

Prediction of Antimicrobial Peptides using Hyperparameter Optimized Support Vector Machines



Musa Nur Gabere

Thesis presented in fulfillment of the requirements for the
Degree of Doctor Philosophiae
at the South African National Bioinformatics Institute
Faculty of Natural Sciences, University of the Western Cape

Advisors: **Prof. Vladimir Bajic** and **Prof. Alan Christoffels**.

September 6, 2011

Declaration

I declare that “**Prediction of Antimicrobial Peptides using Optimized Hyperparameters of Support Vector Machine**” is my own work, that it has not been submitted for degree or examination at any other university, and that all the resources I have used or quoted, and all work which was the result of joint effort, have been indicated and acknowledged by complete references.



Signed September 6, 2011



Abstract

Antimicrobial peptides (AMPs) play a key role in the innate immune response. They can be ubiquitously found in a wide range of eukaryotes including mammals, amphibians, insects, plants, and protozoa. In lower organisms, AMPs function merely as antibiotics by permeabilizing cell membranes and lysing invading microbes. Prediction of antimicrobial peptides is important because experimental methods used in characterizing AMPs are costly, time consuming and resource intensive and identification of AMPs in insects can serve as a template for the design of novel antibiotic. In order to fulfil this, firstly, data on antimicrobial peptides is extracted from UniProt, manually curated and stored into a centralized database called dragon antimicrobial peptide database (DAMPD). Secondly, based on the curated data, models to predict antimicrobial peptides are created using support vector machine with optimized hyperparameters. In particular, global optimization methods such as grid search, pattern search and derivative-free methods are utilised to optimize the SVM hyperparameters. These models are useful in characterizing unknown antimicrobial peptides. Finally, a webserver is created that will be used to predict antimicrobial peptides in haemotophagous insects such as *Glossina morsitan* and *Anopheles gambiae*.

Keywords: antimicrobial peptides, innate immune, machine learning, pattern search, simulated annealing, support vector machine, global optimization, database, insect, *Glossina morsistan*.

Dedication



Dedicated to my Mum and Dad

UNIVERSITY of the
Mariam Elmi Hassan

&

Nur Gabere Muhammed

Acknowledgement

Make it a habit to tell people thank you. To express your appreciation, sincerely and without the expectation of anything in return. Truly appreciate those around you, and you'll soon find many others around you. Truly appreciate life, and you'll find that you have more of it.

Ralph Marston.

First and foremost, I give my appreciation to **ALLAH (swt)**, the Cherisher and Sustainer of the Worlds for His Infinite Blessings and Guidance in my life.

I would like to express my sincere gratitude to my advisors, Prof Vladimir Bajic, Prof Alan Christoffels and our collaborator Prof William Noble, who have put their enormous effort and time in the supervision of this thesis. They have helped me to develop the research skills necessary to produce this document. Their innumerable and insightful suggestions have ultimately shaped this document.

Special thanks goes to the DAMPD database team members, namely Minna, Sundar, Ashley and Saleem. Also a million thanks to Mushal, who managed to convince me to use bibtex and also for explaining immunological concepts.

Warm thanks to the fantastic five, Ferial, Maryam, Junita, Fungiwe and Samantha for their excellent assistance regarding student finance and admin work. To the computer gurus, thanks to Peter and Dale for their technical assistance. I also thank my colleagues who have helped me at every step of the way. Special thanks goes to Sarah, Mario, Sumir, Direko, Oreetseng, Samuel, Monique, Zahra, Edwin, Ruben, Adugna, James, Simon, Junaid, Nicki and Gordon.

I cannot miss to mention the love, support and devotion of my family. They have given me the strength and encouragement, that made me complete my studies. Many thanks goes to my Mum, Mariam, my Dad, Nur, my sisters, Shugri and Amina and all my brothers, Mahammed, Abdi, Hassan, Abdihakim and Bashir I, Hussein, Bashir II, Mahammad and Mustaf. Special thanks goes to my elder brother, Mahammed, for

his constant communication and encouragement.

Lastly, my special gratitude goes to the SANBI, National Research Fund (NRF) for their financial support towards my research. Also thanks to MRC for the travelling fellowship to Mali, Bamako.



Nomenclature

Acronyms

AMP	Antimicrobial peptide
PAMP	Pathogen associated molecular patterns
DAMPD	Dragon antimicrobial peptide database
HAPP	Heamatophagous antimicrobial peptide predictor
GS	Grid search
PS	Pattern search
DFSA	Derivative free simulated annealing
SVM	Support vector machine
GS-SVM	Grid search support vector machine
PS-SVM	Pattern search support vector machine
DFSA-SVM	Derivative-free simulated annealing support vector machine
TP	True positive
FP	False positive
TN	True negative
FN	False negative
RBF	Radial basis function
PEP	Posterior error probability

Superscripts used throughout this thesis

k	Iteration counter
sa	Simulated annealing
t	Temperature counter

General symbols

Ω	Search region
N	Sample size
n	Dimension of the problem
f	Objective function
x	A vector
\min/\max	Minimize/Maximize
x_i	The i^{th} component of the vector x
l_i	Lower bound in the i^{th} dimension
u_i	Upper bound in the i^{th} dimension

Symbols related to pattern search

$x^{(k)}$	k^{th} iterate of x .
Δ_k	Step size parameter at iterate k
∇	First order derivative
D	The set of positive spanning directions

**Symbols related to derivative-free simulated annealing**

χ	Acceptance ratio
m_0	Number of trial points
m_1	Number of successful trial points
m_2	Number of unsuccessful trial points
δ	Cooling rate control parameter
ε_s	Stop parameter
RD	Random direction
MC	Markov chain
Δ_0^{sa}	Initial step size parameter used inside SA

Symbols related to support vector machine

\vec{x}_i	Input vector of features (patterns)
y_i	Target values (classes)
\mathcal{C}	Learner (classifier)
δ	Margin of hyperplane
\vec{x}^T	Transpose of vector \vec{x}
\mathbb{R}	Set of real numbers
X	Feature space
$ X $	Cardinality of set X
\log	Natural logarithm
n	Dimensionality of the input space



Contents

1	Introduction	1
1.1	Biophysical characteristics of antimicrobial peptide activity	3
1.2	Mode of action of AMPs	4
1.3	Approaches to characterize AMPs	5
1.4	Classification paradigm	6
1.5	Rationale of the thesis	8
1.6	The structure of the thesis	10
2	Antimicrobial peptide database: A collection of manually curated antimicrobial peptides	11
2.1	Introduction	12
2.2	Characteristics of the Database	13
2.3	Material and methods	20
2.4	Future work	25
2.5	Summary	25
3	Prediction of AMPs using parameter optimized support vector machines	26
3.1	Introduction	27
3.2	Algorithm	28
3.3	Material	44



3.4	Numerical Results	49
3.5	Discussion	65
3.6	Summary	71
4	HAPP webserver	72
4.1	Introduction	73
4.2	Methods	74
4.3	Estimation of statistical confidence measures	74
4.4	Results and Discussion	78
4.5	Description of the webserver	84
4.6	Summary	87
5	Conclusion and future work	88
5.1	Research contribution and limitations	88
	Appendix	90
	A Supplementary material for Chapter 2	90
	B Supplementary material for Chapter 3	94
	References	98



List of Figures

1.1	Mechanism of peptide action	5
2.1	Histogram of peptide distribution in the DAMPD database.	14
2.2	PHP retrieves MySQL data to produce Web pages.	15
2.3	An example of DAMPD entry	17
2.4	Flowchart describing the procedure for DAMPD database	21
2.5	Annotation error in a peptide with accession number P83141	23
3.1	Schematic of SVM classification	30
3.2	A non-separable one-dimensional problem	31
3.3	Separating the non-separable one-dimensional problem	31
3.4	A mesh plot of the hyperparameters c and σ against the accuracy (f) using grid search	33
3.5	Grid Search in a two dimensional optimization problem	34
3.6	System architecture of the proposed optimization of SVM hyperparameters	39
3.7	Histogram of sequence length distribution	45
3.8	Confusion matrix	48
3.9	Confusion matrix for prediction of amphibian AMPs using GS-SVM	56
3.10	Confusion matrix for prediction of amphibian AMPs using PS-SVM	57
3.11	Confusion matrix for prediction of amphibian AMPs using DFSA-SVM	58

4.1	Confusion matrix for prediction of insects AMPs using GS model	79
4.2	Confusion matrix for prediction of insecta AMPs using PS model	80
4.3	Confusion matrix for prediction of insecta AMPs using SAPS model	81
4.4	The figure represents the distribution of 1284 SVM scores for mixed and null peptides. . .	83
4.5	PEP and q -values	84
4.6	The input interface to the HAPP webserver	86
4.7	Prediction results of HAPP webserver	86
A.1	ClustalW results of AMP family page	90
A.2	Classification results of a query sequence using α -defensin HMM profile.	91
A.3	Hydrocalculator results of α -defensin sequence	92
A.4	SignalP results of α -defensin sequence	93
B.1	Figures (a)-(h) shows how the POLL steps works in the PS method.	95
B.2	Figure shows how the Grid Search works in a two dimensional optimization problem . . .	96

List of Tables

2.1	Amino acid frequency in the DAMPD database	14
2.2	Comparison of DAMPD database with other databases	20
3.1	PS-SVM parameters	50
3.2	DFSA-SVM parameters	50
3.3	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of actinopterygii AMPs	52
3.4	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of amphibian AMPs .	52
3.5	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of arachnida AMPs .	52
3.6	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of bacteria AMPs . .	52
3.7	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of crustacea AMPs .	53
3.8	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of insecta AMPs . .	53
3.9	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of mammalia AMPs .	53
3.10	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of merostomata AMPs	53
3.11	Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of plant AMPs . . .	53
3.12	Classification of actinopterygii antimicrobial peptides into AMP families	54
3.13	Classification of amphibian antimicrobial peptides into AMP families	59
3.14	Classification of arachnida antimicrobial peptides into AMP families	60
3.15	Classification of bacteria antimicrobial peptides into AMP families	60

3.16	Classification of insecta antimicrobial peptides into AMP families	61
3.17	Classification of mammalia antimicrobial peptides into AMP families	62
3.18	Classification of merostomata antimicrobial peptides into AMP families	62
3.19	Classification of plant antimicrobial peptides into AMP families	63
3.20	Classification of crustacea antimicrobial peptides into AMP families	63
3.21	Average performance comparison of the three hybrid methods (generalized AMP model)	64
3.22	Average performance comparison of the three hybrid methods (specialised AMP models)	64
4.1	Performance comparison of the three methods in prediction of Insecta AMPs	78



Chapter 1

Introduction

Multicellular organisms defend themselves against invasion by pathogens by mounting immune responses. An immune system is a network of cells, tissues and organs that work together to defend the organism against attacks by microbes and is divided into two categories namely adaptive immunity and innate immunity (Brahmachary et al., 2004).

Adaptive immunity refers to antigen-dependent immune response. The exposure in adaptive immunity results in immunology memory and there is a lag time between exposure and maximal response. The receptors in adaptive immunity recognize a particular part of the an antigen (epitope) to which an antibody binds. On the other hand, innate immunity refers to nonspecific defense mechanisms that come into play immediately after the appearance of an antigen in the body. The response of innate immunity is immediate, antigen-independent and the repercussion of the exposure is immunologically memoryless. The receptors in innate immunity have a broad specificity i.e., recognize many related molecular structures called pathogen associated molecular patterns (PAMPs). PAMPs are polysaccharides that vary little from one pathogen to another but are not found in the host. The defense mechanism in innate immunity involves physical, chemical and cellular approaches such as the use of antimicrobial peptides, phagocytosis and melanization (Yassine and Osta, 2010). All metazoans have inborn defense mechanisms that constitute innate immunity. Vertebrates have not only innate immunity but also an adaptive immunity (Steiner et al., 1981).

Antimicrobial peptides (AMPs) is a subset of proteins that plays an essential role in an innate immunity system. They are the first line of defense and widely distributed in plants, invertebrates and vertebrates and show activity against a broad spectrum of pathogens. They have antibacterial, antifungal, antiviral

and even antiprotozoal activities. Their resistance to pathogens has certainly contributed to their diversity and survival. Many similar AMPs have been identified from different organisms, proving their evolutionary importance in the defense mechanism. They are mostly cationic (positively charged), however there are examples of anionic peptides which also kill pathogens. Examples of cationic AMPs include but not restricted to cecropin, andropin, drosocin, metchnikowin, attacin, abaecin, α -defensin, β -defensin, penaeidin, drosomycin and gambicin. On other hand, maximins, dermicidin enkelytin, lactoferrin, hemocyanin, N- β -alanyl-5-S-glutathionyl-3,4-dihydroxyphenylalanine are examples of non-cationic AMPs (Vizioli and Salzet, 2002). The antimicrobial peptides selectively target the microbial membrane and takes advantage of the inherent difference between microbial cell membrane and multicellular plants and animals. The outer membrane of the microbe is composed of negatively charged phospholipids, whereas the outer membrane of plant and animal is populated with neutral lipids (Zasloff, 2002).

Antimicrobial peptides have 50% hydrophobic residues within a peptide and this feature enhances membrane permeabilization of the microbial. They are usually less than 100 amino acid residues in length (Hancock and Diamond, 2000). Many of these peptides are gene-encoded and synthesized by ribosomes (Tossi et al., 2002) though some are derived as cleaved sections from larger proteins such as lactoferrin (Bellamy et al., 1992) and buforin II from histone 2A (Park et al., 1998). Most of AMPs are generated from larger precursors that include a signal portion. They undergo post-translational modifications that involve proteolytic processing such as glycosylation (Bulet et al., 1993), carboxyl terminal amidation and amino-acid isomerization and halogenation (Zasloff, 2002).

These peptides are known to be so diverse that the same peptide sequence is rarely recovered from two different species of animal, even those closely related (Maxwell et al., 2003). Exceptions include peptides cleaved from highly conserved proteins, such as buforin II (Zasloff, 2002). However, within the antimicrobial peptides from a single species, and between certain classes of different peptides from diverse species, significant conservation of amino-acid sequences can be recognized in the pre-proregion of the precursor molecules (Simmaco et al., 1998). This suggests that the pre-proregion is probably conserved, as they are involved in secretion and intracellular trafficking of the peptide. The precursor molecule consists of the pre-proregion (signal peptides and preprotein sequences) and the matured peptides. It is in the mature peptide sequence that results in diversity in the AMPs structure and functions. The highly diverse nature of antimicrobial peptides arises from the need of each organism to adapt and survive in different microbial environments. Hence, even single mutations can dramatically alter the biological activity of these peptides (Boman, 2000).

This research is concerned with the characterization of antimicrobial peptides using machine learning. A number of methods have been implemented to characterize AMP by either using experimental or computational approaches. We will review these approaches later in the chapter. In the next section, we will present the biophysical properties, mode of action of AMPs and application of AMPs in medicine.

1.1 Biophysical characteristics of antimicrobial peptide activity

Despite the diversity of antimicrobial peptides in various organisms, many of them share common biophysical properties that endow them with the power to attack the microbial target. These properties include amphipathicity, charge (cationicity), hydrophobicity and conformation. We discuss these properties separately though they function holistically (Yount et al., 2006).

- **Amphipathicity (A):** Amphipathicity is a measure of abundance of hydrophobic and hydrophilic domains in a protein and is calculated using a hydrophobic moment (M_H). Amphipathicity enables permeabilization of the peptide against the microbial target (Yeaman and Yount, 2003).
- **Charge (Q):** Many of the antimicrobial peptides are cationic with a net positive charge ranging from +2 to +9. This is due to the fact that cationic peptides are rich in positively charged residues such as arginine and lysine. Cationicity plays an important role in the initial electrostatic attraction of antimicrobial peptides to negatively charged phospholipid membranes of bacteria and other microorganisms (Giangaspero et al., 2001; Yeaman and Yount, 2003).
- **Hydrophobicity (H):** Peptide hydrophobicity is the percentage of hydrophobic residues within a peptide. It is on average 50% for most antimicrobial peptides. Hydrophobicity is an essential property for antimicrobial peptide membrane interactions, as it enhances effective membrane permeabilization and also governs the extent to which a peptide can partition into the lipid bilayer of target membranes (Yeaman and Yount, 2003).
- **Conformation (χ):** Although antimicrobial peptides differ widely in primary sequence and source, AMPs assume a variety of secondary structures. Majority of the peptides have α -helical and β -sheet structures, whereas the remaining peptides can be classified as those that are enriched in one or more amino acid residues e.g., proline-arginine or tryptophan-rich. Peptides with α -helical structure and two antiparallel β -sheets are very active (Yeaman and Yount, 2003).

1.2 Mode of action of AMPs

Antimicrobial peptides act by targeting only the microbial membranes which have a clearcut difference from the membranes found in multicellular animals. The outermost leaflet of the microbial membrane bilayer, which is an exposed surface, is densely populated with lipids which have negatively charged phospholipids head groups. In comparison, the outer leaflet of the membranes of plants and animals are composed of neutral charged lipids (Matsuzaki, 1999). The antimicrobial interaction initially starts by disruption of the target membranes resulting into changed membrane potential, metabolite leakage and ultimately cell death (Carter and Hurd, 2010). Three mechanisms have been proposed for antimicrobial peptide membrane permeabilization, namely, barrel stave, carpet and torroid-pore as shown in Figure 1.1.

1.2.1 Barrel stave mechanism

In this model, the peptides bind to the membrane through electrostatic interactions. The peptides will take up an α -helical structure and grouped into bundles on the surface of the membrane. The bundles are inserted into the membrane bilayer such that the hydrophobic peptide regions are facing the lipid core of the membrane and the interior of the pore is formed by hydrophilic regions of the peptide. Continuous recruitment of additional peptide monomers leads to an increase in the size of the pore, ultimately resulting to leakage of intracellular components via these pores and subsequently leading to cell death (van 't Hof et al., 2001). Alamethicin peptide is an AMP that kills microbes using a barrel stave model (Brogden, 2005).

1.2.2 Toroidal pore mechanism

This model is similar to the barrel stave model, but there is no formation of bundle. Throughout the whole process, the hydrophilic surface of the peptide is in contact with the hydrophilic head groups of the cell membrane. The peptides and lipids bend inwards together to form well-defined pores. Examples of peptides that employ the toroidal pore mechanism are magainin, protegrin and melittin (Brogden, 2005).

1.2.3 Carpet mechanism

The carpet model proposes that the AMP clusters cover the surface of the membrane like a carpet. The membrane then collapses at the point of saturation of the concentration of the AMPs. Within a short span of time, wormholes are formed all over the membrane leading to an abrupt lysis of the microbial cell. The lipid layer bends back on itself like the inside of a torus. The lateral expansions in the polar head group region of the bilayer are filled up by individual peptide molecules (Shai, 2002). This model has been the proposed mechanism for dermaseptin, cecropin, melittin, caerin and olispirin (Brogden, 2005).

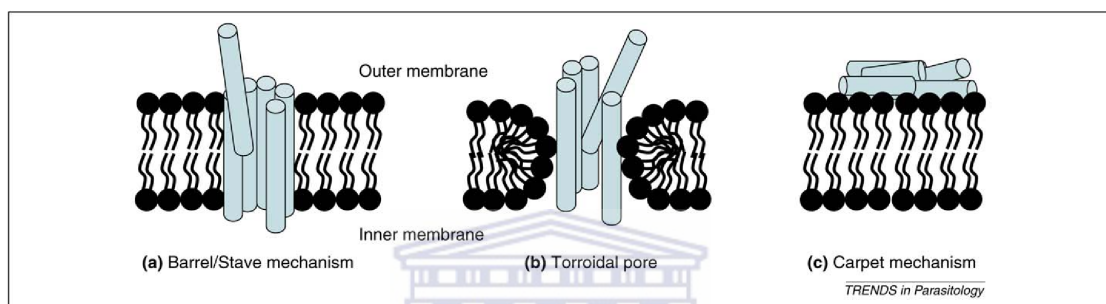


Figure 1.1: Mechanism of peptide action. The three main modes of action for peptide interaction with target membranes are: (a) Barrel stave. (b) Torroidal pore. (c) Carpet mechanism (Carter and Hurd, 2010).

UNIVERSITY of the
WESTERN CAPE

1.3 Approaches to characterize AMPs

Antimicrobial peptides have either antifungal, antibacterial, antiviral or antiprotozoal activities. The determination of the activities of an antimicrobial peptide can be assayed *in vivo* or predicted *in-silico*, i.e., classified into experimental approaches and computational approaches.

Experimental approaches for determining antimicrobial peptide activity include microscopy, fluorescent dyes, ion channel formation, circular dichroism and oriented circular dichroism, solid-state NMR spectroscopy and neutron diffraction. Microscopy is used to visualize the effects of antimicrobial peptides on microbial cells. Fluorescent dyes measure the extent at which antimicrobial peptides permeabilize membrane vesicles of microbial targets. Ion channel formation assesses the formation and stability of an antimicrobial peptide-induced pore. Circular dichroism and orientated circular dichroism measure the orientation and secondary structure of an antimicrobial peptide bound to a lipid bilayer. Solid-state NMR spectroscopy measures the secondary structure, orientation and penetration of antimicrobial peptides

into lipid bilayers. Finally, neutron diffraction quantifies the diffraction patterns of peptide-induced pores within membranes in oriented multilayers or liquids (Brogden, 2005). Although, experimental methods are getting more sophisticated to determine the antimicrobial peptide activities, computational methods take important precedence because of their inherent advantages. Currently, computational methods not only work as necessary supplements for experimental methods but also work as validation methods to remove false positive antimicrobial peptides verified through experimental approaches. Computational methods can be categorised using different approaches. These approaches include similarity search based techniques BLAST (Altschul et al., 1990) and PSI-BLAST (Altschul et al., 1997), profile search methods (profile hidden Markov model) and multivariate classification methods. Both similarity and profile search methods fail to predict new protein when query protein does not have significant similarity with the database proteins. In order to overcome this problem, we developed a support vector machine (SVM) based prediction method in this thesis. The machine learning technique called SVM was used because it extract complex patterns from biological sequence data. These techniques are highly successful for residue state prediction where fixed pattern length is used (Yang, 2004). In addition, SVM gives the best prediction performance because SVMs are designed to maximize the margin to separate two classes so that the trained model generalizes well on unseen data. Nonetheless, SVMs are able to minimize the structural risk by finding a unique hyperplane with maximum margin to separate the data from two classes. Because of this, SVM classifiers provide the best generalized ability to classified unseen data compared with other classifiers (Yang, 2004).

1.4 Classification paradigm

Classification is an important research area. It involves classifying samples according to a multivariate data by assigning each one of them a defined class. The objective in classification is to infer a classification rule from a sample of labelled training examples so that it classifies new examples with high accuracy. More formally, the classifier \mathcal{C} is given a training sample *train* of n examples, i.e.,

$$(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n). \quad (1.1)$$

drawn independently and identically distributed (i.i.d). Each example consists of the vector \vec{x} and the class label y . The vector \vec{x} describes the problem. The form of the class label depends on the type of classification task, and is divided into two main groups namely single-label classification and multi-label

classification (Joachims, 1998).

Single-label classification is concerned with learning from a set of examples that are associated with a single label l from a set of disjoint labels L , where $|L| > 1$. If $|L| = 2$, then the learning problem is called a binary classification problem while if $|L| > 2$, then it is called a multi-class classification problem. In binary classification, there are exactly two classes. For example, these two classes can be "normal" and "abnormal". This implies that the class label y has only two possible values. For notational convenience, let these values be +1 and -1. So $y \in \{-1, +1\}$. Example of binary classification include but are not restricted to classification of drug-likeness and agrochemical-likeness for a large compound collections (Zernov et al., 2003), classification of HIV-1 coreceptor usage i.e., CCR5 or CXCR4 which is useful in developing novel drug class of coreceptor antagonists (Sander et al., 2007). On the other hand multi-class classification involves more than two classes. For example, classifying unknown protein to one out of the ten protein families. This means that the class label y can assume 10, or in general l different values. So without loss of generality, $y \in 1, \dots, l$. The reduction of a multi-class problem into l binary tasks is often called a one-versus-rest (OVR) strategy. In OVR, the simplest approach is to reduce the problem of classifying among k classes into k binary problems, where each problem discriminates a given class from the other $k - 1$ classes (Statnikov et al., 2005). For this approach, we require k binary classifiers, where the k^{th} classifier is trained with positive examples belonging to the class k and negative examples belonging to the other $k - 1$ classes. When testing an unknown example, the classifier producing the maximum output is considered the winner, and this class label is assigned to that example. Another multi-class approach is all-versus-all (AVA). In this approach, each class is compared to each other class (Statnikov et al., 2005). A binary classifier is built to discriminate between each pair of classes, while discarding the rest of the classes. This requires building $\frac{k(k-1)}{2}$ binary classifiers. In testing a new example, a voting is performed among the classifiers and the class with the maximum number of votes wins. Multi-class classification as been used to classify microarray gene expression for cancer diagnosis (Statnikov et al., 2005) and classification of diabetic retinopathy stages into normal retina, non-proliferative diabetic retinopathy, proliferative diabetic retinopathy and macular edema (Acharya et al., 2011). In addition, it as been implemented to identify the states of histidines and cysteines (Passerini et al., 2006).

In multi-label classification, the examples are associated with a set of labels $Y \subseteq L$. Unlike in the single-label case, there is no one-to-one correspondence between class and examples in multi-label classification. Instead, for a fixed number l of categories, each example can be in multiple, exactly one or no category at all. For example, in medical diagnosis, a patient may belong to more than one conceptual

class. For example, a patient may be suffering from diabetes and high blood pressure. Example of multi-label classification include predicting gene function using hierarchical multi-label decision tree ensembles (Schietgat et al., 2010), hierarchical multi-label prediction of gene function (Barutcuoglu et al., 2006) and multi-label literature classification based on the gene ontology graph (Jin et al., 2008).

