from Wright State University in 2001.