Multivariate classification methods are divided into two main branches, namely, multivariate statistics and machine learning. Multivariate statistics is where mathematical models are built to relate data to specific patterns of interest. The main disadvantage of statistical methods is that they are too restrictive and they rely on strict assumptions about the data being analysed. In particular, statistical methods tend to be tailored to modelling linear relationships. As opposed to these methods, machine learning simply learn a mathematical relationship that relates one set of data (the inputs) to another (the outputs). They are not statistically based and make no assumption about the data being analysed. For this reason, in this thesis, we concentrate on machine learning methods, especially support vector machine with hybridized optimization methods. Examples of multivariate statistics methods include k -nearest neighbour approach (Korn et al., 2007), linear discriminant analysis (Ye et al., 2005), principal component analysis (Tipping and Bishop, 1999), Naïve Bayes (Zhang et al., 2005), logistic regression (Popescul et al., 2003) random forest (Breiman, 2001). Examples of machine learning methods include artificial neural networks (Simpson, 1990) and support vector machines (Boser et al., 1992; Vapnik, 2000).

1.5 Rationale of the thesis

The rationale of the thesis derives from the following gaps in the literature, that is,

- The number of uncurated antimicrobial peptides is increasing and therefore there is need to clean and store these peptides. Efforts has been made to create AMP database to act as a repository for AMPs. Some of these databases on AMPs include APD (Wang et al., 2009; Wang and Wang, 2004), AMSdb (<http://www.bbcm.units.it/tossi/amsdb.html>), bactibase (Hammami et al., 2009, 2007) and defensin knowledgebase (Seebah et al., 2007; Verma et al., 2007), ANTIMIC (Brahmachary et al., 2004), PenBase (Gueguen et al., 2006), peptaibol (Whitmore et al., 2003), SAPD (Wade and Englund, 2002), AMPer (Fjell et al., 2007), BAGEL (de Jong et al., 2010, 2006) CAMP (Thomas et al., 2010) CyBase (Mulvenna et al., 2006; Wang et al., 2008a) and PhytAMP (Hammami et al., 2009). These databases have some inherent limitations. Firstly, some of the AMP databases are specialized such as PenBase (penaedin), Cybase (cyclic protein), BAGEL (bacteriocin) and efensin knowledgebase

(defensin). Secondly, they contain few analytical tools to aid in the analysis of AMPs. Thirdly, these databases are not updated on a regular basis. Lastly, they do not contain curated data on experimentally validated AMPs, for example, CAMP contains experimental AMPs, where some of them contain antitumor activities.

- Several computational approaches have been implemented to classify or rather characterize novel antimicrobial peptides from protein sequences. Recently, random forest, SVM and discriminant analysis has been applied in predicting antimicrobial peptides (Thomas et al., 2010). Artificial Neural Networks (ANN), Quantitative Matrices (QM) and Support Vector Machines (SVM) has been designed to predict antibacterial peptides (Lata et al., 2010, 2007). Quadratic discriminant analysis was used in classification of antimicrobial peptides using diversity measure with quadratic discriminant analysis (Chen and Luo, 2009). Fourier transform based method with property based coding strategy could be used to scan the peptide space for discovering new potential antimicrobial peptides (Nagarajan et al., 2006). Decision trees have been developed in for classification of antimicrobial peptides (Lee et al., 2004).
- Characterization of antimicrobial peptides in most of the insects have been well experimented and documented. However, there is less characterization of AMPs in the ongoing genome for *Glossina morsitans* (Tsetse fly) (Hu and Aksoy, 2005; Wang et al., 2008b). Tsetse flies are the medically and agriculturally important vectors of African trypanosomes. Nevertheless, no resource exist to predict antimicrobial peptides with statistical confidence measure in haemotophagous insect.

In order to fill these gaps, we have made a first step towards extracting and curating antimicrobial peptide sequences into a centralized database. This forms a basis for further analysis. Information gained from such analysis is useful for developing models for predicting novel antimicrobial sequences. In summary, the objectives of the thesis is to:

1. build a database of antimicrobial peptides with integrated query, extraction and sequence analysis tools,
2. design a methodology for predicting families of antimicrobial peptides using hybrid of SVM, pattern search and derivative-free simulated annealing method, and
3. create a web server for predicting antimicrobial peptides in haemotophagous (blood feeding insects), coupled with statistical confidence measure.

1.6 The structure of the thesis

The rest of the thesis is organized as follows. Chapter 2, presents the database for antimicrobial peptides termed as Dragon Antimicrobial Peptide Database (DAMPD).

Chapter 3 proposes two new hybrid methods to predict AMPs. These methods are based on the pattern search method and the simulated annealing method for optimizing the hyperparameters of the support vector machine.

Chapter 4 implements a specialized webserver called HAPP which is based on support vector machine to predict antimicrobial peptides in insects. We also discuss methodology for complementing SVM scores with statistical confidence measure, which forms the heart of this chapter.

Chapter 5 summarizes the work in this thesis and propose further avenues to extend and enhance this research. Finally, we give a description of the pattern search method, grid search method, keyword for negative set and feature indices in the appendices.

Chapter 2, 3 and 4 are in the process of being submitted to scholarly journals. In addition, the work presented in this thesis has been presented in the following workshop and conferences:

- Oral presentation on DAD: A database of antimicrobial peptides. *Second Southern African Bioinformatics Workshop held in Johannesburg, Johannesburg, South Africa, 2009.*
- Poster on *In-silico* prediction of antimicrobial peptides in Tsetse fly using profile hidden Markov model and support vector machine. *ISCB Africa ASBCB joint conference on Bioinformatics of Infectious Diseases, Bamako, Mali, 2009.*
- Oral presentation on Dragon antimicrobial peptide database: A collection of manually curated antimicrobial peptides. *22nd International CODATA Conference, South Africa, Stellenbosch, 2010.*
- Poster on Happ: Haemotophagous antimicrobial peptide predictor. *ISCB Africa ASBCB Conference on Bioinformatics, Cape Town, South Africa, 2011.*

Chapter 2

Antimicrobial peptide database: A collection of manually curated antimicrobial peptides



Abstract

Background: Antimicrobial peptides (Amps) are important components of the innate immune system widely distributed in prokaryote and eukaryotes. The interest in (AMPs) is increasing due to an increased tolerance of pathogens to conventional antibiotics.

Methods: The number of AMPs in public databases are not highly curated. In this study, over 4000 AMPs are extracted from UniProt and these peptides are manually curated.

Description: Manually curated 1232 experimentally validated AMPs are contained in the database. An integrated online user interface allows for querying along six search possibilities (taxonomy, species, family, citation, keyword and advance search). Tools such as BLAST, ClustalW, HMMER, hydrocalculator, SignalP, and Graphical views are integrated into the database to augment biological analysis of AMPs. The resulting database is called DAMPD.

Conclusion: This resource will serve as a useful complement to the existing public resources and as a good starting point for researchers interested in AMPs. DAMPD is freely accessible to academic and non-profit users at <http://apps.sanbi.ac.za/dampd>. DAMPD will be updated twice a year.

2.1 Introduction

Antimicrobial peptides (AMPs) are known for their significant role in the innate immune defense for all species of life. AMPs are found in eukaryotes, including mammals, amphibians, insects and plants, as well as in prokaryotes (Cole and Ganz, 2000; Garcia-Olmedo et al., 1998; Hancock and Diamond, 2000; Hoffmann and Hetru, 1992; Lehrer and Ganz, 2002; Rinaldi, 2002). A range of properties have been reported for AMPs including signaling molecular activity, low toxicity to mammals, broad target spectrum, and they may represent natural templates for anti-infectious agents in humans, since many microbes are showing resistance to current antibiotics (Hancock and Lehrer, 1998; Kamysz et al., 2003; van 't Hof et al., 2001). Microbial resistance to AMPs is highly reduced, as it would prove considerably difficult for microbes to modify their cell wall composition or each of the multiple targets of AMPs. Apart from naturally occurring AMPs, the design of novel peptides is receiving increased attention. The synthetic peptides are designed to have specific and enhanced activity in combating infectious agents.

AMPs vary in their mode of action as well as their biological activity. AMPs can cause cell death either by disruption of the microbial cell membrane, inhibiting extracellular polymer synthesis or intracellular functions (Hancock and Diamond, 2000). Studies on AMPs have shown that they are mostly cationic with length ranging from 6 to 100 amino acids with a few exceptions like maximin H5, dermcidin and enkelytin that has been shown to be anionic in nature (Brogden, 2005). AMPs also exhibit a high composition of hydrophobic residues. The majority of AMPs are amphipathic in nature with hydrophilic domain on one side and hydrophobic domain on the other. (Yeaman and Yount, 2003). It is proposed that the interaction of AMPs with the microbial cell membranes leading to cell permeation and lysis, can be attributed to their positive charge, hydrophobic nature and amphipathicity (Yeaman and Yount, 2003; Zasloff, 2002).

The number of uncurated antimicrobial peptides is increasing and therefore there is need to clean and store these peptides. Efforts has been made to create AMP database to act as a repository for AMPs. Some of these databases on AMPs include APD (Wang et al., 2009; Wang and Wang, 2004), AMSdb (<http://www.bbcm.units.it/tossi/amsdb.html>), bactibase (Hammami et al., 2009, 2007) and defensin knowledgebase (Seebah et al., 2007; Verma et al., 2007), ANTIMIC (Brahmachary et al., 2004), PenBase (Gueguen et al., 2006), peptaibol (Whitmore et al., 2003), SAPD (Wade and Englund, 2002), AMPer (Fjell et al., 2007), BAGEL (de Jong et al., 2010, 2006) CAMP (Thomas et al., 2010) CyBase (Mulvenna et al., 2006; Wang et al., 2008a) and PhytAMP (Hammami et al., 2009). These databases have some inherent limitations. Firstly, some of the AMP databases are specialized such as PenBase (pe-

naedin), Cybase (cyclic protein), BAGEL (bacteriocin) and efensin knowledgebase (defensin). Secondly, they contain few analytical tools to aid in the analysis of AMPs. Thirdly, these databases are not updated on a regular basis. Lastly, they do not contain curated data on experimentally validated AMPs, for example, CAMP contains experimental AMPs, where some of them contain antitumor activities. For these reasons, a **Dragon AntiMicrobial Peptide Database (DAMPD)** is created. It is a comprehensive and manually curated database of experimentally verified AMPs coupled with analytical bioinformatics tools.

This chapter is organized as follows: Section 2.2 gives the description of the database. Section 2.3 presents the methodology employed to build the database. Finally, future work and conclusion are made in sections 2.4 and 2.5 respectively.

2.2 Characteristics of the Database

The DAMPD database is the most elaborate repository of experimentally validated AMPs to date that has been manually curated. The database currently has 1232 number of entries (last updated on 6th of April, 2011), extracted from UniProt. The entries contain peptides ranging from both eukaryotic and prokaryotic organisms. The motivation for creating the database is to get reliable data that can be used for modeling of AMPs into their respective families. This database is useful as it will form the dataset used in the modeling processes of chapter 3 and 4.5. In addition to the peptide information, DAMPD database has utilities which assist in searching for AMPs such as species search, families search, taxonomy search, keyword search, citation search and advance search. It has an integrated analytical tools such as BLAST, ClustalW, hydrocalculator, SignalP and HMMER. These tools enhance analysis and classification of AMPs. Figure 2.1 and Table 2.1 and give the statistics of the data stored in DAMPD database. Figure 2.1, shows that most of the peptides have amino acid sequence length varying from 20 to 50 residues. Table 2.1 summarizes the amino acid percentages where glycine (10.44%) is the most abundant amino acid followed by leucine (9.16%).

The characteristics of the database namely, its architecture, organization, utilities, graphical views and tools are presented in subsections 2.2.1, 2.2.2, 2.2.3 and 2.2.5 respectively. Comparison of DAMPD database with existing databases is discussed in section 2.2.6.

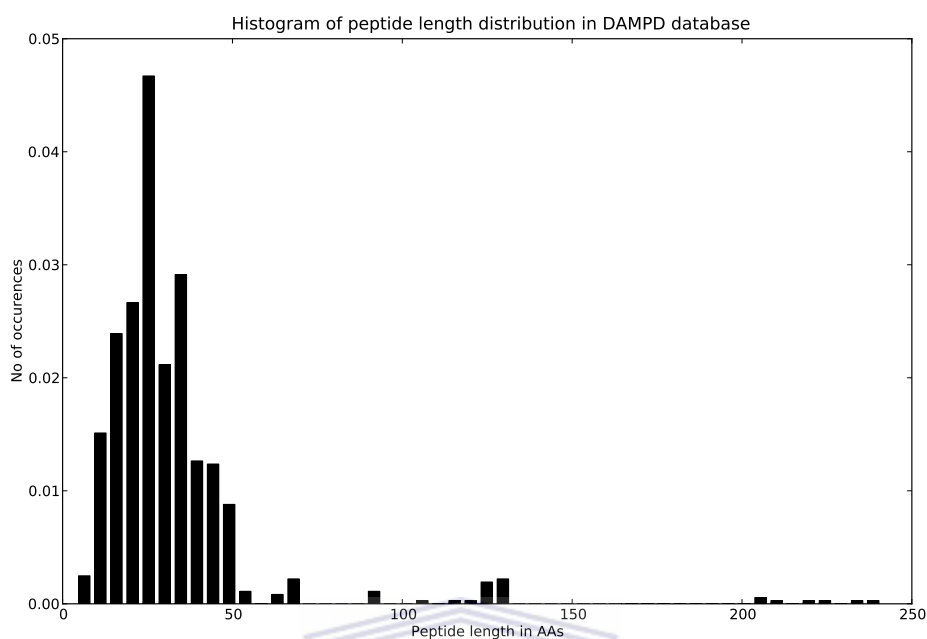


Figure 2.1: Histogram of peptide distribution in the DAMPD database.

Table 2.1: Amino acid frequency in the DAMPD database

Amino acid	Number of residues	% of total residues
C (Cysteine)	213	2.63
G (Glycine)	847	10.44
P (Proline)	417	5.14
A (Alanine)	860	10.6
V (Valine)	523	6.45
L (Leucine)	743	9.16
I (Isoleucine)	396	4.88
M (Methionine)	138	1.70
F (Phenylalanine)	367	4.52
Y (Tyrosine)	166	2.05
W (Tryptophan)	84	1.04
H (Histidine)	229	2.82
K (Lysine)	566	6.98
R (Arginine)	396	4.88
Q (Glutamine)	330	4.07
N (Asparagine)	428	5.27
E (Glutamic acid)	298	3.67
D (Aspartic acid)	298	3.67
S (Serine)	434	5.35
T (Threonine)	381	4.70

2.2.1 Database architecture

The dampd database is built on a linux operating system using the Apache web server, perl, python, PHP and MySQL relational database system. This architecture is shown in Figure 2.2, where PHP retrieves MySQL data.

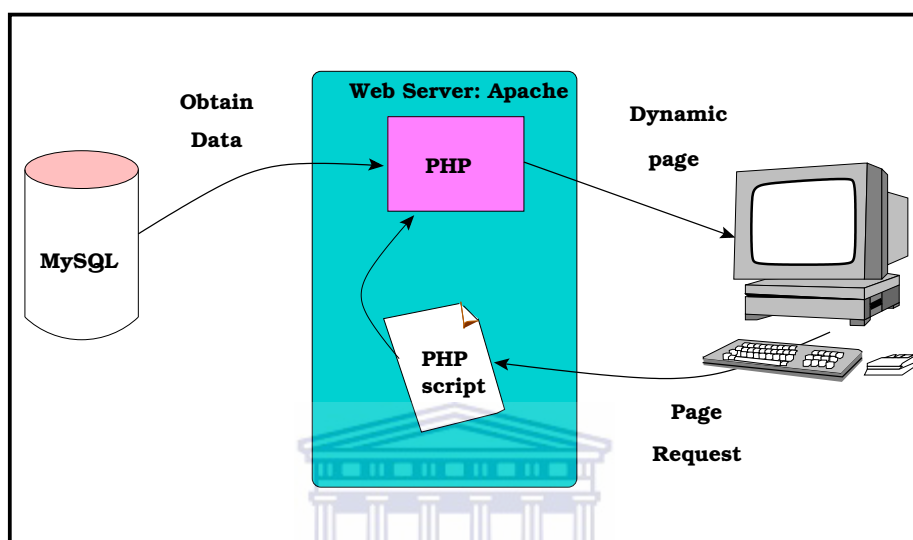


Figure 2.2: PHP retrieves MySQL data to produce Web pages.

UNIVERSITY of the
WESTERN CAPE

2.2.2 Database organization

Each DAMPD entry includes a description of the sequence, i.e., the entry information, name and origin, bibliography, comments, cross-references, DAMPD annotation and sequence information. An example of an entry in DAMPD database is shown in Figure 2.3. The annotation of each entry in the database contains the following fields. A unique DAMPD accession number that defines each record in the DAMPD database. Next, the protein name field gives the name of the peptide according to UniProt nomenclature. The field Entry date identifies the date when the entry was made and the description of the protein is given in protein description field. The organism source of AMPs can be found in the species field and its respective taxonomy is shown in the taxonomy field. The field protein existence gives proof of protein's existence be it at protein level, transcript level, inferred from homology or predicted. The bibliography field contains the literature references of the peptide in question. Relevant comments or remarks can be found in the comment field. The field comments gives the antimicrobial activity (antifungal, antibacterial, antiviral, antiprotozoal), subcellular location and AMP family of the peptide. In cross reference section,

the accession number used in UniProt to identify a given protein together with the hyperlink of the corresponding entry. In addition it provides useful links such as gene ontology (GO) and, other family and domain databases. The DAMPD curated keyword together with its reference is given in DAMPD manual curation field. The details of the sequence regarding the features, length, information (molecular weight) and the peptide sequence is given in the fields “features”, “length”, “sequence info” and “sequence” respectively.


2.2.3 Catalogue utilities

The DAMPD database contains several catalogues and integrated tools to help in data extraction and analysis of AMP sequence. One can extract peptides from the database using the following catalogues namely, taxonomy catalogue, species catalogue, citation catalogue, keyword catalogue, family catalogue and advance search catalogue. These catalogues have a vocabulary of terms whereby the entries in the database are retrieved. It also has additional functionality, which allows the user to choose individual entries from a search pool. That is, after generating a search result, the user can select individual records [from the result pool] for further processing.

In taxonomy catalogue, each peptide entry has a corresponding taxonomical classification, where each catalogue is made up of unique classification along with its corresponding total number of peptides enclosed in bracket. In the species catalogue, each peptide entry comes from a specific species and this is stored with its corresponding peptide ID with its corresponding peptide ID and a catalogue is made with total numbers shown in bracket. As for keyword catalogue, one can extract peptides using certain keywords given in the catalogue. In the family catalogue, one can search peptides using different AMPs sub-classes. The citation catalogue traces back all database entries to the original references. It is sub-divided into title (RA), journal (RL), author (RA) and year of publication (YR). Hence users can track the contribution of authors of a specific sequenced peptide. In advance search catalogue, there is a selection of search terms where the user chooses his own variable. It also allows user to query the database using field names, which are not listed in the other catalogues. For instance, one can search the entries in the database with the term experimental in the comment field.

2.2.4 Graphical views

The graphical views menu gives an external link to different databases of the query sequence and outputs the results in a graphical way. It furnishes additional information regarding a particular peptide. The fol-



DAMPD: Dragon Antimicrobial Peptide Database

Home SelectDB GraphicalViews Tools Acknowledgements Contacts | Links | FAQ | Help

Complete information for DAMPD ID Number : DAMPD:889 [Click WebBrowser Back Arrow to goto previous page]

ENTRY INFORMATION	
Damp Accession Number	DAMPD:889
Protein_Name	B1DYC_RANDY
Entry_Date	13-NOV-2007, integrated into UniProtKB/Swiss-Prot. 13-NOV-2007, sequence version 1. 05-APR-2011, entry version 16.
Protein_Description	RecName: Full=Brevinin-1DYc;

NAME AND ORIGIN	
Gene Name	Nil
Species	Rana dybowskii (Dybovsky's frog) (Korean brown frog).
organelle	Nil
NCBI_ID	NCBI_TaxID=71582
Taxonomy	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Amphibia; Batrachia; Anura; Neobatrachia; Ranoidea; Ranidae; Raninae; Rana; Rana.
Protein_Existence	1: Evidence at protein level

BIBLIOGRAPHY

Ref.No 1: Conlon J.M., Kolodziejek J., Nowotny N., Leprince J., Vaudry H., Coquet L., Jouenne T., Iwamuro S.; Cytolytic peptides belonging to the brevinin-1 and brevinin-2 families isolated from the skin of the Japanese brown frog, *Rana dybowskii*, *Toxicon* 50:746-756(2007)., **Position** : PROTEIN SEQUENCE, FUNCTION, AND MASS SPECTROMETRY., **Research Comment** : TISSUE=Skin secretion, **Links** : PubMed=17688900 [Abstract] DOI=10.1016/j.toxicon.2007.06.023

COMMENTS

FUNCTION: Antimicrobial peptide. Has low activity against the Gram-positive bacterium *S.aureus* and the Gram-negative bacterium *E.coli* (MIC<15 uM). Has a strong hemolytic activity.
SUBCELLULAR LOCATION: Secreted.
TISSUE SPECIFICITY: Expressed by the skin glands.
MASS SPECTROMETRY: Mass=2274.3; Method=MALDI; Range=1-20; Source=PubMed:17688900;
SIMILARITY: Belongs to the frog skin active peptide (FSAP) family, Brevinin subfamily.
Copyrighted by the UniProt Consortium, see <http://www.uniprot.org/terms>
Distributed under the Creative Commons Attribution-NoDerivs License

CROSS REFERENCES

UniProtKB : P0C5W8

- [Ontologies](#)
- [Family and domain databases](#)

DAMPD MANUAL CURATION OF Key words; Validation; Protein Existence Level

Antibacterial; Hemolytic	PubMed=17688900	1
--------------------------	-----------------	---

SEQUENCE DETAILS

FEATURES	PEPTIDE 1 20 Brevinin-1DYc. /FTid=PRO_0000311596. DISULFID 14 20 By similarity.
LENGTH	20
INFO	20 AA; 2278 MW; 69D6856FA52C7595 CRC64;
SEQUENCE	FLPLLLAGLP KLLCLFFKKC

South African National Bioinformatics Institute © 1996 - 2011
OrionCell © 2008 - 2011

Figure 2.3: An example of DAMPD entry

lowing graphical views are integrated: *ProtParam* computes the physico-chemical properties of a peptide sequence (Gasteiger et al., 2005). *Compute PI/MW* allows user to compute isoelectric point and molecular weight (Bjellqvist et al., 1994). *ProtScale* generates a profile of each amino acid on a selected protein

(Gasteiger et al., 2005). *PeptideMass* computes the masses of the generated peptides and also returns theoretical isoelectric point and mass values for the protein of interest (Wilkins et al., 1997). *PeptideCutter* predicts potential cleavage sites cleaved by proteases on a given protein sequence (Gasteiger et al., 2005). *ModBase* is a database of predicted protein structure models (Pieper et al., 2009). *SMART* a (Simple Modular Architecture Research Tool) that maps a protein sequence to its catalogue of target domains (Letunic et al., 2009). *InterPro* uses a host of member databases to generate protein signatures, which are used as a basis to identify distant relationships between potentially novel sequences (Apweiler et al., 2000). *Pfam* is a database of protein family classification, protein domain data and multiple sequence alignments generated using Hidden Markov models (Finn et al., 2010). *Prosite* is a database, which contains descriptions and documentation relating to amino acid profiles, protein domains, families and functional sites (Sigrist et al., 2010). *ProtoNet* is a database of computationally derived protein structures, which have been clustered and then hierarchically structured using data, derived from UniProt/TrEMBL (Sasson et al., 2003).

2.2.5 Tools

The DAMPD database contains the following tools to assist in the analysis of AMP sequences, namely BLAST (Altschul et al., 1990) and ClustalW (Thompson et al., 1994), NJplot (Perrière and Gouy, 1996), HMMER (Eddy, 1998) and Hydrocalculator (Tossi et al., 2002) and SignalP (Bendtsen et al., 2004). They are integrated in the system and can be accessed either from the tool page or from the catalogue results page.

Catalogue-integrated tool

Each catalogue page (taxonomy, species etc) contains integrated tools such as BLAST, ClustalW, HMMER, hydrocalculator and SignalP. When the user performs a search, the result page shows the summary of the peptides and the user can choose to analyse (using tools) the entire result set or chosen set of sequences from the total set.

Standalone tool

The DAMPD database tools can also operate on a standalone basis, which is located on the tool menu. That is, the user can process sequences contained in the database or any other sequences. For example, one can perform multiple alignment of antimicrobial sequences using ClustalW and in addition, one can

view phylogenetic tree of the aligned sequence generated by ClustalW using NJplot. HMMER allows user to tentatively classify unknown sequences into a particular antimicrobial family using two ways: (i) the user can either use 27 predefined antimicrobial library of profiles or (ii) use their own generated profiles. The physicochemical properties of the peptides such as hydrophobicity, net charge, percentage of hydrophobic residues, mean hydrophobicity and mean hydrophobic moment can be calculated using the hydrocalculator tool. SignalP can be used to predict the signal cleavage site of a peptide. The results page for ClustalW, HMMER, hydrocalculator, signalP are given in the Appendix A.1, A.2, A.3 and A.4 respectively.

2.2.6 Comparison of DAMPD database with existing databases

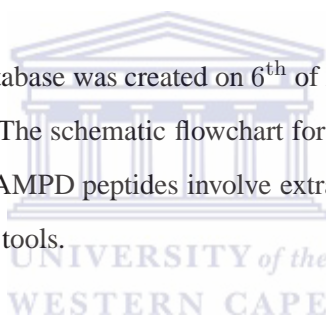
Several database has been created to store AMPs. For example, in APD2 (Antimicrobial Peptide Database) (Wang et al., 2009), the quality of annotation is poor in terms of function and the database does not have links to other databases. CAMP (collection of Anti-Microbial Peptides) database (Thomas et al., 2010) has quite good quality of functional annotation in the entries but not all entries have been fully annotated. It contains 1216 experimentally verified proteins, but at least a hundred of their entries include proteins that are annotated wrongly, or have antitumor activities only. AMSDb (Antimicrobial Sequences Database) <http://www.bbcm.units.it/~tossi/amsdb.html> is another simplified mini-versions of entries found in UniProt. The number of peptide entries has not been updated for the last seven years. There is no analytical tools in this database. Defensin knowledgebase (defensin) (Seebah et al., 2007; Verma et al., 2007) bactibase (bacteriocin) (Hammami et al., 2009, 2007), PenBase (penaeidin) (Gueguen et al., 2006), peptaibol Database (peptaibols) (Whitmore et al., 2003), SAPD (Synthetic Antibiotic Peptides Database) (Wade and Englund, 2002), CyBase (cyclic protein) (Mulvenna et al., 2006; Wang et al., 2008a), BAGEL (bacteriocins) (de Jong et al., 2010, 2006) and PhytAMP (plant) (Hammami et al., 2009) are specialised database and not regularly updated. DAMPD database is the most elaborative warehouse of natural AMPs to date, which has been manually curated. It contains 1232 antimicrobial peptides that have entries obtained from UniProt. The entries come from both eukaryotic and prokaryotic organisms. Nonetheless, the database has utilities and integrated data extraction tools such as search utilities (taxonomy, classification, keyword, citation, families and advance search), graphical views and analytical tools. It is updated after six months. Comparison of DAMPD database with other databases is shown in Table 2.2.

Table 2.2: Comparison of DAMPD database with other databases

Features	Nature	# of Expt. AMPs	Search tools	Analytical tools	Graphical views	AMP Prediction
Defensin	Specific	363	Absent	Absent	Absent	Absent
PenBase	Specific	29	Absent	Absent	Absent	Absent
Peptaibol	Specific	317	Absent	Absent	Absent	Absent
AMSDb	Specific	895	Absent	Present	Absent	Absent
SAPD	Specific	200	Absent	Present	Absent	Absent
APD	General	1502	Absent	Present	Absent	Based on similarity approach
PhytAMP	Specific	273	Present	Absent	Present	Based on HMM profiles
CAMP	General	1216	Present	Absent	Absent	Based on SVM, random forest, discriminant analysis
DAMPD	General	1232	Present	Present	Present	Based on HMM and SVM model

2.3 Material and methods

The Dragon antimicrobial peptide database was created on 6th of April, 2011 with 1232 curated AMP that have been experimentally validated. The schematic flowchart for building the database is given in Figure 2.4. The process for obtaining the DAMPD peptides involve extraction and curation. Then the clean data is coupled with search and analytical tools.



2.3.1 Data extraction

The raw data was retrieved from UniProt database by using the search term “antimicrobial [KW-0929]”. Entries that had been assigned either to protein existence level “evidence at protein level” or “evidence at transcript level” were concentrated on. The extracted raw data from UniProt contains misannotation and hence there is need to curate them.

2.3.2 Data curation

The exponential growth in the amount of biological data means that revolutionary measures are needed for data management, analysis and accessibility. Due to rapid release of new data from genome sequencing projects, the majority of protein sequences in public databases have not been experimentally characterized; rather, sequences are annotated using computational analysis. The level of misannotation and the types of misannotation in large public databases are currently unknown and have not been analyzed in depth (Harris, 2003; Schnoes et al., 2009). For example the entries in UniProtKB should be of high quality

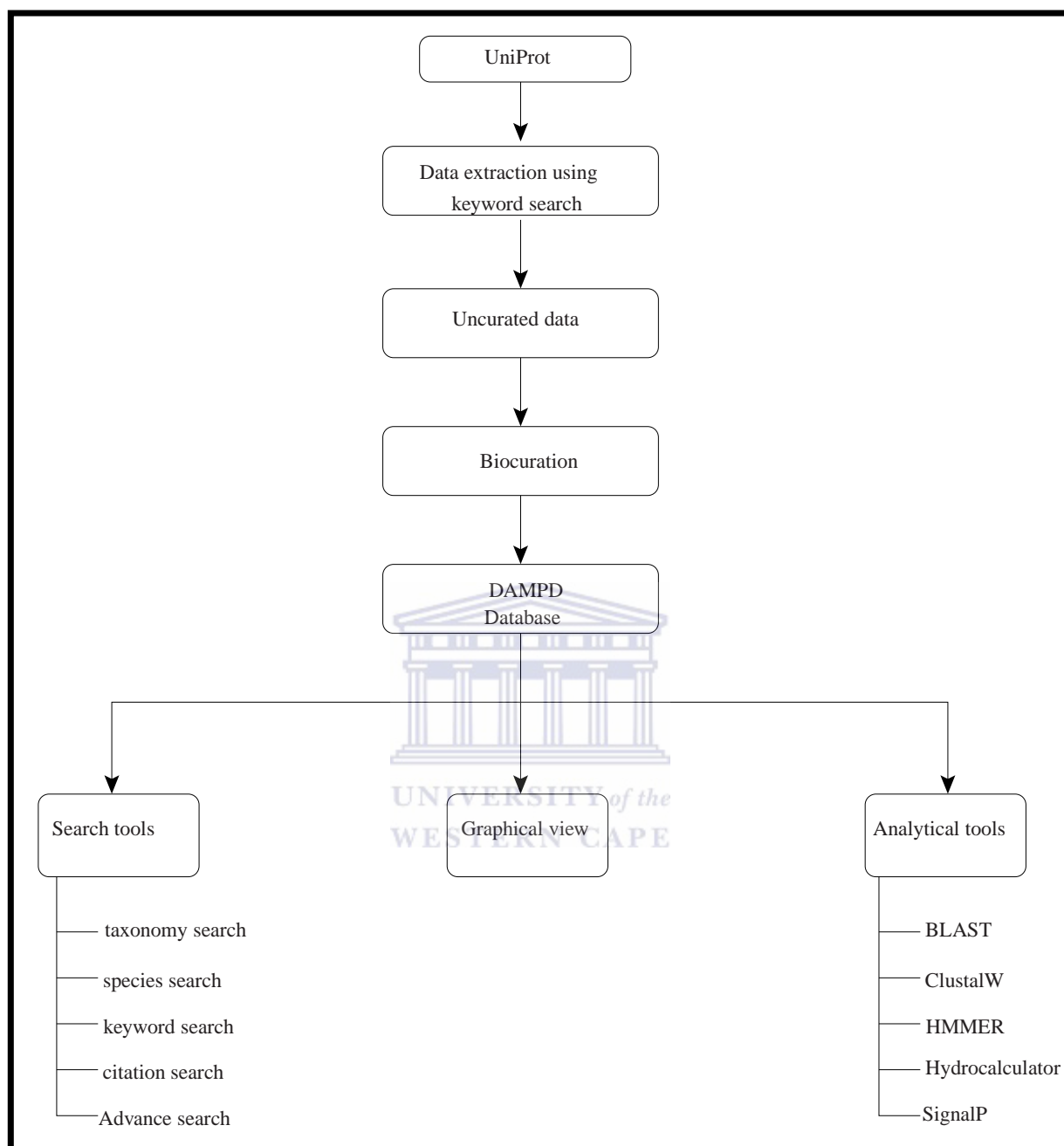


Figure 2.4: Flowchart describing the procedure for DAMPD database

annotation before they are made public. However, there are mistakes and some of these include but not limited to

- wrong keywords, e.g. antibiotic or fungicide tagged in an entry where they should not be,
- incorrect function annotation of a particular entry, and

One strategy to correct inconsistencies and errors in data representation is through biocuration process. Biocuration is the activity of organizing, representing and making biological information accessible to both humans and computers (Howe et al., 2008). For this purpose, the raw (uncurated) data extracted from UniProt was checked manually to ensure that they have the correct annotation by searching the literature. Some of the entries extracted from UniProt have wrong annotation attached to them especially the keyword. An example of an entry in UniProt with wrong keyword (KW) annotation is located at <http://www.uniprot.org/uniprot/P83141>. This is shown in Figure 2.5, where the KW line has the keyword antibiotic but the function line denoted by RT only mentions activity against *Phytophthora infestans*(fungi) but mentions nothing about activity against bacteria. The RT line talks about potent antifungal proteins, meaning that the protein has been experimented on, and the paper only proves antifungal activity but says nothing about the antibacterial activity. Another example of an entry in UniProt with wrong function annotation is of conolysin-Mt1 peptide (<http://www.uniprot.org/uniprot/P0C8S6>). This peptide has the following error in the function tag of the entry, i.e., the curators have introduced the term "Michael Jackson" which is not listed in the original article.

.. Intracranial injection causes mice to shuffle backward until the encounter an obstacle, at which time the mouse jump into the air. The backward shuffle is reminiscent to the signature dance 'moonwalk' that gained widespread popularity after being performed by Michael Jackson

Each annotation of the raw data was verified for its antimicrobial activity using published work. The final curated data set was used as an input for the MySQL database and the online version of the DAMPD database was uploaded in the link <http://apps.sanbi.ac.za/dampd/>. Supplementary material on biocuration of AMPs is found in the link <http://apps.sanbi.ac.za/dampd/biocuration.xls>. The data in the DAMPD database was complemented by additional functionalities to aid in analysis. This include graphical views, search utilities (keyword, family, taxonomy, species, citation, advance) and analytical tools (BLAST, ClustalW, HMMER and hydrocalculator). The process for creating the models using HMMER is discussed in the next section 2.3.3.

2.3.3 Building HMMER profiles for prediction of AMPs

An integrated antimicrobial peptide analysis tool called HMMER is created with the objective to infer AMP family of a query sequence. The HMMER program has three functionalities namely hmmbuild, hmmcalibrate and hmmsearch (Eddy, 1998). The HMMER tool has precompiled libraries of AMP family profiles by using "HMMER: Query Profile" option. Nevertheless, user can build tailored profiles based on

```

ID   AFP1L_MALPA          Reviewed;          15 AA.
AC   P83141;
DT   06-DEC-2002, integrated into UniProtKB/Swiss-Prot.
DT   01-DEC-2001, sequence version 1.
DT   05-APR-2011, entry version 25.
DE   RecName: Full=Antifungal protein 1 large subunit;
DE   AltName: Full=CW-1;
DE   Flags: Fragment;
OS   Malva parviflora (Little mallow) (Cheeseweed mallow).
OC   Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
OC   Spermatophyta; Magnoliophyta; eudicotyledons; core eudicotyledons;
OC   rosids; malvids; Malvales; Malvaceae; Malvoideae; Malva.
OX   NCBI_TaxID=145753;
RN   [1]
RP   PROTEIN SEQUENCE, AND FUNCTION.
RC   TISSUE=Seed;
RX   MEDLINE=20568734; PubMed=11118343; DOI=10.1006/bbrc.2000.3997;
RA   Wang X., Bunkers G.J.;
RT   "Potent heterologous antifungal proteins from cheeseweed (Malva
RT   parviflora).";
RL   Biochem. Biophys. Res. Commun. 279:669-673(2000).
CC   -!- FUNCTION: Possesses antifungal activity against P.infestans but
CC   not F.graminearum.
CC   -!- SUBUNIT: Heterodimer of a large and a small subunit.
CC   -!- MISCELLANEOUS: Antimicrobial activity is not affected by salt
CC   concentration.
CC   -----
CC   Copyrighted by the UniProt Consortium, see http://www.uniprot.org/terms
CC   Distributed under the Creative Commons Attribution-NoDerivs License
CC   -----
DR   GO; GO:0042742; P:defense response to bacterium; IEA:UniProtKB-KW.
DR   GO; GO:0050832; P:defense response to fungus; IDA:UniProtKB.
DR   GO; GO:0031640; P:killing of cells of other organism; IEA:UniProtKB-KW.
DR   GO; GO:0006805; P: xenobiotic metabolic process; IDA:UniProtKB.
PE   1: Evidence at protein level;
KW   Antibiotic; Antimicrobial; Direct protein sequencing; Fungicide.
FT   CHAIN             1       >15       Antifungal protein 1 large subunit.
FT                                     /FTid=PRO_0000064478.
FT   NON_TER           15       15
SQ   SEQUENCE          15 AA; 1783 MW; 2CB3079F53CC70F9 CRC64;
      VAGPFRIPPL RREFQ
//

```

Figure 2.5: Annotation error in a peptide with accession number P83141

their own sequences by choosing the option “HMMER: Build Profile”. HMMER profiles has been created out of mature peptide for different families. The procedure involved in building profiles is described as follows:

- each family protein sequence are aligned using ClustalW (Thompson et al., 1994).
- build HMM profile from the aligned sequences using hmmbuild module.

- calibrate the profile HMM using `hmmcalibrate` modules in order to increase sensitivity of the database search.

The profiles from the above procedure is saved as a specific AMP family library, for example `defensin.hmm`, `brevinin.hmm` etc.

2.3.4 Methodology for hydrocalculator tool

The hydrophobic residues are I, V, L, F, C, M, A and W. The percentage of hydrophobic residues of a peptide sequence (*seq*) is

$$\% \text{ of hydrophobic residues} = \frac{\text{Number of hydrophobic residues in } seq}{\text{Length of the sequence } (seq)} \quad (2.1)$$

The positively charged residues are I, V, L, F, C, M, A, W, R, H and K. The negatively charged residues are D and E. The remaining of the 20 amino acid residues are neutral. The net charge Q of a sequence is the summation of charges of each its residues.

Hydrophobicity is a fundamental attributes of amino acid residues that determines protein folding, protein subunits interactions binding to receptors and interactions of proteins and peptides with biological membranes (Tossi et al., 2002). The mean hydrophobicity \bar{H} of a sequence is given by

$$\bar{H} = \frac{\sum_{i=1}^n f^k(i)}{n}, \quad (2.2)$$

where

- n is the length of the primary protein sequence,
- i the i^{th} amino acid
- $f^k(i)$ is the value of the i^{th} amino acid of the respective k^{th} amino acid property,

The hydrophobic moment of a sequence *seq* gives an indication as to how the hydrophobicities of its constituent residues if a particular segment of the the sequences happens to be folded into particular conformation, i.e, α -helix or β -helix. The hydrophobic moment of a sequence is given by

$$M_H = \left\{ \left[\sum_{residue\ n} H_n \sin(n\sigma) \right]^2 + \left[\sum_{residue\ n} H_n \cos(n\sigma) \right]^2 \right\}^{\frac{1}{2}} \quad (2.3)$$

where

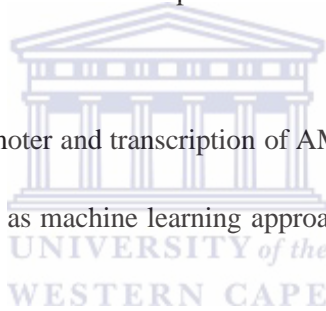
- $\sum_{residue\ n}$ is the summation over all residues of the sequence,
- H_n is the hydrophobicity of the n th residue,
- σ is the angle at which successive side chains emerge from the central axis of the secondary structure segment where $\sigma = 100$ for α -helix (Tossi et al., 2002).

The mean hydrophobic moment is M_H /sequence length.

2.4 Future work

Antimicrobial database is important for scientists in academia and industry. In order for the database to maintain its usefulness, regular updating and data enrichment with additional information on AMP is crucial. Nonetheless, more analytical tools inform experimentalist is needed. Therefore, some of the future work entails:

- furnishing information on promoter and transcription of AMP immunity genes
- including robust methods such as machine learning approach to aid in classification of AMPs into distinct families.



2.5 Summary

DAMPD is a database that has been built with the aim of making a comprehensive repository of experimentally validated AMPs complemented by search and analytical tools to help in extraction and analysis of AMPs. DAMPD has useful tools such as BLAST, ClustalW, SignalP, hydrocalculator and HMMER. The HMMER query profile module, enables users to predict the AMP families of a query sequence. It also assists in capturing of new peptide homologs from other public databases and laboratories.

Chapter 3

Prediction of AMPs using parameter optimized support vector machines

Abstract

Background: Antimicrobial peptides (AMPs) are important components of the innate immune systems of many species. The peptides can serve as a natural templates for the design of novel antibiotics. The number of uncharacterized proteins are increasing, there is need to develop robust computational techniques that can be used to mine new potential AMPs from the protein universe.

Methods: Support vector machine (SVM) is a classification technique that highly depends on certain hyperparameters that affects the classification accuracy. The aim of this study is to obtain the best hyperparameter values of SVM. In particular, three optimization methods, namely grid search (GS), pattern search (PS) and derivative-free simulated annealing (DFSA) are used to select SVM hyperparameters, denoted by GS-SVM, PS-SVM and DFSA-SVM respectively.

Results: The SVM models were created using two experiments, first based on the whole AMPs of a particular taxonomy (generalized model), second, based on family classification of each taxonomy (specialised model). Results indicates that DFSA-SVM method was the best overall with an accuracy of 97.95% using generalized model while PS-SVM is the overall best method with an accuracy of 99.25% using specialized model.

Conclusion: The selection of SVM hyperparameters is important in order to get useful models to predict AMPs. Prediction of AMPs using specialized models is more robust than generalized models.

3.1 Introduction

Antimicrobial peptides (AMPs) are an important component of the natural defense system of most living organisms against invading pathogens. They are widely distributed in eukaryotes and prokaryotes, such as bacteria, insects, plants, amphibians and viruses. They are relatively small in size, less than 10 kDa in size. These peptides either have cationic or amphiphatic forms with variable length, sequence and structures which contribute to the diversity of the AMPs. They play an important role in innate immunity and are the first line of defense (Hancock and Chapple, 1999; Wang and Wang, 2004).

The number of AMPs is increasing and there are well over four thousand peptides in UniProtKB (Bairoch and Apweiler, 2000) of which only 1232 are experimentally validated AMPs found in dragon antimicrobial peptide database (see chapter 2). Experimental methods used in characterizing AMPs are costly, time consuming and resource intensive. Thus, there is need to develop computational tool for predicting AMPs, in order to inform experimental approaches. Furthermore, identification of AMPs can serve as a natural template for designing novel antibiotics useful in combating or controlling diseases.

In the past, computational approaches have been designed to predict novel antimicrobial peptide from protein sequences. Recently, random forest has been applied in predicting antimicrobial peptides (Thomas et al., 2010). Artificial Neural Networks (ANN), Quantitative Matrices (QM) and Support Vector Machines (SVM) has been designed to predict antibacterial peptides (Lata et al., 2010, 2007). Quadratic discriminant analysis was used in classification of antimicrobial peptides using diversity measure with quadratic discriminant analysis (Chen and Luo, 2009). Fourier transform based method with property based coding strategy could be used to scan the peptide space for discovering new potential antimicrobial peptides (Nagarajan et al., 2006). Decision trees have been developed for classification of antimicrobial peptides (Lee et al., 2004).

To identify the AMPs computationally, a support vector machine (SVM) (Boser et al., 1992; Vapnik, 2000) is implemented. The SVM learn patterns based on examples and creates a model that classifies the positive and negative AMPs. The discriminative quality of the model depends on two hyperparameters of the SVM namely, trade off (c) and RBF kernel parameter (σ) (Duan et al., 2003). SVM has been used to predict AMPs as mentioned above. Nonetheless, no effort has been made to optimize the hyperparameters of the SVM, that ultimately improves the classification accuracy.

Several optimization methods have been suggested to select SVM model hyperparameters. For example, a hybrid of SVM with a genetic algorithm (Samanta et al., 2006) and simulated annealing (Lin et al.,

2008). However, these approaches are computationally expensive and suffer from slow convergence. In addition, direct search methods such as Nelder and Mead (Damaševičius, 2010) have been employed. One of the disadvantages of this method is that it lacks mathematical proof for convergence. Grid search has also been used to select SVM hyperparameters (Samanta et al., 2006). Grid search is computationally expensive for a larger number of parameters and the solution depends upon the coarseness of grid. Nonetheless, it lacks optimality criteria for solution (Damaševičius, 2010).

To ameliorate the above limitations of selecting hyperparameters, two approaches are utilized namely pattern search (PS) (Abramson et al., 2004; Audet et al., 2008; Kolda et al., 2003) and derivative-free simulated annealing (DFSA) (Gabere, 2007). Both PS and DFSA are recent direct search methods for local and global optimization respectively. The important feature of these methods, is that they guarantee mathematical convergence (Gabere, 2007; Torczon, 1991).

In this chapter, SVM is hybridized with three different optimization methods namely, GS, PS and DFSA, where the fundamental structure of SVM is kept intact. These hybrid methods are denoted as GS-SVM, PS-SVM and DFSA-SVM and are used to predict antimicrobial peptides in various taxa. The proposed hybrid methods, GS-SVM, PS-SVM and DFSA-SVM methods are shown to be efficient methods in predicting AMPs.

The chapter is organized as follows. Section 3.2 presents the algorithm, that is, support vector machine and proposed algorithms for optimizing SVM hyperparameters. Section 3.3 discusses the material used that is, dataset, multi-class strategy, feature representation, scaling and performance measure. Results are presented in section 3.4 followed by discussion in section 3.5. Finally, summary is made in section 3.6.

3.2 Algorithm

3.2.1 Support vector machines

The Support vector machine is a modeling technique that performs data classification by constructing an n -dimensional hyperplane that optimally separates the data into two classes (Cristianini and Shawe-Taylor, 2001; Vapnik, 2000). The input of an SVM is a training set

$$S = (\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$$

of vector of features $\vec{x}_i \in X$ together with their known classes $y_i \in \{-1, +1\}$. On the other hand, the output of SVM is a model

$$c : X \mapsto \{-1, +1\}$$

which predicts the class $c(\vec{x})$ of any new object $\vec{x} \in X$.

The essence in classification is to minimize the probability of error in using the trained classifier. This is referred to as the structural risk and SVM are able to minimize the structural risk using four fundamental concepts. These concepts are separating hyperplane, maximum-margin hyperplane, the soft margin and the kernel function (Noble, 2006).

The separating hyperplane is the division that separates two or more classes. In case of a one-dimensional problem, a single point can divide these classes. In a two dimensional case, the division is a line. For a three dimensional problem, the division is a plane and in general, a hyperplane in case of higher dimension. In searching for the maximum hyperplane, find a set of data point that are the most difficult points to classify. These data points are called support vectors. In constructing an SVM classifier, the support vectors are closest to the hyperplane and are located in the boundaries of the margin between the two classes. The maximum-margin hyperplane is the distance from the hyperplane to the nearest support vector. SVM selects the hyperplane by maximizing the margin between the support vector to the hyperplane while minimizing the structural risk. The trade-off parameter c controls the trade-off between separating margin and the error. The line shown in Figure 3.1 separates the two classes. However, in practical situations, datasets cannot be separated 100% as shown in the Figure 3.1 where there are misclassifications. SVM allows for a number of misclassification through constructing a soft margin. Therefore, introducing the soft margin requires the user to choose a parameter that controls misclassification of examples. This soft margin parameter c controls the trade off between allowing training errors and forcing rigid margins, i.e., creates a soft margin that permits some misclassification. Increasing the value of c increases the cost of misclassifying points and forces the creation of a more accurate model that may not generalize well. The hyperplane presented in Figure 3.1 that separates the two classes is linear. However, in most problems, this is not the case, as shown in Figure 3.2. This is an example of a non-separable one dimensional problem. In order to separate them, a kernel trick is required so that it can transform the one-dimensional problem into a two-dimensional problem as shown in Figure 3.3. In general, a kernel function projects non-separable data from a lower dimensional space to a higher dimensional space in order to make it separable using a hyperplane. There are several kernel tricks in SVM, namely, polynomial, radial basis function (rbf) and

sigmoid. In this thesis, a radial basis function kernel (rbf-kernel) is employed and is defined by

$$K(x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{\sigma^2}\right), \quad (3.1)$$

where x_i and x_j are the two vectors where one of them is a support vector and σ is an adjustable parameter that determine the area of influence of the support vector over the data space. Larger value of σ reduce the number of support vectors, since each support vector covers more data space.

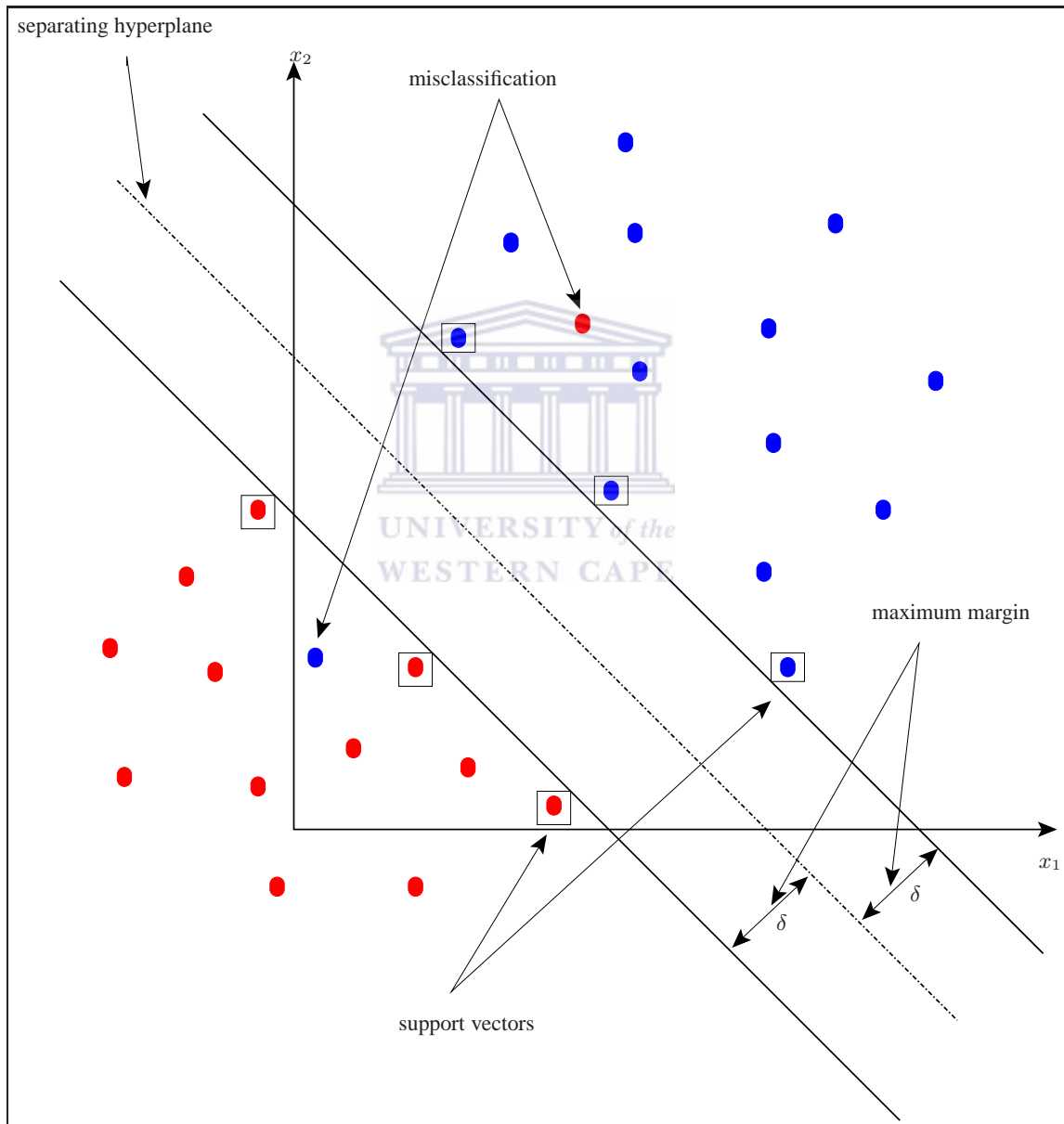


Figure 3.1: The figure illustrates positive and negative examples (represented by red and blue circle) in two dimensional space. The SVM learned the representation of a hyperplane, here illustrated through an enclosed rectangle that best separates the two classes of examples from each other. The examples that lie on the edge of the hyperplane (enclosed in a rectangle) are the so called support vectors (the actual representation learned by the SVM).

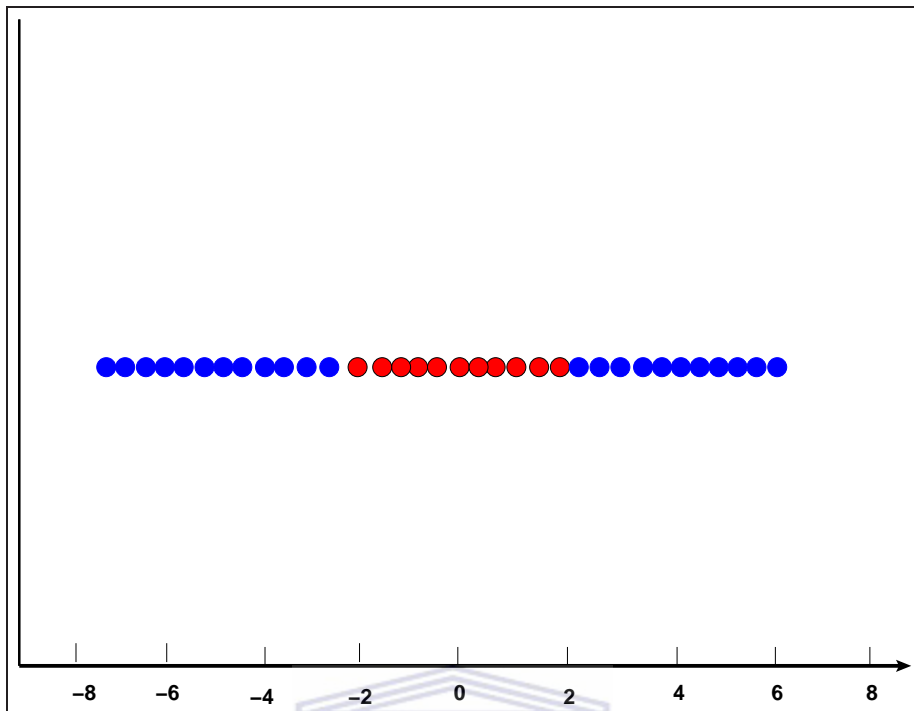


Figure 3.2: A non-separable one-dimensional problem (Noble, 2006).

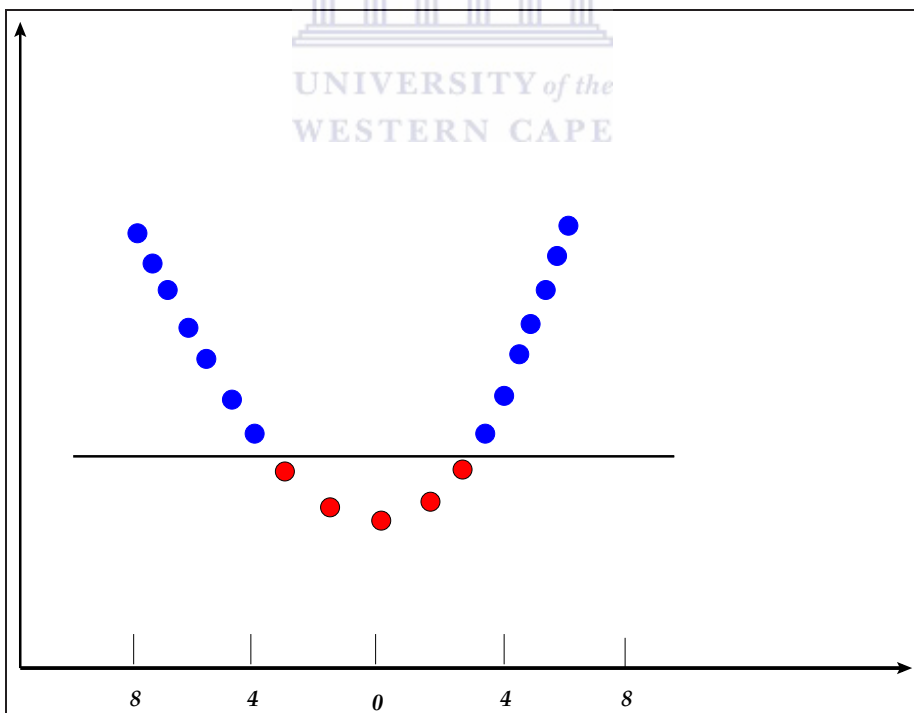


Figure 3.3: Separating the non-separable one-dimensional problem in Figure 3.2 using a kernel trick (Noble, 2006).

The SVM implementation used in the present study is SVM^{light} (Joachims, 1999). This program is freely downloadable from <http://svmlight.joachim.org/>. SVM^{light} has several hyperparameters which should be optimized in order to obtain a generative model. These hyperparameters include but are not restricted to the trade-off (c) and the RBF kernel parameter (σ). Optimization of these hyperparameters can be treated as a black box and hence the need for derivative-free optimization methods. In the next section, the three optimization methods for selecting the SVM hyperparameters are presented.

3.2.2 Proposed algorithms for optimizing SVM hyperparameters

The quality of an SVM model largely depends on the selection of the two model hyperparameters c and σ . Without loss of generality, the selection of these model hyperparameters can be considered as a global optimization problem, see Figure 3.4. The mathematical formulation of global optimization is defined as follows:

$$\boxed{\text{maximise } f(h) \text{ subject to } h \in \Omega,} \quad (3.2)$$

where $f : \Omega \subseteq \mathbb{R}^2 \rightarrow \mathbb{R}$ is a continuous real-valued function and $\Omega = \{(h_1, h_2) \in \mathbb{R}^2 \mid l_i \leq h_i \leq u_i, l_i, u_i \in \mathbb{R}\}$ is the hyperparameter search region. In this formulation, $h = (h_1, h_2)$ is a 2-dimensional vector of the two SVM hyperparameters c and σ , i.e., $h_1 = c$ and $h_2 = \sigma$. The hyperparameter search region is defined as

$$\Omega = \{(h_1, h_2) \in \mathbb{R}^2 \mid 2^{-5} \leq h_1 \leq 2^3, 2^{-15} \leq h_2 \leq 2^3\} \quad (3.3)$$

In this study, the objective function $f(h)$ is the test set performance accuracy of the model defined in equation (3.25).

In order to solve the problem defined in (3.2), three hybrid methods that combine SVM with either grid search (GS), pattern search (PS) or derivative-free simulated annealing (DFSA) (Gabere, 2007) are implemented. These hybrid methods optimize the SVM hyperparameters. The GS method is presented first then followed by PS and DFSA methods.

3.2.3 Grid search method

Optimization by grid search is described as follow. Once the parameter search space is defined, each parameter dimension is split into k parts. The intersections of the splits form a multidimensional grid. The value of the objective function defined in (3.2) is evaluated in each point of the grid and the global optimum is found. The coarseness of the grid depends on the grid step length Δ_{GS} used. The smaller

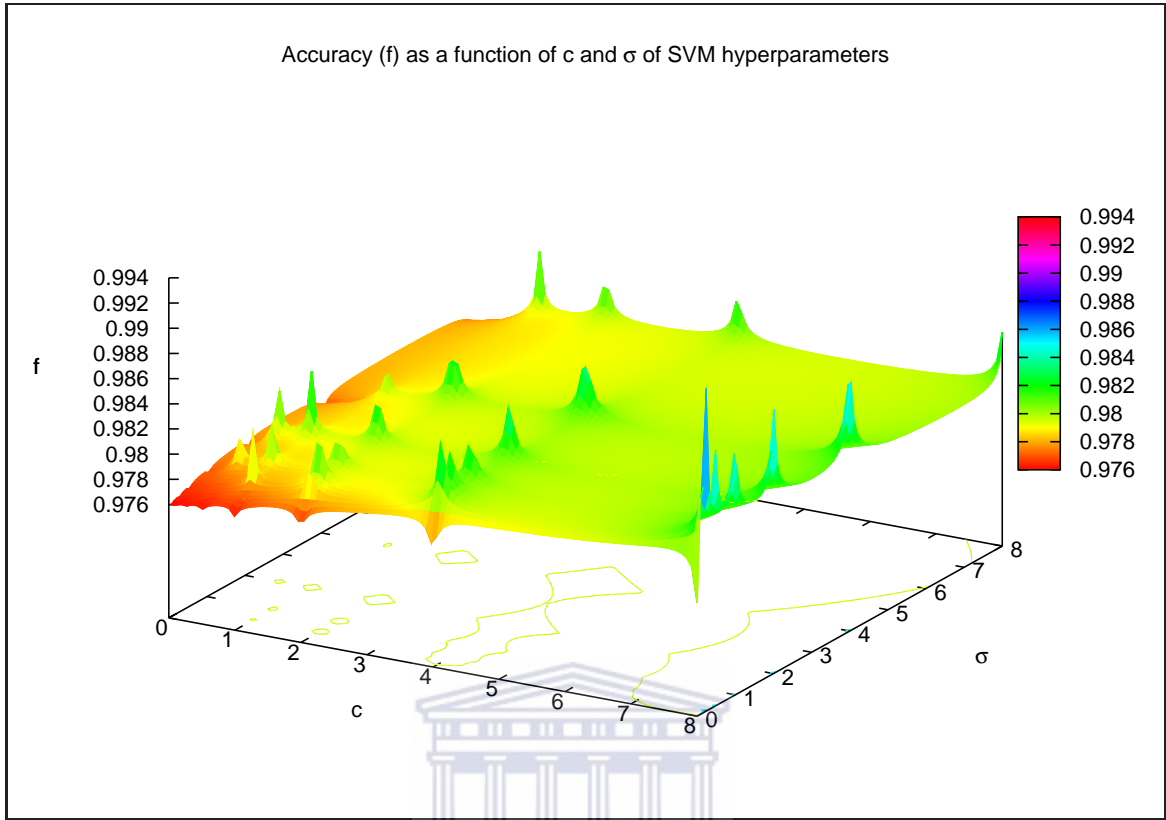


Figure 3.4: A mesh plot of the hyperparameters c and σ against the accuracy (f) using grid search

Δ_{GS} is the courser the grid and vice versa. An illustrative example of grid search method is shown in the Figure 3.5.

3.2.4 Pattern search method

PS is a derivative-free iterative local search procedure with convergence properties (Kolda et al., 2003). In its simplest form PS works as follows. Starting with an initial point x^k and an initial step length Δ^k , $k=0$, PS generates trial points around x^k (k being the iteration counter of PS) by successively using directions d^i , where d^i form the columns of the matrix

$$D = (d^1, \dots, d^n, d^{n+1}, \dots, d^{2n}) = (e_1, \dots, e_n, -e_1, \dots, -e_n), \quad (3.4)$$

e_i being the i^{th} unit coordinate vector. The trial points generated for each k are members of the poll set

$$P^k = \{p^i \in \mathbb{R}^n \mid p^i = x^k + \Delta^k d^i : d^i \in D, i = 1, \dots, 2n\}. \quad (3.5)$$

At each k^{th} iteration of PS, the i^{th} trial point p^i is examined so as to see if it is better than the current iterate x^k . If a point $p^i \in P^k$ such that $f(p^i) > f(x^k)$, then the trial point generation at the current poll

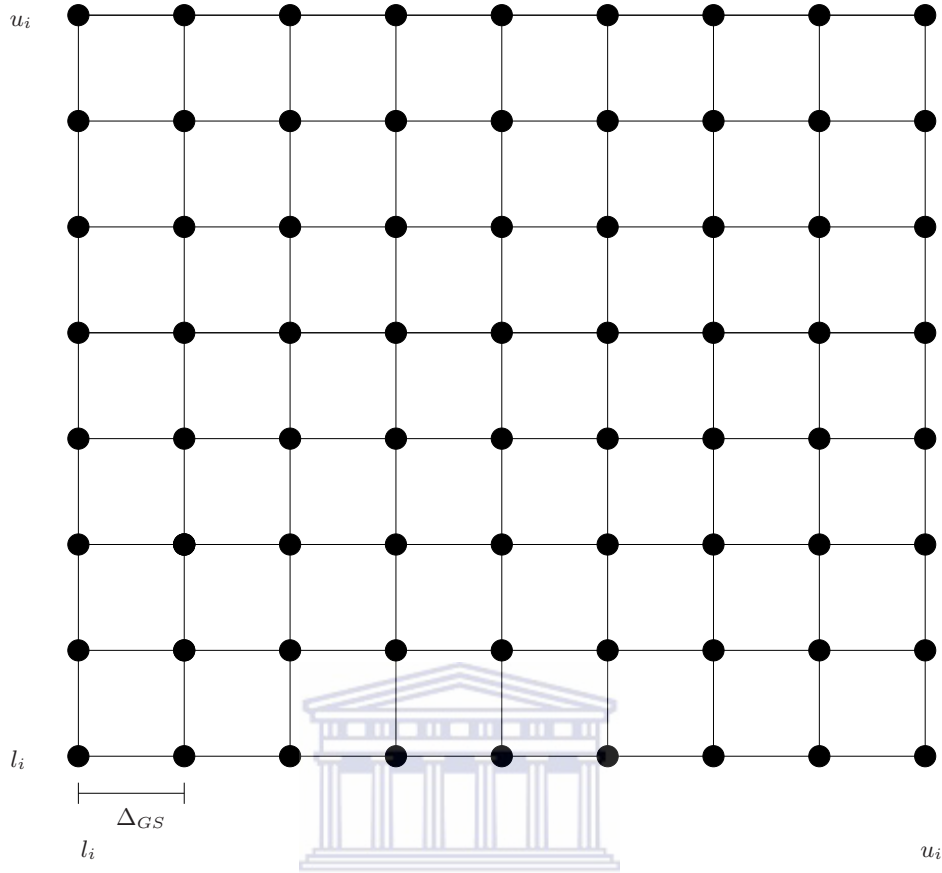


Figure 3.5: Grid Search in a two dimensional optimization problem

stops, the step length Δ^{k+1} is increased and a new poll starts at the new current iterate $x^{k+1} = p^i$. If $f(p^i) \leq f(x^k), \forall p^i \in P^k$ then the step length Δ^{k+1} is decreased and the current iterate is retained i.e. $x^{k+1} = x^k$. Therefore, the next iterate is updated as follows:

$$x^{k+1} = \begin{cases} p^i & \text{if } f(p^i) > f(x^k), \text{ for some } p^i \in P^k, \\ x^k & \text{otherwise.} \end{cases}$$

The step size parameter is updated (Kolda et al., 2003) as follows:

$$\Delta^{k+1} = \begin{cases} 2\Delta^k & \text{if } f(p^i) > f(x^k), \text{ for some } p^i \in P^k, \\ \frac{1}{2}\Delta^k & \text{otherwise.} \end{cases}$$

The above two updates continue until the step size parameter Δ^k gets sufficiently small (within the tolerance Δ^{tol}), thus ensuring convergence to a local maximum. Ali and Gabere (2010) described the step by step description of the basic PS and detailed below.

Algorithm 1: The PS algorithm.**1. Initialization:**

Initialize $x^k \in \Omega$ and $\Delta^k > 0$. Initialize D with j^{th} column being the direction d^j , $j = 1, 2, \dots, 2n$. Set $k = 0$ and $i = 1$. Set $\Delta^{\text{tol}} > 0$.

2. Trial point generation:

2(a) Evaluate $f(p^i)$ where $p^i = (x^k + \Delta^k d^i) \in P^k$, $d^i \in D$.

2(b) **If** $f(p^i) > f(x^k)$ **then** set $x^{k+1} = p^i$ and go to step 3.

Otherwise, set $i = i + 1$ and go to step 2(c).

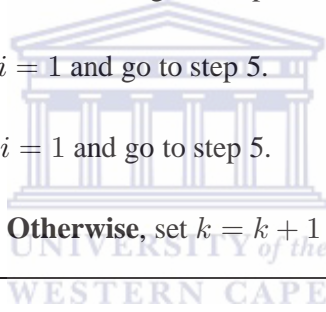
2(c) **If** $i \leq 2n$ **then** go to step 2(a).

Otherwise, set $x^{k+1} = x^k$ and go to step 4.

3. Update $\Delta^{k+1} = 2\Delta^k$. Set $i = 1$ and go to step 5.

4. Update $\Delta^{k+1} = \frac{1}{2}\Delta^k$. Set $i = 1$ and go to step 5.

5. **If** $\Delta^{k+1} < \Delta^{\text{tol}}$ **then** stop. **Otherwise**, set $k = k + 1$ and go to step 2.

**3.2.5 Derivative-free simulated annealing (DFSA)**

In this section, the full details of the hybrid method known as the derivative-free simulated annealing or DFSA in short is presented (Ali and Gabere, 2010). The structure of DFSA is similar to the simulated annealing algorithm proposed by (Dekkers, 1991). It uses similar distribution for generating the trial points. The only difference is that DFSA implements a gradient-free local technique. The local technique of DFSA selects uniformly a direction from a given set of directions. An important property of the set of directions is that at least one of the directions in the set is a descent direction at x . The main parts of the DFSA is described in the subsequent subsections, namely the generation mechanism and the cooling schedule of DFSA.

Generation mechanism

DFSA uses the following generation mechanism to generate trial points using the following probability distribution:

$$g_{xy} = \begin{cases} \frac{1}{m(\Omega)} & \text{if } \omega \leq \psi, \\ RD(x) & \text{if } \omega > \psi, \end{cases} \quad (3.6)$$

where ω is a random number in $(0, 1)$ and $\psi = 0.75$. $RD(x)$ (stands for random direction) is a local technique. $RD(x)$ generates the trial point y in the neighborhood of x . An important feature of $RD(x)$ is that only one function call is needed each time it is invoked. When $RD(x)$ is invoked at the t^{th} Markov chain (MC), the procedure of generating y from x is as follows. The trial point y is calculated by moving a step of length Δ_t^{sa} from x along the direction d , i.e.,

$$y = x + \Delta_t^{sa} d, \quad (3.7)$$

where $d \sim Unif\{d^1, \dots, d^n, d^{n+1}, \dots, d^{2n}\} \in D$, defined in equation (3.4). The step length Δ_t^{sa} is initialized as:

$$\Delta_0^{sa} = \zeta \max\{u_i - l_i \mid i = 1, \dots, n\}, \quad (3.8)$$

where $\zeta \in (0, 0.05)$ is a small parameter. The step length, Δ_t^{sa} , is updated at the end of each MC.

Updating of the step size parameter Δ_t^{sa} : The step length Δ_t^{sa} in GM varies with MC and is updated as follows: At the end of each t^{th} MC, the ratio ra is computed by

$$ra = \frac{nacp}{nops}, \quad (3.9)$$

where $nops$ is the number of times $RD(x)$ is invoked to generate trial points and $nacp$ is the number of times the trial points generated by $RD(x)$ are accepted in the t^{th} MC. The ratio, ra , determines whether to increase or decrease Δ_t^{sa} . Thus, the next step length Δ_{t+1}^{sa} to be used in the $(t+1)^{\text{th}}$ MC is updated as follows:

$$\Delta_{t+1}^{sa} = \begin{cases} (1 + \alpha)\Delta_t^{sa} & \text{if } ra \geq 0.6, \\ (1 - \alpha)\Delta_t^{sa} & \text{if } ra < 0.4, \\ \Delta_t^{sa} & \text{if } 0.4 \leq ra < 0.6, \end{cases} \quad (3.10)$$

where $\alpha \in (0, 0.2)$ is a parameter. The motivation for the above update can be found in (Gabere, 2007).

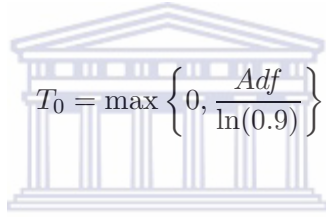
Cooling schedule for DFSA

The choice of a cooling scheduling has an important bearing on the performance of the DFSA algorithm. The cooling schedule suggested by Hedar and Fukushima (2004) is implemented. Generally, choosing a proper cooling schedule is not a trivial task. First, the initial temperature T_0 is set large enough to make the initial probability of accepting transition close to 1. Beside the initial point x , another point y is generated in a neighborhood of x to calculate T_0 as

$$T_0 = \frac{1}{\ln(0.9)} |f(y) - f(x)| \quad (3.11)$$

However, in this thesis, the initial temperature T_0 defined in equation (3.13) is modified and is calculated as follows:

$$Adf = \frac{\sum_{i=1}^z |f(y_i) - f(x_0)|}{z} \quad (3.12)$$



$$T_0 = \max \left\{ 0, \frac{Adf}{\ln(0.9)} \right\} \quad (3.13)$$

where

- x_0 is an initial point and y_i is another point generated randomly in the search space Ω ,
- f is the accuracy defined in equation (3.25),
- z is the number of sample points. In this case z is set to 100 sample points.

The length of Markov chain is generated using a fixed number of points (Dekkers, 1991), i.e.,

$$L = 10n. \quad (3.14)$$

In this thesis, the length of the Markov chain is set to $L = 10$.

The decrement rule for T_t : T_t is decreased at the end of each MC according to the equation (3.16) as suggested (Hedar and Fukushima, 2004).

$$T_{t+1} = T_t \times 0.9 \quad (3.15)$$

Stopping condition: The stopping condition proposed by Hedar and Fukushima (2004) is adopted. The DFSA algorithm is terminated after the temperature falls below a certain tolerance i.e.,

$$T_t \leq \min(10^{-3}, 10^{-3}T_0). \quad (3.16)$$

Description of the DFSA algorithm

In this section, the full details of the DFSA algorithm is presented. DFSA utilizes the point generation scheme defined in equation (3.6). In addition, DFSA keeps a record of the best point found during the search process using a singleton set S . The set S is updated when a better point found in the MC. DFSA initializes Δ_t^{sa} , $t = 0$, the initial point x and the cooling schedule before the beginning of the first MC. The set S initially contains the point $x_1^p = x$.

Structurally, like any other simulated annealing algorithm, the DFSA algorithm has two loops. The outer loop decreases the temperature and updates step length Δ_t^{sa} of $RD(x)$. The inner loop generates trial points in the MC using the generation mechanism defined in (3.6) and updates the best point found the moment a better point is found. Therefore, the set S contains the best point visited by the DFSA algorithm. The detailed structure of DFSA using a flowchart is shown in Appendix E. The step by step description of the DFSA algorithm is given below.

Algorithm 2: The DFSA algorithm.

1. **Initialization** : Generate an initial point x . Set $x_1^p = x$, $x_1^p \in S$. Set the temperature counter $t = 0$. Compute the initial temperature T_0 using equation (3.13). Calculate an initial step size parameter Δ_0^{sa} using equation (3.8).
2. **The inner and outer loops:**
 - while the stopping condition is not satisfied do
 - begin
 - for $i = 1$ to L do
 - begin
 - generate y from x using the mechanism in (3.6);
 - if $f(y) - f(x) \geq 0$ then accept;
 - else if $\exp((f(y) - f(x))/T_t) > \text{random}(0, 1)$ then accept;
 - if accept then $x = y$;
 - update the set S , i.e., if $f(x) > f(x_1^p)$ then $x_1^p = x$;**
 - end;
 - $t = t + 1$;
 - lower T_t using equation (3.16) ;
 - update Δ_t^{sa} using equation (3.10);**
 - end.

Note that the integration of PS and SVM is denoted as PS-SVM. Similarly for GS-SVM and DFSA-SVM. The main structure of the hybrid methods is represented in Figure 3.6 using a flowchart.

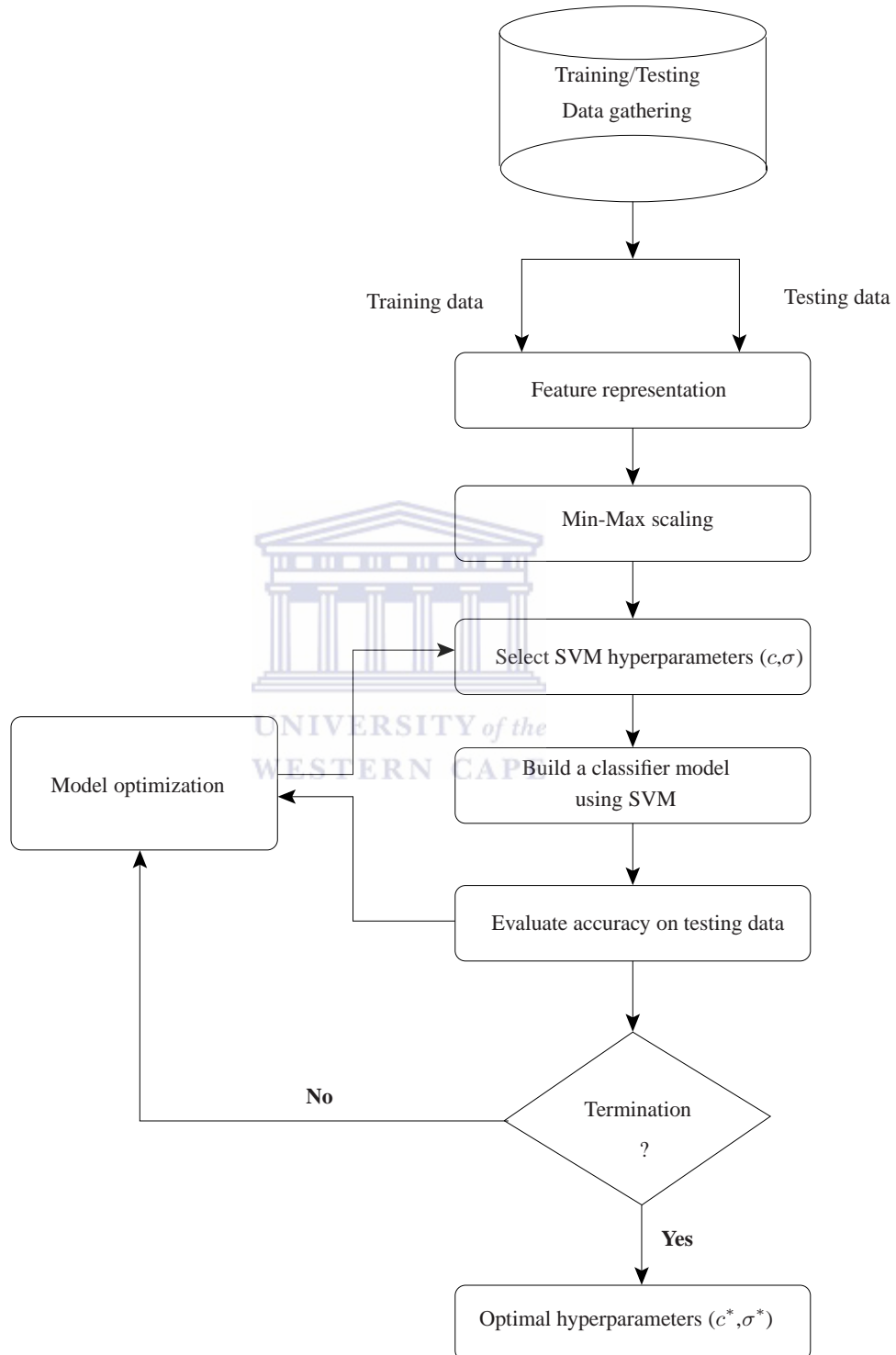


Figure 3.6: System architecture of the proposed optimization of SVM hyperparameters. The optimization method can either be GS, PS or DFSA.

3.2.6 Proposed PS-SVM algorithm

In this section, the details of the main hybrid method PS-SVM is elucidated. PS-SVM is a combination of the machine learning SVM and the pattern search method. The procedure of the proposed PS-SVM approach is shown in Algorithm 3. Structurally, PS-SVM consists of an initialization stage and pattern search stage and they are described as follows:

Initialization stage

In this stage, the algorithm initializes the starting step size Δ^k , the spanning set of direction D , step size tolerance Δ^{tol} . In addition, it generates n sample points (h_1, \dots, h_n) uniformly distributed over the search space Ω . Note that $h_i = (c_j, \sigma_j)$ for $j = 1, \dots, n$, is a sequence of SVM hyperparameters. For each of these n sample points, the training set X_{train} is trained with the hyperparameter $h_j, j = 1, \dots, n$ to obtain the predictors, i.e., $predictor^{(1)}, \dots, predictor^{(n)}$, respectively. The testing data is classified separately using each of the above n predictors ($predictor^{(1)}, \dots, predictor^{(n)}$) and their respective objective functions $f(h_j), j = 1, \dots, n$ is evaluated. Note that the objective function values $f(h_j)$ is the classification accuracy rate given in equation (3.25) of the testing set X_{test} given the classifier $predictor^{(j)}$. With these accuracy values $f(h_1), f(h_2), \dots, f(h_n)$, the best hyperparameter point h_{best} with the best maximum accuracy value f_{best} is selected. After the initialization stage, the PS is invoked and is explained below.

Pattern search stage

At this stage, the pattern search is invoked on each of the sample points $(h_1, h_2, \dots, h_n) \in \Omega$ generated in the above initialization stage. At each iteration, the point $h^k = h_j, k = 0, j = 1, 2, \dots, n$ are taken as the starting point. A poll step is initiated at the current point h^k by determining a trial point p^i given by

$$p^i = (h^k + \Delta^k d^i) \in P^k, d^i \in D \quad (3.17)$$

where h^k is the current iterate, Δ^k is the step size, d^i is a unit direction in

$$D = \begin{pmatrix} d^1 & d^2 & d^3 & d^4 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \quad (3.18)$$

Note that unit directions are $d^1 = (1, 0), d^2 = (0, 1), d^3 = (-1, 0)$, and $d^4 = (0, -1)$. The training set, X_{train} is trained using the the current point p^i .

The trial point p^i is examined by classifying the testing set X_{test} so as to determine if it is a better solution than the current iterate h^k . The PS samples $2n$ points (n is the dimension of the problem which is

2) in the search space in a fixed pattern, controlled by a step size Δ^k about the current incumbent h^k . The poll step calculates the accuracy values at these points, point by point. If a point is found to be better than the incumbent, then the new point becomes the incumbent at the next iteration and the stepsize parameter Δ^k is doubled. On the other hand, if the function values at all $2n$ points fail to produce a higher accuracy value than the accuracy value at the incumbent point, then the stepsize Δ^k is reduced by half. The search continues until the stepsize gets sufficiently small. More detailed and formal description of the PS-SVM method is shown in the Algorithm 3. The setting of parameters used in this algorithm will be discussed later in 3.4.1.

3.2.7 Proposed DFSA-SVM algorithm

The procedure for the second hybrid method DFSA-SVM is explained in this section. It consists of an initialization stage and the inner and outer loop of DFSA stage. The pseudocode of the proposed DFSA-SVM method is given in Algorithm 4.

Initialization stage

In this stage, the current temperature T is set to T_0 using equation (3.13). The step size parameter Δ_0^{sa} is initialized using equation (3.8). The initial feasible solution h_{best} is computed as follows. The search space Ω defined in equation (3.3) is divided into six regions respectively. The points on the boundaries of the regions are taken as possible solutions, hence there are 49 initial solutions to be tested (Lin et al., 2008). The best of the 49 solutions is assigned h_{best} and $x = h_{best}$ and $f(x) = f_{best}$.

Inner and outer loop of DFSA algorithm

In the inner and outer loop of DFSA process, an initial solution h is randomly generated from x using the generation mechanism of equation (3.6). The training set X_{Train} is trained with the hyperparameters h in order to obtain the model $predictor^{(i)}$. The testing data X_{test} is classified using $predictor^{(i)}$ and the function value, i.e., $f(h)$ is computed using the objective function value, that is, the classification accuracy rate of SVM given in equation (3.25). If the change $\Delta f_{xh} = f(h) - f(x)$ represents an increase in the value of the objective function then the new point h is accepted. If the change represents a decrease in the objective function value then the new point h is accepted using a Metropolis acceptance probability

$$A_{xh}(T_t) = \min\{1, \exp((f(h) - f(x))/T_t)\}. \quad (3.19)$$

This process is repeated for a large enough number of iterations for each T_t . A new Markov chain is then generated (starting from the last accepted point in the previous Markov chain) for a reduced temperature

until the algorithm stops. The algorithm for DFSA-SVM hybrid is sketched in Algorithm 4.

Algorithm 3: The PS-SVM algorithm.

0. Input:

X_{train} = training data

X_{test} = testing data

1. Initialization:

Initialize $\Delta^k > 0$.

Initialize D with j^{th} column being the direction $d^j, j = 1, 2, \dots, 2n$.

Set $k = 0$ and $i = 1$. Set $\Delta^{tol} > 0$.

Generate n random sample points $(h_1, h_2, \dots, h_n) \in \Omega$. Train SVM to obtain

$predictor^{(j)} = \text{svm_train}(X_{train}, h_j)$, where $j = 1, 2, \dots, n$. Compute classification accuracy, i.e., $f(h_j) = \text{svm_test}(X_{test}, predictor^{(j)})$, for $j = 1, 2, \dots, n$. Calculate

$$h_{best} = \arg \max_{h \in (h_1, \dots, h_n)} f(h) \text{ and } f_{best} = f(h_{best})$$

2. Pattern search:

Apply pattern search on each of the sample points generated above, i.e., $(h_1, h_2, \dots, h_n) \in \Omega$, from step 2.1 to 2.4 of the poll step. The initial point h^k for PS is set to $h^k = h_j$, where $j = 1, 2, \dots, n$.

2.1 Trial point generation:

2.1(a) $predictor^{(i)} = \text{svm_train}(X_{train}, p^i)$. Evaluate $f(p^i) = \text{svm_test}(X_{test}, predictor^{(i)})$, where $p^i = (h^k + \Delta^k d^i) \in P^k, d^i \in D$ defined in equation (3.18).

2.1(b) **If** $f(p^i) > f(h^k)$ **then** set $h^{k+1} = p^i$, **If** $f(p^i) > f_{best}$ **then** $f_{best} = f(p^i), h_{best} = p^i$ and go to step 2.2.

Otherwise, set $i = i + 1$ and go to step 2.1(c).

2.1(c) **If** $i \leq 2n$ **then** go to step 2.1(a).

Otherwise, set $h^{k+1} = h^k$ and go to step 2.3.

2.2 Update $\Delta^{k+1} = 2\Delta^k$. Set $i = 1$ and go to step 2.4.

2.3 Update $\Delta^{k+1} = \frac{1}{2}\Delta^k$. Set $i = 1$ and go to step 2.4.

2.4 **If** $\Delta^{k+1} < \Delta^{tol}$ **then** stop. **Otherwise**, set $k = k + 1$ and go to step 2.1.

Algorithm 4: The DFSA-SVM algorithm.**0. Input:**

X_{train} = training data and X_{test} = testing data

1. Initialization:

Set the temperature counter $t = 0$.

Compute the initial temperature T_0 using equation (3.13)

Calculate an initial step size parameter Δ_0^{sa} using equation (3.8)

Find the initial feasible solution h_{best} with accuracy value $f_{best} = f(h_{best})$. Initialize $x = h_{best}$ and $f(x) = f_{best}$

2. The inner and outer loops of DFSA algorithm:

while the stopping condition is not satisfied do

begin

for $i = 1$ to L do

begin

generate h from x using the mechanism in (3.6);

$predictor^{(i)} = svm_train(X_{train}, h)$ and $f(h) = svm_test(X_{test}, predictor^{(i)})$

if $f(h) - f(x) \geq 0$ then accept;

else if $\exp((f(h) - f(x))/T_t) > \text{random}(0, 1)$ then accept;

if accept then $x = h$;

if $f(x) > f_{best}$ then $h_{best} = x$ and $f_{best} = f(h_{best})$;

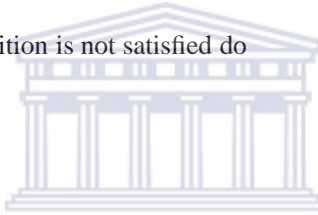
end;

$t = t + 1$;

lower T_t using equation (3.16) ;

update Δ_t^{sa} using equation (3.10);

end.



3.3 Material

3.3.1 Dataset

Two primary protein sequence sets are utilised in this study, an antimicrobial peptide dataset (positive set) and non-antimicrobial peptide dataset (negative set). The positive dataset consists of AMPs from different taxa. For the positive set, the mature part of the peptide is selected and the signal and propeptide sequence of the peptide are left out because it is the mature part that has an antimicrobial activity. The positive set was obtained from DAMPD (<http://apps.sanbi.ac.za/dampd>) and consisted of 1232 experimentally validated AMPs.

The model created for prediction purpose should be able to distinguish between positive and negative AMPs. Therefore, it is important to feed the machine learning classifier with a negative examples as well. The negative set consists of proteins belonging to various intracellular locations such as nucleus, cytoplasm, endoplasmic reticulum, golgi bodies and mitochondria (Lata et al., 2010). The negative set was downloaded from UniProt for each taxa and extracted sequences only with length varying from 5 to 100, because majority of the AMPs have length in this range. The number of negative set consisted of 4724 protein sequences. The keyword used to extract the negative set was

((golgi OR cytoplasm OR endoplasmic reticulum OR mitochondria) NOT antimicrobial) AND length:
[2 to 100] AND taxonomy: "name of taxonomy"),

where the name of taxonomy can be amphibian, mammalia or insecta. See the supplementary material B for the specific keywords used to extract negative sequences for each taxa. These dataset (both positive and negative sets) were purged to remove redundancies and they contain only those sequences which have 90% sequence similarity. Any sequence which have more than 90% sequence similarity is removed from the dataset by using CD-HIT software (Li and Godzik, 2006; Li et al., 2002). The reason for purging is to ensure that the problem is not easy. Therefore, 742 positive examples and 2134 negative examples remained after purging. The distribution of the sequence length of positive set, negative set and the combined (positive and negative set) are shown in Figure 3.7.

After purging, the numerical matrix of features were generated for both positive and negative set using amino acid composition and physicochemical properties defined in equation (3.20) and (3.21) respectively.

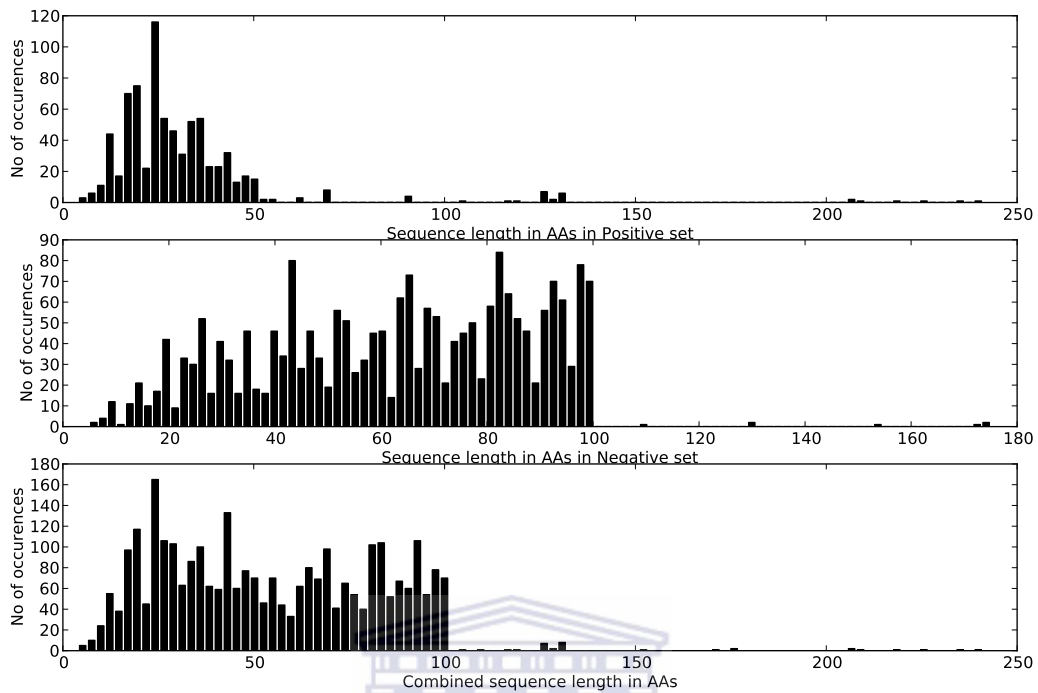


Figure 3.7: Histogram of sequence length distribution of positive set (top), negative set (middle) and combined (positive and negative sets) (down)

UNIVERSITY of the
WESTERN CAPE

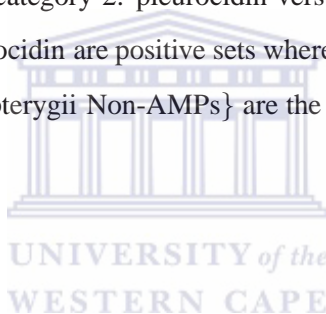
3.3.2 Multi-class strategy

Multi-class SVM was employed to determine the prediction models. The models were developed to predict AMPs belonging to different AMPs families across nine taxa. The AMP families in each of the nine taxa are shown below:

- actinopterygii (families: grammistin,pleurocidin)
- amphibian (families: aurein-citropin, bombinin, brevinin 1, brevinin 2, caerin, dermaseptin, esculentin, gastrin, phylloseptin, temporin, uperin)
- mammalia (families: alpha-defensin, beta-defensin,cathelicidin, cathelin related, glycosyl hydrolase 22, histone H2B)
- insecta (families: AMP insect, apidaecin, attacin, cecropin, invertebrate defensin, crabolin, mastoporan, ponericin 1, ponericin 2)

- arachnida (families: cupiennin, cytoinsectoxin, laticarin, oxyopinin, scorpion NDBP5)
- bacteria (families: bacteriocin, lantibiotic, thiocillin)
- crustacea (families: penaedin)
- merostomata (families: tachyplesin)
- plantae (families: DEFL, thaumatin)

For each taxa, a multi-class SVMs are created, that is one-versus-rest (OVR) by constructing k binary SVM classifiers namely, category 1 (positive set) versus all other categories (negative set), category 2 versus all the other categories, \dots , category k versus all other categories. For argument sake, for the case of actinopterygii taxonomy, the OVR is determined as: category 1: grammistin versus {pleurocidin and actinopterygii Non-AMPs} and category 2: pleurocidin versus {grammistin and actinopterygii Non-AMPs}, where grammistin and pleurocidin are positive sets whereas {pleurocidin and actinopterygii Non-AMPs} and {grammistin and actinopterygii Non-AMPs} are the negative set for category 1 and category 2 respectively.



3.3.3 Feature representation

The positive and negative dataset are converted into numeric representation which becomes the input for the SVM training process. An amino acid composition coupled with seven physicochemical properties namely, Eisenberg hydrophobicity scale, Hoop-Woods hydrophilicity, electron-ion interaction potential (Veljkovic), hydrophobicity (Zimmerman), bulkiness (Zimmerman) and polarity (Zimmerman) and kyte and doolittle hydrophobicity are adopted. These scales are obtained from AAindex (amino acid index database) (Kawashima et al., 2008). Therefore these spans an input vector of 27 features where the first 20 features comes from the amino acid composition. Note that the properties discussed in section 1.1 of Chapter 1 were not used in feature calculation except for hydrophobicity. Hydrophobic moment which is a measure of amphipathicity is not used because it requires knowledge of a particular conformation, i.e., α -helix and β -helix. Charge is not used because some AMPs are negatively charged. Conformation requires information regarding secondary structure. The amino acid composition a_j of protein sequences is computed using equation (3.20):

$$a_j = n_j/N \quad (3.20)$$

where j can be any of the 20 amino acids, N is the length of the sequence and n_j is the number of j^{th} amino acids. Therefore, for any given protein sequence, the amino acid composition calculations yield a fix length vector of 20 values. The remaining 7 features out of the possible 27 features are computed as follows. The feature representation $F(seq)$ for a sequence seq consists of 7 features \bar{f}^k , each representing the average of one of the 7 properties k over its amino acid sequence, $F(seq) = (\bar{f}^1, \bar{f}^2, \dots, \bar{f}^7)$, with $k = 1, \dots, 7$. An individual feature \bar{f}^k for amino acid property k is computed in equation (3.21) below:

$$\bar{f}^k = \frac{\sum_{i=1}^n f^k(i)}{n}, \quad (3.21)$$

where n is the length of the primary protein sequence, i the i^{th} amino acid and $f^k(i)$ is the value of the i^{th} amino acid of the respective k^{th} amino acid property, $k = 1, \dots, 7$. $k = 1, \dots, 531$ and $i = 1, \dots, 20$

Thus, each sequence, disregarding the length of its amino acid sequence is represented with the same length of feature representation. If an amino acid in the sequence is either "X" or "U", then the amino acid was disregarded from the averaging process.

3.3.4 Min-Max Scaling

The feature values may differ considerably, e.g., one feature of a peptide may be large while counts of other features may be small integer values. In order to standardize the contribution of each feature, the features have to be scaled within the interval $(0, 1]$. The scaling technique employed here is commonly known as min-max scaling and is described as follows:

1. For all values v_f of feature f over all examples, find the minimum value $v_{f_{min}}$ and the maximum value $v_{f_{max}}$
2. For an individual value w_f and feature f , calculate the new scaled value $w_{f_{scaled}}$ as

$$w_{f_{scaled}} = \frac{w_f - v_{f_{min}}}{v_{f_{max}} - v_{f_{min}}} \quad (3.22)$$

This results in a scaling for each feature $w_{f_{scaled}}$ is in $0 \leq w_{f_{scaled}} \leq 1$. When a model is utilised that was trained on the scaled training set, to classify examples of a test set, then the values have to be scaled before classification according to the minimum and maximum values for each feature found when scaling the training set. Thus, the scaled values of the testing set do not necessarily within the interval of zero and one but are scaled according to minimum and maximum value of the training set, to allow effective classification of the SVM.

3.3.5 Classifier performance measure

Several classifier measures are used to judge the performance of a classification system that is based on machine learning. Considering the confusion matrix presented in Figure 3.8, TP represents correctly predicted positive examples (AMPs), TN is correctly predicted negative examples (non-AMPs), FP is the number of non-AMPs examples wrongly predicted as AMPs and FN is the number of AMPs wrongly predicted as non-AMPs. The measure used in assessing the performance of the model are sensitivity, specificity, accuracy and Mathew's correlation coefficient (MCC) are described as follows:

		Predicted class	
		Positive	Negative
Actual class	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

UNIVERSITY of the
WESTERN CAPE

Figure 3.8: Confusion matrix

- **Sensitivity** is the percentage of AMPs (positive examples) correctly predicted as AMPs (positive). The sensitivity (recall) is defined as:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100 \quad (3.23)$$

- **Specificity** is the percentage of non-AMPs (negative examples) correctly predicted as non-AMPs (negative). The specificity is defined as:

$$\text{Specificity} = \frac{TN}{TN + FP} \times 100 \quad (3.24)$$

- **Accuracy** is the percentage of correctly predicted peptides (AMPs and non-AMPs). The accuracy is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \quad (3.25)$$

- **Mathews correlation coefficient (MCC)** is a measure of both sensitivity and specificity. $MCC = 0$ indicates completely random prediction, while $MCC = 1$ indicates perfect prediction. It is defined as:

$$MCC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}} \quad (3.26)$$

3.4 Numerical Results

In this section, the numerical results for the three hybrid methods, namely, GS-SVM, PS-SVM and DFSA-SVM discussed in section 3.2.3, 3.2.6 and 3.2.7 are presented. In the first subsection, the parameter settings of GS-SVM, PS-SVM and DFSA-SVM is presented. In the second subsection, the detailed description of the models from two experiments using three hybrid methods are presented. In the third subsection, the results of three methods based on an independent data set using the models derived from the leave-one-out approach are presented.

3.4.1 Parameter settings

The parameter setting used to carry out the experiment is given below. The initial step size parameter Δ_0 is set $\Delta_0 = 1$ for both PS-SVM and DFSA-SVM. The number of sample points generated in PS-SVM is set to $n = 10$. The parameter for c and σ used in GS-SVM were tested on an exponential growing sequence ($c \in \{2^{-5}, 2^{-4}, \dots, 2^3\}$, $\sigma \in \{2^{-15}, 2^{-14}, \dots, 2^3\}$). In total, GS-SVM used a combination of 114 parameters. As for PS-SVM and DFSA-SVM, since c and σ take a large range of values, a log scale was used to cover such a large region. The transformation is defined as $c_t = \log c$ and $\sigma_t = \log \sigma$. Thus PS-SVM and DFSA-SVM scan the space with the range $-3.47 \leq c_t \leq 2.08$ and $-10.4 \leq \sigma_t \leq 2.08$ (Momma and Bennett, 2002). PS-SVM was terminated when the step size parameter Δ^{tol} decreased below a certain tolerance, Δ^{tol} , i.e., when $\Delta_k < \Delta^{tol} = 0.001$. The spanning set of directions used by both PS-SVM and DFSA-SVM is denoted by D . The parameters for PS-SVM and DFSA-SVM are tabulated in Table 3.1 and Table 3.2.

3.4.2 Generating AMPs models for GS-SVM, PS-SVM and DFSA-SVM

In this section, the details of two experiments for creating AMP models using GS-SVM, PS-SVM and DFSA-SVM are presented. Two experiments were conducted to train the dataset using GS-SVM, PS-SVM and DFSA-SVM. First, training was conducted using all AMPs in each of the nine taxa as a positive

Table 3.1: PS-SVM parameters

Parameter	Definition	Value
Δ_0	Initial stepsize	1
Δ^{tol}	Termination tolerance	0.001
D	Spanning direction	$\begin{Bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{Bmatrix}$

Table 3.2: DFSA-SVM parameters

Parameter	Definition	Value
Δ_0	Initial stepsize	1
ζ	Δ_0^{sa} fraction	0.01
L	Length of Markov chain	10
α	Stepsize expansion factor	0.15
T_0	Initial temperature	$\max\{0, \frac{Adf}{\ln(0.9)}\}$
T_{min}	minimum temperature	$\min\{10^{-3}, 10^{-3}T_0\}$
Δ_0^{sa}	Random direction stepsize	$0.01 * \max\{u_i - l_i i = 1, \dots, n\}$
D	Spanning direction	$\begin{Bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{Bmatrix}$

example and their respective non-AMPs as negative examples. For instance, in insecta taxonomy, all AMPs in insecta are taken as positive examples and its corresponding insect non-AMPs are taken as a negative set. The negative set was presented in section 3.3.1. Second, training was performed using a multi-class classification, that is, one-vs-rest strategy on each and every AMP family of a particular taxa as explained in section 3.3.2. Note that in both experiments, the model selection is based on leave-one-out cross validation approach and the scaling of the testing set is according to the maximum and minimum values of the training set. The partitioning of the total dataset (positive and negative examples) is as follows: half of the total dataset was allotted to the training set. The remaining half was assigned to the testing set and the balance apportioned to the blind set.

SVM was trained using the training and testing sets of the two experiments to obtain generalized model (experiment one) and specialized model (experiment two). The generalized and specialized models created in both experiments were tested on a blind set and the results are presented first for experiment one and then experiment two, in the next subsection.

The hyperparameters selection (model) and the performance evaluation process ran on a Linux Pentium 4 core duo machine with a 1.8GHz CPU and 2GB memory in roughly 6 hours for the whole simulation.

3.4.3 Evaluating the performance of GS-SVM, PS-SVM and DFSA-SVM on a blind set

After creating the generalized and specialised models of the two experiments discussed in the previous subsection, it is imperative as an acid test to evaluate the performance of the prediction models on an independent (blind) set of AMPs and non-AMPs. Note that the independent set was not used for developing the models either in training and testing. Therefore, in this subsection, the AMPs models created by GS-SVM, PS-SVM and DFSA-SVM for the two experiments presented in the previous subsection are evaluated on a blindset. The results for the first experiment, that is based on generalized AMP model of each taxa is discussed first, thereafter the results for the second experiment, which is based on specialized AMP family model.

Evaluating the performance of GS-SVM, PS-SVM and DFSA-SVM on a blindset based on experiment one

The prediction results for the first experiment (generalized model) on a blindset are presented in Table 3.3 to Table 3.11. In these tables, the first column, "Algorithm" designates the AMPs in a particular taxa enclosed with the hybrid SVM method utilised. For example, in Table 3.3, actinopterygii (GS-SVM) denotes that actinopterygii AMP model was created using GS-SVM method. The second column labelled "model" indicates the value of (c, σ) , where c is the trade-off parameter and σ is the RBF kernel parameter of SVM. The third column, "# of peptides" designate the number of positive blind set and the number of negative blind set enclosed in brackets. Referring to the same table, the number of blind set in actinopterygii (GS-SVM) is 3(55) meaning that the number of data utilised as positive blind set is 5; negative blind set is 55. The remaining columns are sensitivity, specificity, accuracy, and MCC as defined in section 3.3.5.

In Table 3.3 to Table 3.11, the accuracies achieved by the classification model based on GS-SVM were 100%, 95.6%, 100%, 100%, 94.4%, 93.1%, 96.3% 100% and 98.5% for actinopterygii, amphibian, arachnida, bacteria, crustacea, insecta, mammalia, merostomata and plantae respectively. On the other hand, the accuracies achieved by the prediction model based on PS-SVM were 100%, 96.5%, 100%, 95.8%, 94.4%, 95.8%, 95.8%, 100% and 98.5% for actinopterygii, amphibian, arachnida, bacteria, crustacea, insecta, mammalia, merostomata and plantae respectively. As for DFSA-SVM, the accuracies achieved by the classification model were 100%, 96.5%, 100%, 100%, 94.4%, 95.8%, 96.3%, 100% and 98.5% for

actinopterygii, amphibian, arachnida, bacteria, crustacea, insecta, mammalia, merostomata and plantae respectively . Generally, the overall performance of GS-SVM, PS-SVM and DFSA-SVM was generally good.

Table 3.3: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of actinopterygii AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Actinopterygii (GS-SVM)	(8.000, 0.500)	3 (55)	100.0	100.0	100.0	1.0
Actinopterygii (PS-SVM)	(4.617, 0.098)	3 (55)	100.0	100.0	100.0	1.0
Actinopterygii (DFSA-SVM)	(2.219, 0.103)	3 (55)	100.0	100.0	100.0	1.0

Table 3.4: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of amphibian AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Amphibian (GS-SVM)	(2.000, 0.125)	91 (26)	100.0	80.8	95.6	0.9
Amphibian (PS-SVM)	(3.596, 0.098)	91 (26)	100.0	84.6	96.5	0.9
Amphibian (DFSA-SVM)	(1.261, 0.195)	91 (26)	100.0	84.6	96.5	0.9

Table 3.5: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of arachnida AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Arachnida (GS-SVM)	(8.000, 1.000)	7 (13)	100.0	100.0	100.0	1.0
Arachnida (PS-SVM)	(4.617, 0.098)	7 (13)	100.0	100.0	100.0	1.0
Arachnida (DFSA-SVM)	(0.431, 0.661)	7 (13)	100.0	100.0	100.0	1.0

Table 3.6: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of bacteria AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Bacteria (GS-SVM)	(8.000, 1.000)	13 (11)	100.0	100.0	100.0	1.0
Bacteria (PS-SVM)	(4.617, 0.098)	13 (11)	92.3	100.0	95.8	0.9
Bacteria (DFSA-SVM)	(0.769, 0.498)	13 (11)	100.0	100.0	100.0	1.0

Table 3.7: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of crustacea AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Crustacea (GS-SVM)	(8.000, 8.000)	1 (17)	0.0	100.0	94.4	0.0
Crustacea (PS-SVM)	(4.617, 0.098)	1 (17)	0.0	100.0	94.4	0.0
Crustacea (DFSA-SVM)	(7.992, 0.291)	1 (17)	0.0	100.0	94.4	0.0

Table 3.8: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of insecta AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Insecta (GS-SVM)	(8.000, 2.000)	31 (44)	96.4	90.9	93.1	0.9
Insecta (PS-SVM)	(7.992, 1.504)	31 (44)	96.4	95.5	95.8	0.9
Insecta (DFSA-SVM)	(1.387, 1.362)	31 (44)	96.4	95.5	95.8	0.9

Table 3.9: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of mammalia AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Mammalia (GS-SVM)	(8.000, 1.000)	24 (167)	90.9	97.0	96.3	0.8
Mammalia (PS-SVM)	(4.617, 1.192)	24 (167)	86.4	97.0	95.8	0.8
Mammalia (DFSA-SVM)	(7.992, 1.190)	24 (167)	90.9	97.0	96.3	0.8

Table 3.10: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of merostomata AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Merostomata (GS-SVM)	(8.000, 8.000)	1 (8)	100.0	100.0	100.0	1.0
Merostomata (PS-SVM)	(4.617, 0.098)	1 (8)	100.0	100.0	100.0	1.0
Merostomata (DFSA-SVM)	(1.871, 0.324)	1 (8)	100.0	100.0	100.0	1.0

Table 3.11: Performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of plant AMPs

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Plant (GS-SVM)	(8.000, 0.500)	11 (189)	72.7	100.0	98.5	0.8
Plant (PS-SVM)	(1.446, 0.635)	11 (189)	72.7	100.0	98.5	0.8
Plant (DFSA-SVM)	(5.233, 0.560)	11 (189)	72.7	100.0	98.5	0.8

Evaluating the performance of GS-SVM, PS-SVM and DFSA-SVM on a blind set based on experiment two

The prediction results for experiment two (specialized models) on the blind set are presented in Table 3.12 to Table 3.20. The column headings for each table is the same as mentioned in the previous tables, except for the first column heading labelled "Algorithm" and the third column "# of peptides". The column "Algorithm" designates the "AMP family" enclosed by the hybrid SVM method employed. For instance, in Table 3.12, Grammistin (**GS-SVM**) means that grammistin AMP family model was created using GS-SVM hybrid approach. Similarly for Grammistin (**PS-SVM**) and Grammistin (**DFSA-SVM**). The second column "# of peptides" designate the number of positive blind set and the number of negative blind set enclosed in brackets. Note that the negative set in experiment two differs from experiment one in that experiment one, the negative set consist of only non-AMPs. However, in experiment two, the negative set consists of non-AMPs and AMPs. For example, in Table 3.12, the "# of peptides" for Grammistin (GS-SVM) is 1(57), where 1 indicates one AMP from grammistin family and 57 consists of 55 non-AMPs and 2 AMPs from pleurocidin family. This is because of multi-class arrangement based on one-vs-rest.

The tabulated results of classification actinopterygii AMPs into the listed families is given in Table 3.12. The hybrid methods GS-SVM and PS-SVM were the overall best in terms of accuracy. For the classification of actinopterygii AMPs into grammistin and pleurocidin families, their respective accuracies were both 100% using GS-SVM and PS-SVM methods.

Table 3.12: Classification of actinopterygii antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Grammistin (GS-SVM)	(8.000, 8.000)	1 (57)	100.0	100.0	100.0	1.0
Grammistin (PS-SVM)	(1.741, 0.494)	1 (57)	100.0	100.0	100.0	1.0
Grammistin (DFSA-SVM)	(7.992, 0.160)	1 (57)	100.0	100.0	100.0	1.0
Pleurocidin (GS-SVM)	(8.000, 0.250)	2 (56)	100.0	100.0	100.0	1.0
Pleurocidin (PS-SVM)	(6.905, 0.246)	2 (56)	100.0	100.0	100.0	1.0
Pleurocidin (DFSA-SVM)	(7.992, 0.304)	2 (56)	50.0	100.0	98.3	0.7

The classification of amphibian AMPs into eleven disjoint families is presented in Table 3.13. DFSA-SVM performed better than GS-SVM in predicting dermaseptin while GS-SVM performed better than DFSA-SVM in predicting phylloseptin and uperin. PS-SVM outperformed GS-SVM in classifying brevinin 2 in terms of accuracy measure. The three hybrid methods were 100% sensitive in discriminating AMPs

in aurein-citropin, brevin 1, esculentin, gastrin and temporin from the negative set.

The confusion matrix for prediction of amphibian AMPs using GS-SVM model is shown in Figure 3.9. Each row in a matrix explains how examples in an AMP family are classified by the hybrid algorithm. For example, out of the eight independent examples in aurein-citropin, GS-SVM model classified eight of them correctly as aurein-citropin. Note that the lightness of a cell indicates the percentage of examples assigned to the cell as shown in the gradient colour palette. Therefore, the lighter the diagonal of a confusion matrix, the more accurate the corresponding algorithm. In Figure 3.10, that is, prediction of amphibian AMPs using PS-SVM model, one of uperin examples were misclassified as aurein-citropin and two of them were classified correctly as uperin. The confusion matrix for prediction of amphibians AMPs using DFSA-SVM model is shown in Figure 3.11.

The prediction of arachnida AMPs into cupiennin, cytoinsectoxin, latarcin, oxyopinin and scorpion NDBP5 are presented in Table 3.14. The three SVM hybrid methods performed equally in predicting AMP families for cupienin, cytoinsectoxin, latarcin, oxyopinin, except for scorpion NDBP5. DFSA-SVM outperformed GS-SVM and PS-SVM, in predicting scorpion NDBP5. In Table 3.15, the overall best performers are PS-SVM and DFSA-SVM in discriminating AMPs from non-AMPs in bacteria taxonomy.

Table 3.16 shows the classification of insects AMPs. GS-SVM outperforms PS-SVM and DFSA-SVM in predicting cecropin in terms of accuracy. The three hybrid methods failed dismally in the classification of AMPs in attacin family.

Classification of mammalia AMPs, the three hybrid methods performed equally as shown in Table 3.17. However, the sensitivity for the three hybrid methods were 50% in prediction of cathelicidin AMPs. The three hybrid methods performed well in predicting merostomata AMP families, though the positive set consisted of only one sequence as shown in Table 3.18.

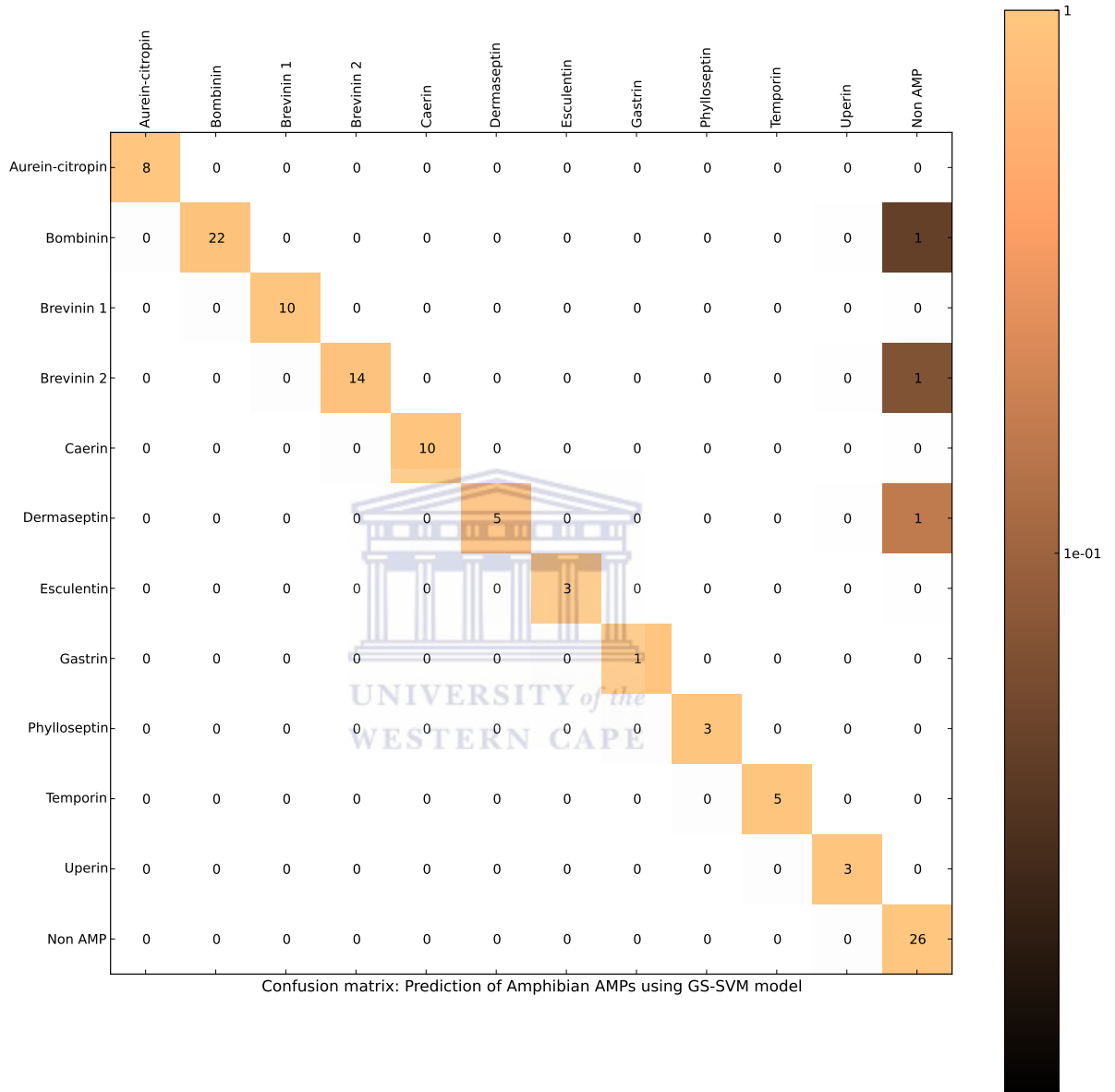


Figure 3.9: Confusion matrix for prediction of amphibian AMPs using GS-SVM model. Each row in a matrix explains how examples in a particular independent set are classified by an algorithm. The matrix is read row-wise. For example, out of 23 examples in bombinin, 22 were classified as bombinin and 1 as non-AMP.

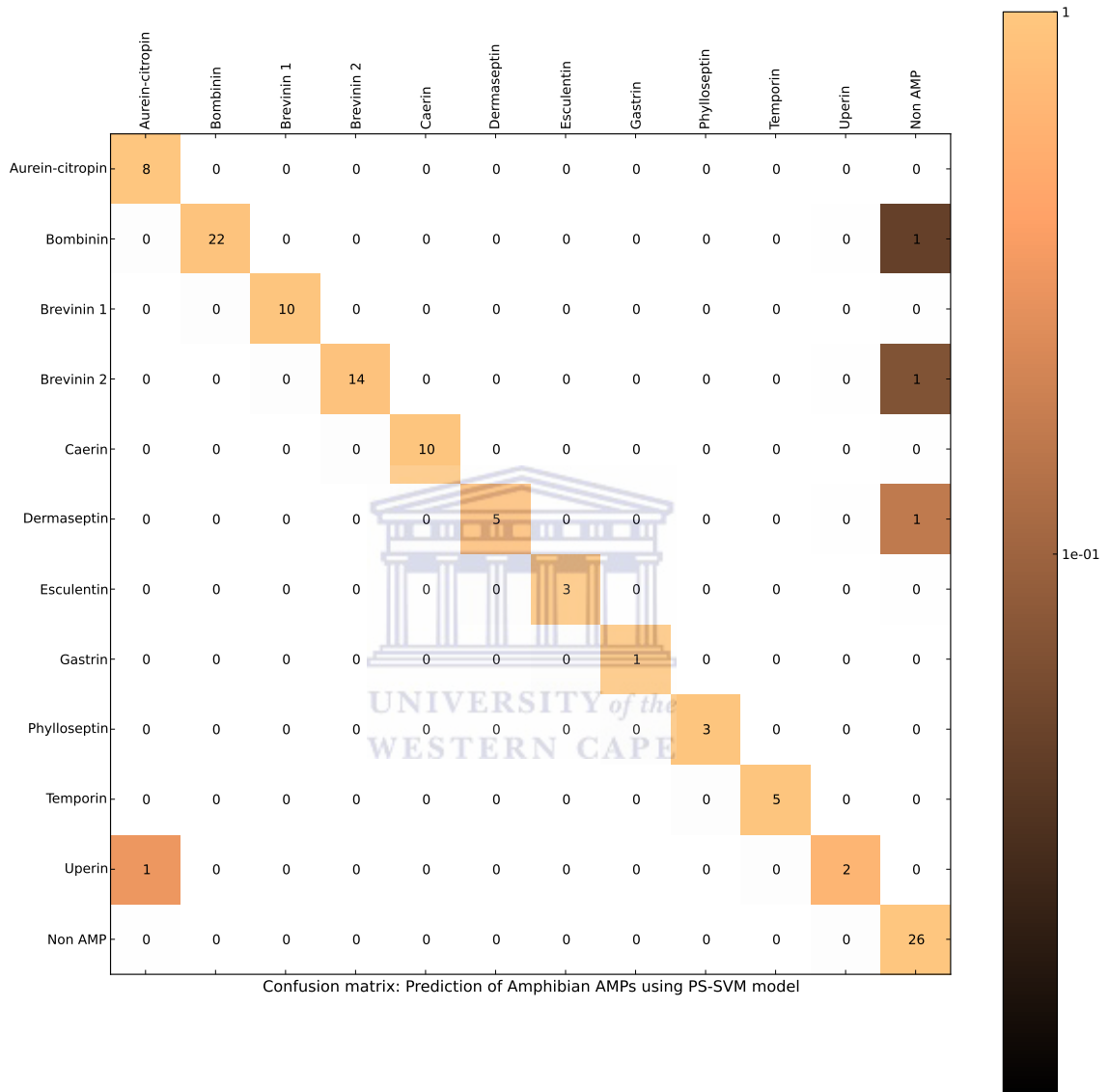


Figure 3.10: Confusion matrix for prediction of amphibian AMPs using PS-SVM model. Each row in a matrix explains how examples in a particular independent set are classified by an algorithm. The matrix is read row-wise. For example, out of 23 examples in bombinin, 22 were classified as bombinin and 1 as non-AMP.

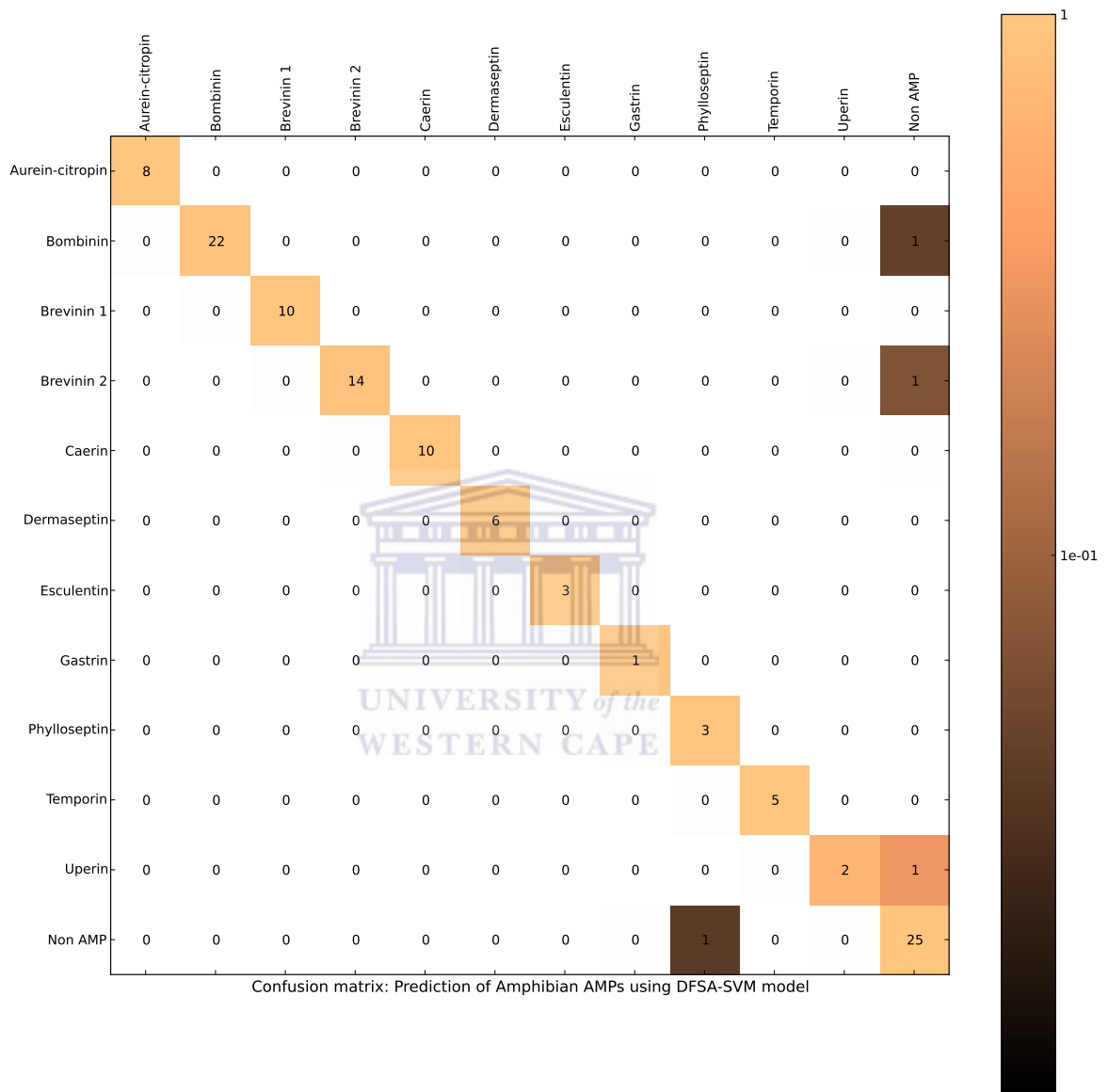


Figure 3.11: Confusion matrix for prediction of amphibian AMPs using DFSA-SVM model. Each row in a matrix explains how examples in a particular independent set are classified by an algorithm. The matrix is read row-wise. For example, out of 23 examples in bombinin, 22 were classified as bombinin and 1 as non-AMP.

Table 3.13: Classification of amphibian antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Aurein-citropin (GS-SVM)	(8.000, 2.000)	8 (109)	100.0	100.0	100.0	1.0
Aurein-citropin (PS-SVM)	(6.905, 0.669)	8 (109)	100.0	99.0	99.1	0.9
Aurein-citropin (DFSA-SVM)	(6.926, 1.442)	8 (109)	100.0	100.0	100.0	1.0
Bombinin (GS-SVM)	(8.000, 0.250)	23 (95)	95.7	98.9	98.2	0.9
Bombinin (PS-SVM)	(4.732, 0.300)	23 (95)	95.7	98.9	98.2	0.9
Bombinin (DFSA-SVM)	(7.992, 0.304)	23 (95)	95.7	98.9	98.2	0.9
Brevinin 1 (GS-SVM)	(8.000, 1.000)	10 (108)	100.0	100.0	100.0	1.0
Brevinin 1 (PS-SVM)	(6.905, 0.669)	10 (108)	100.0	100.0	100.0	1.0
Brevinin 1 (DFSA-SVM)	(7.992, 0.304)	10 (108)	100.0	100.0	100.0	1.0
Brevinin 2 (GS-SVM)	(8.000, 1.000)	15 (103)	93.3	99.0	98.2	0.9
Brevinin 2 (PS-SVM)	(6.905, 0.669)	15 (103)	93.3	100.0	99.1	1.0
Brevinin 2 (DFSA-SVM)	(2.667, 0.881)	15 (103)	93.3	99.0	98.2	0.9
Caerin (GS-SVM)	(8.000, 4.000)	10 (108)	100.0	100.0	100.0	1.0
Caerin (PS-SVM)	(6.905, 0.669)	10 (108)	100.0	100.0	100.0	1.0
Caerin (DFSA-SVM)	(1.722, 2.796)	10 (108)	100.0	100.0	100.0	1.0
Dermaseptin (GS-SVM)	(8.000, 2.000)	6 (111)	83.3	100.0	99.1	0.9
Dermaseptin (PS-SVM)	(7.992, 2.029)	6 (111)	83.3	100.0	99.1	0.9
Dermaseptin (DFSA-SVM)	(7.992, 1.310)	6 (111)	100.0	100.0	100.0	1.0
Esculentin (GS-SVM)	(8.000, 4.000)	3 (115)	100.0	100.0	100.0	1.0
Esculentin (PS-SVM)	(7.992, 3.649)	3 (115)	100.0	100.0	100.0	1.0
Esculentin (DFSA-SVM)	(3.170, 1.642)	3 (115)	100.0	100.0	100.0	1.0
Gastrin (GS-SVM)	(8.000, 8.000)	1 (117)	100.0	100.0	100.0	1.0
Gastrin (PS-SVM)	(1.741, 0.494)	1 (117)	100.0	100.0	100.0	1.0
Gastrin (DFSA-SVM)	(5.501, 0.304)	1 (117)	100.0	100.0	100.0	1.0
Phylloseptin (GS-SVM)	(8.000, 1.000)	3 (115)	100.0	100.0	100.0	1.0
Phylloseptin (PS-SVM)	(1.741, 0.494)	3 (115)	100.0	100.0	100.0	1.0
Phylloseptin (DFSA-SVM)	(7.992, 0.304)	3 (115)	100.0	99.1	99.1	0.9
Temporin (GS-SVM)	(8.000, 1.000)	5 (112)	100.0	100.0	100.0	1.0
Temporin (PS-SVM)	(1.741, 0.494)	5 (112)	100.0	100.0	100.0	1.0
Temporin (DFSA-SVM)	(5.501, 0.304)	5 (112)	100.0	100.0	100.0	1.0
Uperin (GS-SVM)	(1.000, 2.000)	3 (115)	100.0	100.0	100.0	1.0
Uperin (PS-SVM)	(1.741, 0.494)	3 (115)	66.7	100.0	99.1	0.8
Uperin (DFSA-SVM)	(3.829, 3.100)	3 (115)	66.7	100.0	99.1	0.8

Table 3.14: Classification of arachnida antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Cupiennin (GS-SVM)	(8.000, 2.000)	1 (19)	100.0	100.0	100.0	1.0
Cupiennin (PS-SVM)	(1.741, 0.494)	1 (19)	100.0	100.0	100.0	1.0
Cupiennin (DFSA-SVM)	(7.992, 0.304)	1 (19)	100.0	100.0	100.0	1.0
Cytoinsectoxin (GS-SVM)	(8.000, 8.000)	2 (18)	100.0	100.0	100.0	1.0
Cytoinsectoxin (PS-SVM)	(1.741, 0.494)	2 (18)	100.0	100.0	100.0	1.0
Cytoinsectoxin (DFSA-SVM)	(5.501, 0.304)	2 (18)	100.0	100.0	100.0	1.0
Latarcin (GS-SVM)	(8.000, 0.500)	1 (18)	100.0	100.0	100.0	1.0
Latarcin (PS-SVM)	(4.732, 0.494)	1 (18)	100.0	100.0	100.0	1.0
Latarcin (DFSA-SVM)	(7.992, 0.304)	1 (18)	100.0	100.0	100.0	1.0
Oxyopinin (GS-SVM)	(8.000, 4.000)	1 (19)	100.0	100.0	100.0	1.0
Oxyopinin (PS-SVM)	(1.741, 0.494)	1 (19)	100.0	100.0	100.0	1.0
Oxyopinin (DFSA-SVM)	(5.501, 0.304)	1 (19)	100.0	100.0	100.0	1.0
Scorpion NDBP5 (GS-SVM)	(8.000, 1.000)	1 (19)	0.0	100.0	94.7	0.0
Scorpion NDBP5 (PS-SVM)	(1.741, 0.494)	1 (19)	0.0	100.0	94.7	0.0
Scorpion NDBP5 (DFSA-SVM)	(5.501, 0.304)	1 (19)	100.0	100.0	100.0	1.0

Table 3.15: Classification of bacteria antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Bacteriocin (GS-SVM)	(8.000, 1.000)	6 (18)	66.7	100.0	91.7	0.8
Bacteriocin (PS-SVM)	(1.741, 0.494)	6 (18)	83.3	100.0	95.8	0.9
Bacteriocin (DFSA-SVM)	(0.969, 0.530)	6 (18)	83.3	100.0	95.8	0.9
Lantibiotic (GS-SVM)	(8.000, 0.500)	5 (19)	80.0	100.0	95.8	0.9
Lantibiotic (PS-SVM)	(1.741, 0.494)	5 (19)	80.0	100.0	95.8	0.9
Lantibiotic (DFSA-SVM)	(7.992, 0.304)	5 (19)	80.0	100.0	95.8	0.9
Thiocillin (GS-SVM)	(8.000, 0.500)	2 (22)	100.0	100.0	100.0	1.0
Thiocillin (PS-SVM)	(4.732, 0.494)	2 (22)	100.0	100.0	100.0	1.0
Thiocillin (DFSA-SVM)	(7.992, 0.059)	2 (22)	100.0	100.0	100.0	1.0

Table 3.16: Classification of insecta antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
AMP insect (GS-SVM)	(8.000, 1.000)	1 (74)	100.0	100.0	100.0	1.0
AMP insect (PS-SVM)	(6.905, 0.669)	1 (74)	100.0	100.0	100.0	1.0
AMP insect (DFSA-SVM)	(7.992, 0.910)	1 (74)	100.0	100.0	100.0	1.0
Apidaecin (GS-SVM)	(8.000, 8.000)	4 (70)	100.0	100.0	100.0	1.0
Apidaecin (PS-SVM)	(1.741, 0.494)	4 (70)	100.0	100.0	100.0	1.0
Apidaecin (DFSA-SVM)	(0.459, 6.619)	4 (70)	100.0	100.0	100.0	1.0
Attacin (GS-SVM)	(8.000, 4.000)	1 (74)	0.0	100.0	98.6	0.0
Attacin (PS-SVM)	(6.905, 0.669)	1 (74)	0.0	100.0	98.6	0.0
Attacin (DFSA-SVM)	(3.170, 1.642)	1 (74)	0.0	100.0	98.6	0.0
Cecropin (GS-SVM)	(8.000, 2.000)	7 (68)	100.0	100.0	100.0	1.0
Cecropin (PS-SVM)	(1.741, 0.494)	7 (68)	85.7	100.0	98.6	0.9
Cecropin (DFSA-SVM)	(1.563, 0.587)	7 (68)	85.7	100.0	98.6	0.9
Invertebrate defensin (GS-SVM)	(8.000, 0.250)	8 (67)	100.0	100.0	100.0	1.0
Invertebrate defensin (PS-SVM)	(1.741, 0.182)	8 (67)	100.0	98.4	98.6	0.9
Invertebrate defensin (DFSA-SVM)	(4.960, 0.083)	8 (67)	100.0	98.4	98.6	0.9
Crabolin (GS-SVM)	(8.000, 2.000)	1 (74)	100.0	100.0	100.0	1.0
Crabolin (PS-SVM)	(1.741, 0.494)	1 (74)	0.0	100.0	98.6	0.0
Crabolin (DFSA-SVM)	(7.992, 0.304)	1 (74)	100.0	100.0	100.0	1.0
Protonectin (GS-SVM)	(8.000, 2.000)	1 (74)	0.0	100.0	98.6	0.0
Protonectin (PS-SVM)	(6.905, 0.669)	1 (74)	100.0	100.0	100.0	1.0
Protonectin (DFSA-SVM)	(7.992, 1.983)	1 (74)	0.0	100.0	98.6	0.0
Mastoparan (GS-SVM)	(8.000, 0.500)	2 (73)	50.0	100.0	98.6	0.7
Mastoparan (PS-SVM)	(6.905, 0.669)	2 (73)	50.0	100.0	98.6	0.7
Mastoparan (DFSA-SVM)	(7.992, 0.304)	2 (73)	50.0	100.0	98.6	0.7
Ponericin 1 (GS-SVM)	(8.000, 0.500)	1 (74)	100.0	100.0	100.0	1.0
Ponericin 1 (PS-SVM)	(6.905, 0.669)	1 (74)	100.0	100.0	100.0	1.0
Ponericin 1 (DFSA-SVM)	(2.734, 0.772)	1 (74)	100.0	100.0	100.0	1.0
Ponericin 2 (GS-SVM)	(8.000, 2.000)	2 (73)	100.0	100.0	100.0	1.0
Ponericin 2 (PS-SVM)	(6.905, 0.669)	2 (73)	100.0	100.0	100.0	1.0
Ponericin 2 (DFSA-SVM)	(1.944, 2.216)	2 (73)	100.0	100.0	100.0	1.0

Table 3.17: Classification of mammalia antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Alpha defensin (GS-SVM)	(4.000, 1.000)	10 (180)	80.0	100.0	98.9	0.9
Alpha defensin (PS-SVM)	(1.741, 0.814)	10 (180)	80.0	100.0	98.9	0.9
Alpha defensin (DFSA-SVM)	(0.718, 0.593)	10 (180)	80.0	100.0	98.9	0.9
Beta defensin (GS-SVM)	(8.000, 0.500)	6 (184)	83.3	100.0	99.5	0.9
Beta defensin (PS-SVM)	(6.905, 0.669)	6 (184)	83.3	100.0	99.5	0.9
Beta defensin (DFSA-SVM)	(6.125, 0.711)	6 (184)	83.3	100.0	99.5	0.9
Cathelicidin (GS-SVM)	(8.000, 1.000)	2 (189)	50.0	100.0	99.5	0.7
Cathelicidin (PS-SVM)	(6.905, 0.669)	2 (189)	50.0	100.0	99.5	0.7
Cathelicidin (DFSA-SVM)	(7.992, 0.632)	2 (189)	50.0	100.0	99.5	0.7
Glycosyl hydrolase 22 (GS-SVM)	(8.000, 8.000)	2 (188)	100.0	100.0	100.0	1.0
Glycosyl hydrolase 22 (PS-SVM)	(0.771, 5.516)	2 (188)	100.0	100.0	100.0	1.0
Glycosyl hydrolase 22 (DFSA-SVM)	(7.992, 0.304)	2 (188)	100.0	100.0	100.0	1.0
Histone H2B (GS-SVM)	(8.000, 8.000)	2 (188)	50.0	100.0	99.5	0.7
Histone H2B (PS-SVM)	(0.771, 5.516)	2 (188)	50.0	100.0	99.5	0.7
Histone H2B (DFSA-SVM)	(1.722, 2.796)	2 (188)	50.0	100.0	99.5	0.7

Table 3.18: Classification of merostomata antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Tachyplesin (GS-SVM)	(8.000, 8.000)	1 (8)	100.0	100.0	100.0	1.0
Tachyplesin (PS-SVM)	(1.741, 0.494)	1 (8)	100.0	100.0	100.0	1.0
Tachyplesin (DFSA-SVM)	(7.992, 0.077)	1 (8)	100.0	100.0	100.0	1.0

In Table 3.19, the take home message is that the three hybrid methods performed poorly in identifying AMPs in thaumatin family in terms of accuracy.

Table 3.19: Classification of plant antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
DEFL (GS-SVM)	(8.000, 2.000)	9 (191)	88.9	100.0	99.5	0.9
DEFL (PS-SVM)	(1.741, 0.494)	9 (191)	88.9	100.0	99.5	0.9
DEFL (DFSA-SVM)	(1.131, 0.981)	9 (191)	88.9	100.0	99.5	0.9
Thaumatoin (GS-SVM)	(8.000, 0.125)	2 (198)	0.0	100.0	99.0	0.0
Thaumatoin (PS-SVM)	(1.741, 0.494)	2 (198)	0.0	100.0	99.0	0.0
Thaumatoin (DFSA-SVM)	(4.361, 0.295)	2 (198)	0.0	100.0	99.0	0.0

Classification of crustacea AMPs into penaeidin, is shown in Table 3.20. Both GS-SVM and DFSA-SVM failed to predict the single positive penaeidin sequence, though they correctly classified all the non-AMPs.

Table 3.20: Classification of crustacea antimicrobial peptides into AMP families

Algorithm	Model	# of peptides	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Penaedin (GS-SVM)	(8.000, 8.000)	1 (17)	0.0	100.0	94.4	0.0
Penaedin (PS-SVM)	(1.741, 0.494)	1 (17)	100.0	100.0	100.0	1.0
Penaedin (DFSA-SVM)	(1.722, 2.796)	1 (17)	0.0	100.0	94.4	0.0

3.4.4 Performance comparison of GS-SVM, PS-SVM and DFSA-SVM

The performance comparison of GS-SVM, PS-SVM and DFSA-SVM on a blindset have so far been presented above for the two different experiments, namely experiment one (generalized) and experiment two (specialised) AMP models. In this subsection, comparison of these methods obtained for the above two scenario are presented. Note that the same parameter settings are used in two experiments in order to have a fair comparison. The respective performance measures (sensitivity, specificity, accuracy and MCC) of all the tables from Table 3.3 to Table 3.11 are added together and the average values are presented in Table 3.21. Similarly, Table 3.22, is the average of the total sum of the respective performance measures of Table 3.12 to Table 3.20.

The accuracy is used here as a measure of performance to compare the three hybrid methods. The average comparison of GS-SVM, PS-SVM and DFSA-SVM using generalized models is given in Table 3.21. Clearly DFSA-SVM was the overall best hybrid method with an accuracy of 97.95% in discriminating the positive set from the negative set, followed by GS-SVM and PS-SVM in that order.

The average comparison of GS-SVM, PS-SVM and DFSA-SVM using specialised model (AMP families) is presented in Table 3.22. GS-SVM is the best performer in terms of specificity, while PS-SVM is the best in terms of sensitivity and accuracy. In both experiments, PS-SVM was the best method with an accuracy of 99.25%, in prediction of AMPs into their respective families.

Classification of AMPs using generalized and specialized models shows that specialized model are more specific and accurate than generalized models. However, the generalized model has better sensitivity than specialized model. There is an improvement in specificity in specialised model as compared to generalized model. This is because the multi-class scheme is more class based and hence the model is tailored to a specific class rather than a general class.

The results achieved in all the classification be it generalized or specialized suggests that although AMPs are diverse, the pattern becomes apparent for an AMP family in a particular taxa. This is clear from the high accuracies achieved using specialised models.

Table 3.21: Average performance comparison of the three hybrid methods (generalized AMP model)

Algorithm	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
GS-SVM	84.45	96.52	97.54	0.82
PS-SVM	83.09	97.45	97.43	0.82
DFSA-SVM	84.45	97.45	97.95	0.83

Table 3.22: Average performance comparison of the three hybrid methods (specialised AMP models)

Algorithm	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
GS-SVM	80.53	99.95	99.10	0.83
PS-SVM	82.26	99.91	99.25	0.85
DFSA-SVM	81.42	99.88	99.20	0.84

3.5 Discussion

Pathogens have ingeniously grown resistant to conventional antibiotics and this has led to pharmaceutical industries to seek another alternative. Antimicrobial peptides are considered to be an alternative drug as compared to conventional antibiotics because:

- AMPs have a broad range of activity as they work against all microbes and parasites.
- AMPs also have high specificity as they can recognize and destroy only microbes without disrupting any other cells in the body.
- AMP shows either no or very low drug resistance.

For these reasons, this has generated a lot of interest in pharmaceutical industries to create these peptides synthetically and also create hybrids of these peptides to increase efficacy of their functional range (Ferre et al., 2006). Pexiganan is used as topical antibiotic for the treatment of infected diabetic foot ulcers (Dutton et al., 2002). Dermaseptin peptides were shown to be active toward human erythrocytes infected by the malaria parasite *Plasmodium falciparum* (Ghosh et al., 1997). Indolicidin-analogue is used for treatment of acne vulgaris. Plectasin have microbicidal activity against antibiotic-resistant bacteria (Guani-Guerra et al., 2010).

The characterization of an antimicrobial peptide can be assayed *in vivo* or predicted *in-silico*, i.e., classified into experimental approaches and computational approaches. Experimental approaches for determining antimicrobial peptide activity include but are not restricted to microscopy, fluorescent dyes, ion channel formation, circular dichroism and oriented circular dichroism, solid-state NMR spectroscopy and neutron diffraction. (Brogden, 2005). Even though many new AMPs with improved activity have been reported, rarely has any method been used to scan the potential vast amount of genomic and proteomic data, to discover unknown AMPs. As of September 2009, data available to the public indicates that there are 890 complete genomes and 5643 ongoing genome projects (Kyrpides, 1999). Due to rapid release of new data from genome sequencing projects, the majority of protein sequences in public databases have not been experimentally characterized. Hence the need to develop computational approach to identify potential AMPs.

Several computational approaches have been implemented in classifying or rather characterizing novel antimicrobial peptide from protein sequences. These approaches include similarity search based tech-

niques such as BLAST (Altschul et al., 1990) and PSI-BLAST (Altschul et al., 1997), profile search methods (profile hidden Markov model) and multivariate classification methods. Both similarity and profile search methods fail to predict a new protein when a query protein does not have significant similarity with the database proteins. Recently, random forest, SVM and discriminant analysis has been applied in predicting antimicrobial peptides (Thomas et al., 2010). Artificial neural networks (ANN), quantitative matrices (QM) and support vector machines (SVM) has been designed to predict antibacterial peptides (Lata et al., 2010, 2007). Quadratic discriminant analysis was used in classification of antimicrobial peptides using diversity measure (Chen and Luo, 2009). Fourier transform based method with property based coding strategy used to scan the peptide space for discovering new potential antimicrobial peptides (Nagarajan et al., 2006). Decision tree have been developed in for classification of antimicrobial peptides (Lee et al., 2004).

These methods for predicting AMPs have some bottlenecks. For example, Thomas et al. (2010) generated a generalized model to predict AMPs in their online CAMP database. One limitation of this model is that it is not specific. They have predicted AMPs using a generalized model. On the other hand, the specific models used in this study is based on AMP families which is more robust and nevertheless, it not only predict AMPs with high accuracy, but also classifies them into specific subclasses such as cecropin, defensin, α -defensin etc. ANTIBP2 is another tool for predicting AMPs based on families. However, this method suffers from one drawback, in that the training set of sequences from APD database (Wang and Wang, 2004), which consists of predicted and experimentally validated AMPs. Fourier transform and decision based methods predict AMPs of a particular group, that is, antibacterial and anticancer respectively. GS-SVM, PS-SVM and DFSA-SVM were compared with ANTIBP2 (Lata et al., 2010). ANTIBP2 is an online tool for predicting AMPs based on families. The comparison were made based on accuracy and Mathews correlation coefficient (MCC) parameters. The results shows that GS-SVM, PS-SVM and DFSA-SVM are superior (in terms of accuracy) than ANTIBP2 in predicting

- frog AMPs namely bombinin, brevinin, caerin and dermaseptin,
- insect AMPs namely apidaecin and attacin, except invertebrate defensin and cecropin, and
- mammal AMPs namely alpha-defensin, beta-defensin and cathelicidin.

The method implemented here utilises only primary protein sequences to build a matrix representation of AMPs sequences based on amino acid composition and physicochemical properties. These methods for generating features are based on an averaging scheme influenced by sequence length of the protein.

One disadvantage of this approach is that it does not consider coupling effect among the neighbouring residues. Nevertheless, it considers the whole protein sequence as a whole rather than placing emphasis on certain motifs or domains of the sequence. Although the approach is easy to implement, it might obscure certain domain specific characteristics, and hence the mean values might reduce the robustness of the model. Future work entails implementation of more sophisticated methods for generating features. These methods are based on string kernels for protein sequences and they include but are not restricted to SVM-Fisher (Jaakkola et al., 2000), SVM-pairwise (Liao and Noble, 2002), eMotif kernel (Ben-Hur and Brutlag, 2003), mismatch kernel (Leslie et al., 2004), cluster kernel (Weston et al., 2005), spectrum kernel (Leslie et al., 2002) and profile-based string kernels (Kuang et al., 2005). The features used by the spectrum kernel are the set of all possible subsequences of amino acid of a fixed length l . If two protein sequences contain many of the same l subsequences, then their inner product $K(\vec{x}_1, \vec{x}_2) = \vec{x}_1^T \cdot \vec{x}_2$ under the k -spectrum kernel will be large (Leslie et al., 2002). Another example of a string kernel is the mismatch kernel, which counts slightly mismatched strings of sequences as being similar. Mismatch kernel is based on the mismatch neighbourhood $N_{(k,m)}(S)$ of a sequence S and is the set of all k -mers within m mismatches from S (Leslie et al., 2004). Profile kernels, on the other hand, use probabilistic profiles such as those produced by the PSI-BLAST algorithm to define position-dependent mutations neighbourhoods along protein sequences for exact matching of k -length subsequences (k -mers) in the data (Kuang et al., 2005). The implementation of these kernel is beyond the scope of the thesis.

The machine learning technique implemented for the prediction of AMPs is based on an SVM. SVMs have been widely used in many practical applications. Some of these applications include but are not restricted to drug discovery (Zernov et al., 2003), automatic naming of proteins due increasing demand in data mining (Mika and Rost, 2004), early detection and prognosis of cancer (Kapetanovic et al., 2004), prediction of protein-protein interactions (Soong et al., 2008), transcription initiation site prediction (Yang, 2004), prediction of HIV coreceptor usage (Boisvert et al., 2008; Garrido et al., 2008; Prospero et al., 2009; Sander et al., 2007; Skrabal et al., 2007), identification of biomarker for cancer diagnosis (Abeel et al., 2010), identification of diabetic retinopathy stages (Acharya et al., 2011), prediction of microRNA coding regions in genome scale sequences (Wu et al., 2011), and classification of lip color in relation to personal health (Zheng et al., 2011).

The performance of the SVM model applied in the present study achieved an average prediction accuracies of 99.25%, 99.10% and 99.20% in classifying peptides of various taxa into AMPs families using PS-SVM, GS-SVM and DFSA-SVM respectively. The performance of the model that achieved the highest

accuracy is chosen and reported. Different modelling criteria require that the recall of the model should be 100%, so as to not lose any positives that are within the set. On the other hand, other criteria require a precision of 100%, so as to be absolutely sure about a positive predicted element. Other criteria can be used to evaluate the performance of supervised learning, namely F-measure, MCC and the area under the curve (AUC). The accuracy measure is known to have several defects in that it does not exclude the influence of the class distribution which may enable a completely uninformed classifier to trivially achieve high classification accuracy. A remedy to this is to use a two level measure known as area under the curve:accuracy, in short $AUC : acc$ (Huang and Ling, 2007). Suppose AUC is denoted as f and accuracy as g , then the two level measure is defined as

Definition 1. A two-level measure ψ formed by f and g , denoted by $f : g$, is defined as:

- $\psi(a) > \psi(b) \iff f(a) > f(b) \text{ and } g(a) > g(b)$
- $\psi(a) = \psi(b) \iff f(a) = f(b) \text{ and } g(a) = g(b)$;

that is, if AUC values of the two ranked prediction lists a and b of two classifiers are different, then the new two-level measure $AUC : acc$ agrees with AUC , no matter what value of accuracy is. On the other hand, if AUC values are the same, then the two-level measure agrees with accuracy.

The performance of SVMs depends heavily on the selected hyperparameters c and σ employed to train the model. The approach of selecting these SVM hyperparameters with grid search is not global since no optimal criteria for convergence of solution. To circumvent this problem, a hybridized SVM with two optimization methods, namely pattern search (PS) and derivative-free simulated annealing (DFSA) is implemented to optimize the SVM hyperparameters. Both PS-SVM and DFSA-SVM are robust and rarely get trapped in local minima. PS-SVM is the best SVM hybrid method for selecting SVM hyperparameters because PS is started at several multi-start points in the search space. Nevertheless, PS updates the next stepsize parameter Δ^{k+1} to be equal to $\alpha \times \Delta^k$, where α is the expansion factor. In the implementation of PS, the value of α is set to 2. The expansion factor $\alpha = 2$ is compared with $\alpha = 1$. The expansion factor $\alpha = 2$ is much superior to $\alpha = 1$ in terms of accuracy. This is due to the fact that $\alpha = 2$ is more opportunistic than $\alpha = 1$ when a better point is obtained. In other words it speeds up the convergence of PS method to the global point.

The optimization of SVM hyperparameters is a black-box simulation and it is interesting to note that there are several global minima solutions of a given modelling process. This is known as a multi-modal

problem. At each iteration of GS-SVM, PS-SVM, DFSA-SVM, the best solution is chosen, once a solution with either equal or greater accuracy value than the current best solution is found. The problem with this approach is that it selects the values of c and σ blindly. This can lead to overfitting, hence the derived model will not predict with high accuracy when tested on the blindset. If the hyperparameter value c is too high, then classification rate is very high in the training stage but very low in the testing stage. On the other hand, if c is too small, then the classification accuracy rate is unsatisfactory, hence not a useful model. Hyperparameter σ has more effect than c , in that it influences the separation outcome in the feature space. A high value for σ leads to overfitting while a low value results in underfitting. Therefore, one strategy to deal with these extremities would be to select c and σ conservatively (Lin et al., 2008). If different solutions have equal classification accuracy value, then the one with the smaller c value is selected. If the c values are identical, then the one with the smaller σ value is chosen.

In order to additionally evaluate the performance of the model, it was tested on an independent data set. The accuracy rate was very high in most of AMPs prediction in various taxa. However, the worst performed prediction of AMPs was in thaumatin, attacin and cathelicidin. In thaumatin, there is a wide variation in sequence length, for instance, in thaumatin AMP family. Since, our featurization method is influenced by the length of the sequence, this obscures the pattern and hence affecting the model. The other reason is that the thaumatin sequences are highly diverse, that is, less conservation depicted in their sequences.

The conservation of antibacterial peptides in amino acid sequence has been well documented across evolutionary distant taxa. However, there is a wide genetic variation within taxa. Lazzaro et al. (2001) researched on the quantity and origin of polymorphisms in *Drosophila melanogaster* Attacin genes. Attacins represent one of the most taxonomically widespread classes of antibacterial peptides. Mature attacin peptides are typically ≈ 190 amino acids in length and adopt a random coil structure in solution. This loose, flexible structure is devoid of disulfide bonds and does not take a rigid conformational shape. This lack of strict structural constraint may allow relatively free amino acid substitution, explaining the lower level of amino acids identity between attacin homologs in distant taxa. There is however, conservation of general structure and functional activity. This may explain the underlying reasons why the sensitivity of the prediction of attacin AMPs was zero. Similarly the prediction performance for cathelicidins was too low, i.e., a sensitivity of 50%. This is because cathelicidins are composed of a large and particularly diverse family of AMPs that derived from prepropeptides with a particularly well conserved N-terminal prepropeptide segment (the cathelin domain) of approximately 100 amino acid residues (Bulet et al., 1993, 2004).

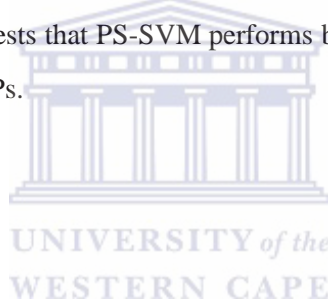
The variation and diversity of the AMP sequence within the same family and species (Maxwell et al., 2003) makes it difficult to identify or predict new AMPs, thus another methods has been proposed for some specific class of genes (Wasserman and Fickett, 1998) based on the model of the gene's promoter region. This approach seems reasonable to use for the purpose of AMP gene discovery as the literature suggests that the promoter regions of the highly diverse AMPs are fairly conserved (Brahmachary et al., 2006; Ganz, 2003). This method can be complemented with homology based gene identification methods to increase the possibilities of mining novel AMPs from the whole genome.

The reliability of a trained model from any classifier mainly depends on the four factors, namely the number of clean data, the selection of classifier hyperparameters, feature representation and feature selection. In this study, the best features to be used in the featurization of our examples was not conducted, instead the 27 features were all used in the training. This may suggest the reason for not getting a reliable model. Feature selection is very important and therefore having so many features that contain redundancy or high level of noise may decrease the accuracy of the solution. Removal of such features can improve the search speed and accuracy rate. Therefore, the objective of feature selection is to come up with useful features that correlates between the positive and negative sets. There are two methods for feature selection, namely filter approach and wrapper approach. In filter method, the feature selection and classifier design are separated in that a subset of features are first selected and then the classifiers are trained based on the selected features. Examples of filter methods include but are not limited to *t*-statistics (Pan, 2002), FDR (Pavlidis et al.) and signed-FDR (Golub et al., 1999). Filter methods are very fast and simple to implement but come with several costs. These methods require users to set a cut-off point in the ranking scores under which features are deemed to be irrelevant for classification. However, the optimal number of features, that is, the cut-off is usually unknown. Another problem is that the ranking criteria do not take the combined effect of features into account (Lin et al., 2008). To circumvent these limitations, is to include the feature selection in the training process so that the performance of classifiers can guide the selection progress. This approach is called the wrapper method. In this approach, the feature selection algorithm conducts a search for a good subset of features using the classifier itself as part of the evaluating the objective function, i.e., the accuracy. The advantage of wrapper methods is that it takes into account the effect of selected feature subset on the performance of the classification algorithm during the search. Examples of wrapper methods include implementing population-based global optimization methods such as genetic algorithm, particles swarms optimization, tabu search and ant colony optimization methods.

In most of the prediction, the specificity of 100% was achieved because the model was created on the entire protein universe as the negative set. The specificity value would be lower if you use a larger dataset of non-AMPs.

3.6 Summary

Three SVM hybrid methods namely, GS-SVM, PS-SVM and DFSA-SVM for selecting optimal hyperparameters of support vector machine have been presented. The model generated for these three hybrid methods were tested on a blind data set from various taxa, namely insecta, amphibian, merostomata, mammalia, plantae, arachnida, actinopterygii and bacteria. Prediction of AMPs using their respective families are more accurate than training based on generalized AMPs. One advantage of PS-SVM and DFSA-SVM hybrid is that they optimally select the SVM hyperparameters which is an important aspect in classification process. Numerical results suggests that PS-SVM performs better than GS-SVM and DFSA-SVM in discriminating AMPs from non-AMPs.



Chapter 4

Haemotophagous antimicrobial peptide predictor webserver



Abstract

Background: Innate immunity has a primary role in protecting organisms from a diverse spectrum of microorganisms in the invertebrates. In insect vectors, which transmit parasites that cause major human and animal diseases, antimicrobial peptides (AMPs) play an essential role in innate immunity. AMPs in insects are grouped into eight major families, namely invertebrate defensin, attacin, cecropin, AMP insect, crabolin, protonectin, mastoparan and ponerin.

Methods: In this study, the haemotophagous antimicrobial peptide predictor (HAPP) webserver was designed using a support vector machine coupled with optimization methods (grid search, pattern search and derivative-free simulated annealing) to predict classes of AMPs in haemotophagous insects. For each SVM raw score, a complementary statistical confidence measure called posterior error probability is computed using QVALITY program.

Results: HAPP webserver predicts with an accuracy of 95%.

Conclusion: The HAPP webserver can be used to predict AMPs into their respective families in haemotophagous insects and can be a useful resource to characterize peptides in ongoing genomes in insects. The HAPP webserver can be accessed at <http://apps.sanbi.ac.za/Happ/>.

4.1 Introduction

Insects, the most abundant metazoans on earth, have a well-developed innate immune system that respond to infection. The innate immunity in insects is divided according to the type of immune response, namely, humoral and cellular response. The humoral response is based on the products of characterized immune genes induced by microbial infection and encode antimicrobial peptides. On the other hand, cellular response are performed by hemocytes and include phagocytosis and encapsulation (Hoffmann and Hetru, 1992).

A number of AMPs in insects have been isolated and characterized in insects. Examples include but not restricted to cecropin, attacin, insect defensin and dipterin in *Drosophila* (Akuffo et al., 1998; Samakovlis et al., 1990; Valanne et al., 2011; Zhao et al., 2011). Cecropins are 31 to 39 residue peptide lacking cysteine and are highly active against gram-positive or gram-negative bacteria. Attacins are typically \approx 190 amino acids in length and are characterized by high content of glycine residues and show activity against gram-negative bacteria. Insect defensin are cationic peptides composed of 32 to 51 amino acid residues and all contain a characteristic motif of six cysteines bonded in three intramolecular disulfide regions. They attack mostly gram-positive bacteria. Dipterin are antibacterial peptides of about 82 amino acids and shows activity against gram-negative bacteria. Nonetheless, there are a number of uncharacterized peptides in many insect vectors. For example in VectorBase (Lawson et al., 2009), there are several ongoing genomes for haemotophagous insects namely *Glossina morsitans morsitans*, *Rhodnius prolixus*, *Anopheles* species cluster (*An. gambiae*, *An. fenestus*, *An. stephensi*, *An. arabiensis*, *An. quadriannulatus*, *An. merus*), *Culex quinquefasciatus*, *Ixodes scapularis* and *Aedes aegypti*. Therefore, there is need for a computational approach to characterize AMPs in these vectors.

These disease vectors are rich in AMPs, which are induced upon parasitic infections and involved in controlling parasite development (Boulanger et al., 2002). They transmit parasites that cause major diseases such as malaria, sleeping sickness, leishmaniasis and filariasis in human and nagana in animals. There is need to identify the AMPs in haemotophagous insects and the reason for this can be explained in three folds: Firstly, experimental methods used in characterizing AMPs are costly, time consuming and resource intensive. Thus, there is need to develop computational tool for predicting AMPs, in order to inform experimental approaches. Secondly, identification of AMPs in insects can serve as a natural template for designing novel antibiotics useful in combating or controlling diseases such as malaria and sleeping sickness. Thirdly, the antimicrobial molecules are highly attractive for use in transgenic technology in

insect vectors.

In this study, a machine learning approach using the support vector machines is utilised in the prediction of AMPs in insects. Optimization methods namely grid search, pattern search and derivative free simulated annealing methods are utilised to select SVM hyperparameter. However, the prediction produced by the SVM classification method raises a related question, that is, how confident can we be that the classifier has actually identified the peptide as AMP or not? To answer this question, a posterior error probability (PEP) of a given peptide SVM prediction score is computed using QVALITY program (Käll et al., 2009). PEP is defined as the probability that a single peptide score called significant is actually incorrect.

This chapter is organized as follows. Section 4.2 briefly presents the methodology employed. Section 4.3 presents the estimation of statistical confidence measures. Results and discussions are presented in section 4.4. Section 4.5 describes the webserver and a summary is made in section 4.6.

4.2 Methods



The methodology used in creating the webserver is based on what has been presented in Chapter 3. The three hybrid algorithms namely, GS-SVM, PS-SVM and DFSA-SVM and the materials discussed in section 3.2 and 3.3 respectively of Chapter 3, are implemented here. In addition to these, a procedure to measure the statistical confidence (posterior error probability) of SVM prediction is incorporated in the pipeline. This is presented in the next section.

4.3 Estimation of statistical confidence measures

A classifier such as an SVM is useful if it delivers scores that have well-defined semantics. In this work, an empirical post-processing procedure for converting the unitless SVM discriminant score into two complementary statistical confidence measures is computed. Both measures rely on the notion of a *null model*, which represents the noise of the process being modeled. This procedure, randomly generated strings of amino acids are used as an empirical null model, and will be described later. There are two measures namely, false discovery rate (FDR) and posterior error probability (PEP).

The first measure is based on the estimated *false discovery rate* (FDR) (Käll et al., 2008a,c, 2009;

Noble, 2009). The FDR is defined as the percentage of scores above a specified threshold that are drawn according to the null hypothesis. In practice, raw FDR estimates are problematic because the FDR is not monotonically related to the underlying score. A sequence is monotonic if it is consistently increasing or never decreasing or consistently decreasing and never increasing in value. Therefore, instead a q -value is reported, which is defined as the minimal FDR at which a given score is deemed significant (Käll et al., 2008a,c). The q -value is thus an analog of the p -value that incorporates multiple testing correction.

The second measure is the *posterior error probability* (PEP) (Käll et al., 2008a,c), defined as the probability that the score is drawn according to the null hypothesis. In statistics literature, the PEP is sometimes referred to as the *local false discovery rate*. The q -value and PEP are complementary confidence measures. The q -value is easier to estimate accurately, nonetheless it only provides information about the set of scores at or above a specified threshold. The PEP is more difficult to estimate accurately but provides information about an individual score. Which score is relevant will depend in general upon what type of follow-up experiments are planned: for batch validation, the q -value is appropriate; for follow-up of individual predictions, the PEP is relevant. For more details on PEP and q -value, see (Käll et al., 2008a,c).

To estimate both measures, an empirical null model (noise) coupled with a standard FDR inference procedures are used. In order to come up with a measure, one needs to compare the observed scores of target (real) sequences with the sequence scores from a null or rather a decoy database. A null database is a warehouse of amino acid sequences that are derived from the original target protein database called FIXME. There are several ways to generate the null database namely by: reversing the target sequences (Moore et al., 2002) shuffling the target sequences (Klammer and MacCoss, 2006) and generating the decoy sequences at random using a Markov model with parameters derived from the target sequences (Colinge et al., 2003). There is no optimal way to generate a null database, however, we have ensured that the sequences in the decoy database are different from the target database. In this thesis, the decoy sequences are generated at random using the Markov model. The detailed procedure for generating the decoy database and estimating the non-parametric estimation of the q -values and PEP is described as follows:

1. **Gather a non-redundant set of insect proteins**

A non-redundant set of insect proteins are used by purging 378 FIXME proteins sequences (138 AMPs, 240 non-AMPs) by using CD-HIT (Li and Godzik, 2006). This is done to prevent any pair of sequences from sharing greater than 40% sequence identity. This procedure yields a total of FIXME sequences, which consists of 106 AMPs and 178 non-AMPs.

2. Train a Markov chain

From these sequences, the parameters of the zero-order and first-order Markov chain are estimated. This procedure yields a total of 420 (20 zero-order and 400 first-order) parameters. A first-order Markov chain is where the transition of one event to the other event is dependent on the one immediately preceding it, unlike a zero-order Markov chain.

3. Generate empirical null sequences

The Markov chain is then used as a generative model and the steps involved to generate the null sequences is described as follows:

- (a) select a protein sequence uniformly at random from the given initial collection of proteins and record the protein's length, l .
- (b) randomly select the first amino acid in the simulated protein according to the zero-order Markov frequencies (parameters).
- (c) randomly select the next amino acid in the simulated protein according to the first-order Markov frequencies, conditioning on the previous amino acid.
- (d) repeat step (c) until the protein is of length, l .

This procedure is repeated until a specified number of simulated proteins have been generated. In this case, 1000 null sequences was generated.

4. Apply the trained classifier to the null sequences

The trained SVM model is applied to each of the null sequences, recording the resulting scores. The resulting score distribution serves as our empirical null model.

5. Apply the trained classifier to the real sequences

The trained SVM model is applied to each target sequence in a collection of proteins of interest (284 sequences), storing the observed SVM scores.

6. Estimate q -value and PEPs

Many tools exist for estimating q -value and PEPs (Strimmer, 2008b). Some of these tools include but not restricted to locfdr tool (Efron, 2004), BUM (Pounds and Morris, 2003) and fdrtool (Strimmer, 2008a). However, these tools have limitations in that it requires the user to furnish either with p -values, z -scores, t -scores, or correlation scores. Therefore, for this purpose, QUALITY software

(Käll et al., 2009) is used, which takes as an input both the observed and the null SVM scores and produces both q -values and PEPs.

An example is given below to expound on step 3 above, that is how to generate the null sequences.

Example: For simplicity case, suppose the four nucleotide bases namely “**A**”, “**C**”, “**G**” and “**T**” are used instead of the 20 amino acids. In this case, 20 parameters are generated, i.e., 4 zero-order frequencies and 16 first-order frequencies. Suppose further that the zero-order frequencies $F^{(0)}$ are

$$F^{(0)} = \begin{matrix} \mathbf{A} \\ \mathbf{G} \\ \mathbf{C} \\ \mathbf{T} \end{matrix} \begin{pmatrix} 0.2 \\ 0.1 \\ 0.6 \\ 0.1 \end{pmatrix}$$

and the first-order frequencies $F^{(1)}$ as follows

$$F^{(1)} = \begin{matrix} & \mathbf{A} & \mathbf{G} & \mathbf{C} & \mathbf{T} \\ \mathbf{A} & \begin{pmatrix} 0.2 & 0.4 & 0.3 & 0.1 \end{pmatrix} \\ \mathbf{G} & \begin{pmatrix} 0.1 & 0.25 & 0.4 & 0.25 \end{pmatrix} \\ \mathbf{C} & \begin{pmatrix} 0.34 & 0.23 & 0.21 & 0.22 \end{pmatrix} \\ \mathbf{T} & \begin{pmatrix} 0.25 & 0.35 & 0.15 & 0.25 \end{pmatrix} \end{matrix}$$

For instance, the first-order frequency of **C** is **CA** : 0.34, **CG** : 0.23, **CC** : 0.21, **CT** : 0.22. In addition, suppose that the sequence “CCGTTTTA” is chosen randomly from the target database. The procedure to generate the null sequences is as follows:

- (a) the length of the protein is $l = 8$.
- (b) randomly select the first amino acid in the simulated protein according to the zero-order Markov frequencies. Here a random number φ is generated and will be assigned **A** if $0 < \varphi \leq 0.2$, **C** if $0.2 < \varphi \leq 0.8$, **G** if $0.8 < \varphi \leq 0.9$ and **T** if $0.9 < \varphi \leq 1.0$, i.e., $x_k, k = 0$

$$x_0 = \begin{cases} \mathbf{A} & \text{if } 0 < \varphi \leq 0.2, \\ \mathbf{C} & \text{if } 0.2 < \varphi \leq 0.8, \\ \mathbf{G} & \text{if } 0.8 < \varphi \leq 0.9, \\ \mathbf{T} & \text{if } 0.9 < \varphi \leq 1.0 \end{cases} \quad (4.1)$$

Suppose the number 0.3 is randomly chosen, so using the equation (4.1), **C** is selected as the first amino acid in the simulation, because 0.3 is in $0.2 < \varphi \leq 0.8$.

- (c) randomly select the next amino acid in the simulated protein according to the first-order Markov frequencies, conditioning on the previous amino acid. Since **C** was chosen, then transition probabilities of **C** using the first order frequencies $F^{(1)}$ is considered, i.e.,

$$\mathbf{CA} : 0.34, \mathbf{CG} : 0.23, \mathbf{CC} : 0.21, \mathbf{CT} : 0.22.$$

Suppose a number $\varphi = 0.9$ is randomly chosen, then the next base to be selected depends on the following:

$$x_{k+1} = \begin{cases} \mathbf{A} & \text{if } 0 < \varphi \leq 0.34, \\ \mathbf{G} & \text{if } 0.34 < \varphi \leq 0.57, \\ \mathbf{C} & \text{if } 0.57 < \varphi \leq 0.78, \\ \mathbf{T} & \text{if } 0.78 < \varphi \leq 1.0 \end{cases} \quad (4.2)$$

i.e., the base **T** is chosen because $\varphi = 0.9$ is within $0.78 < \varphi \leq 1.0$. Therefore, **T** becomes the current base to generate the next sequence in the growing string of sequences.

- (d) repeat step (c) until the protein is of length, l , i.e. $k = l - 1$.

4.4 Results and Discussion

In this section, the prediction results for the three hybrid methods, namely, GS-SVM, PS-SVM and DFSA-SVM discussed in chapter 3 are presented. In Table 4.1, the three hybrid methods performed well in discriminating the insect AMPs into their respective families.

Table 4.1: Average performance of GS-SVM, PS-SVM and DFSA-SVM models in classification of insecta AMPs

Algorithm	Precision	Sensitivity(%)	Specificity(%)	Accuracy(%)	MCC
Average (GS-SVM)	80.00	75.00	100.00	99.58	0.77
Average (PS-SVM)	78.89	73.57	99.84	99.31	0.76
Average (SAPS-SVM)	78.89	73.57	99.84	99.31	0.76

The confusion matrix for GS-SVM model is shown in Figure 4.1. Each row in a matrix explains how AMP family of a particular taxa are classified by the hybrid algorithm, In this Figure, one attacin AMP was classified as non-AMP using the GS-SVM model. Similarly for PS-SVM and DFSA-SVM as shown in Figure 4.2 and 4.3.

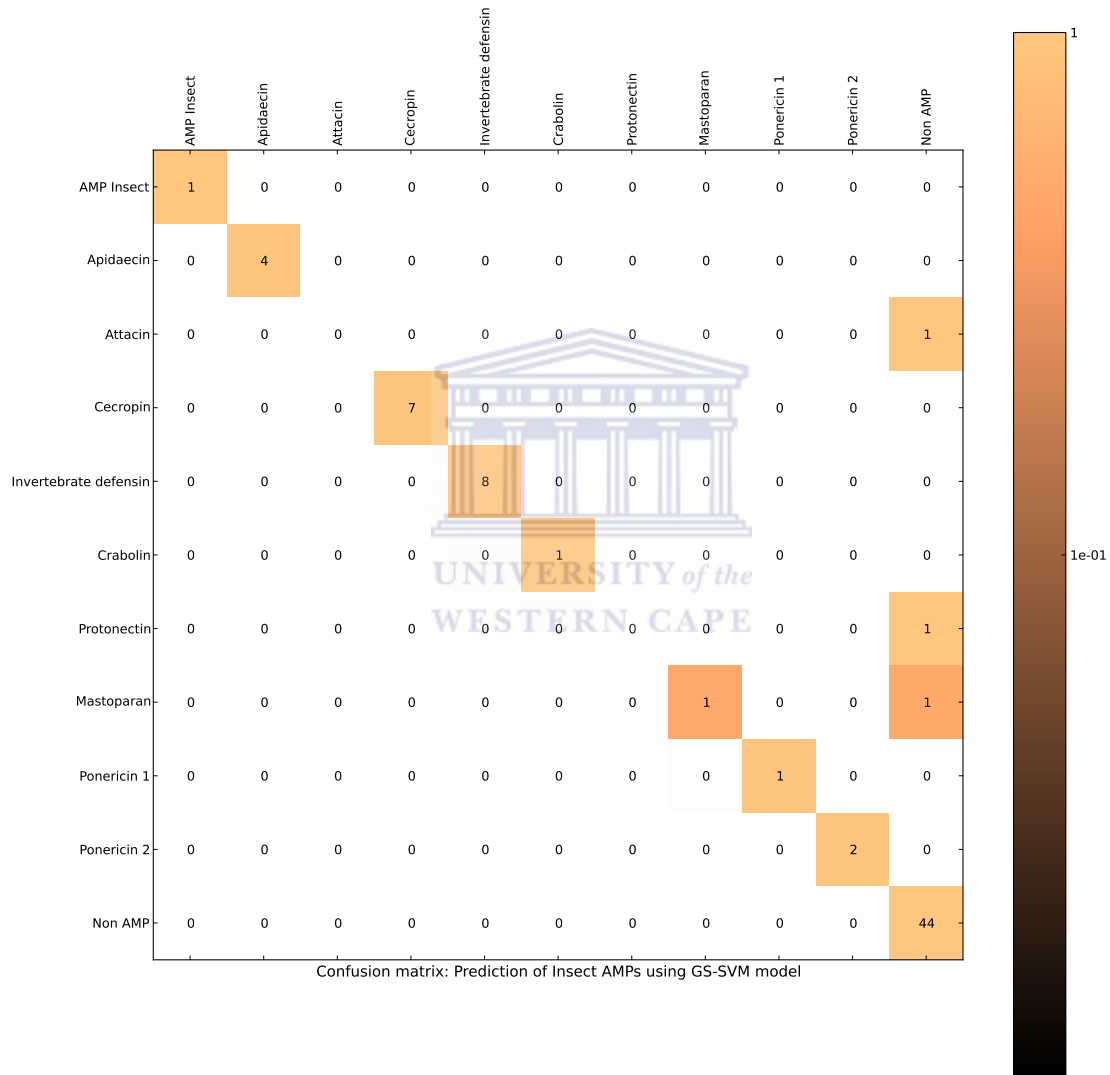


Figure 4.1: Confusion matrix for prediction of insects AMPs using GS model

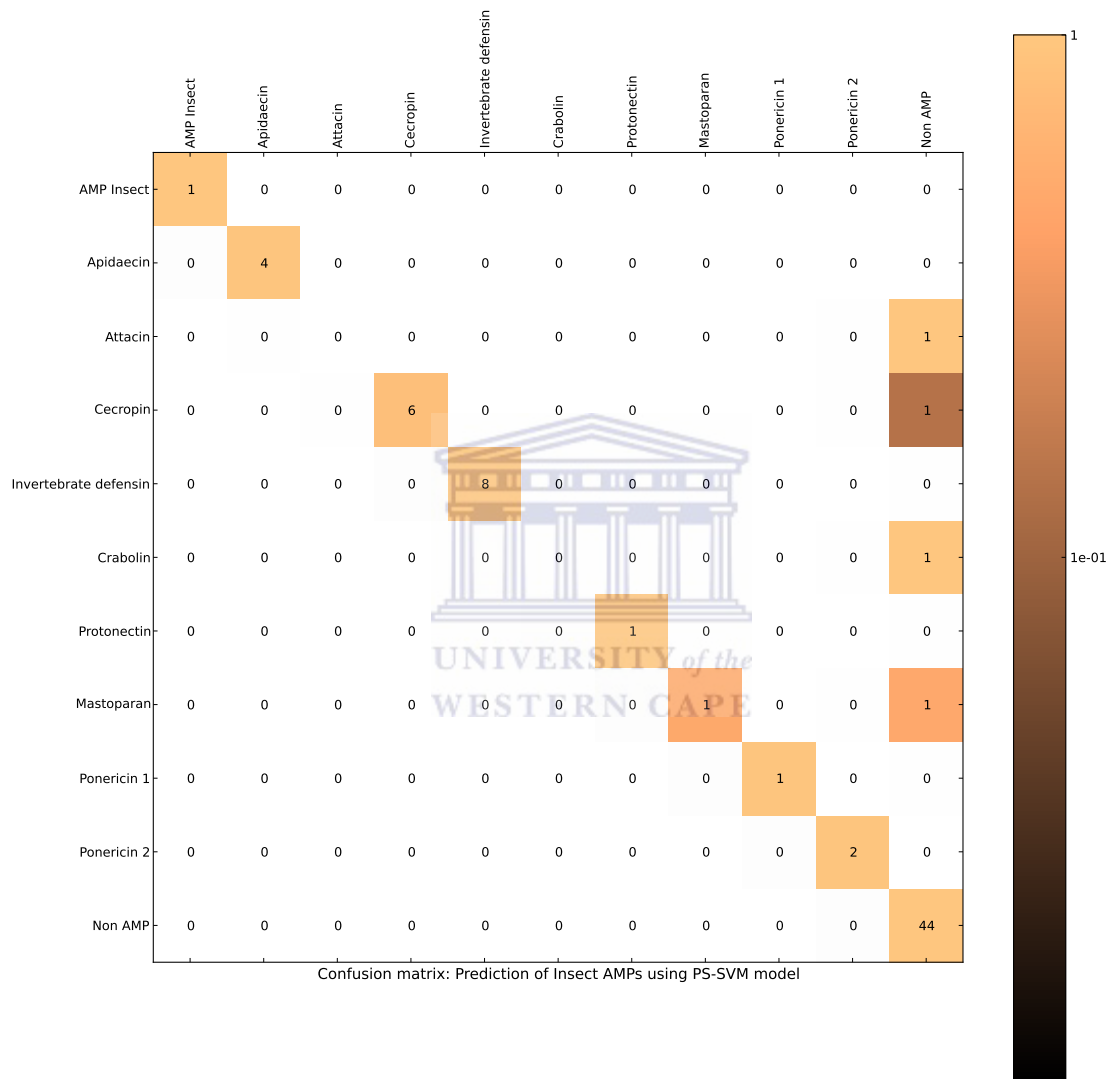


Figure 4.2: Confusion matrix for prediction of insecta AMPs using PS model

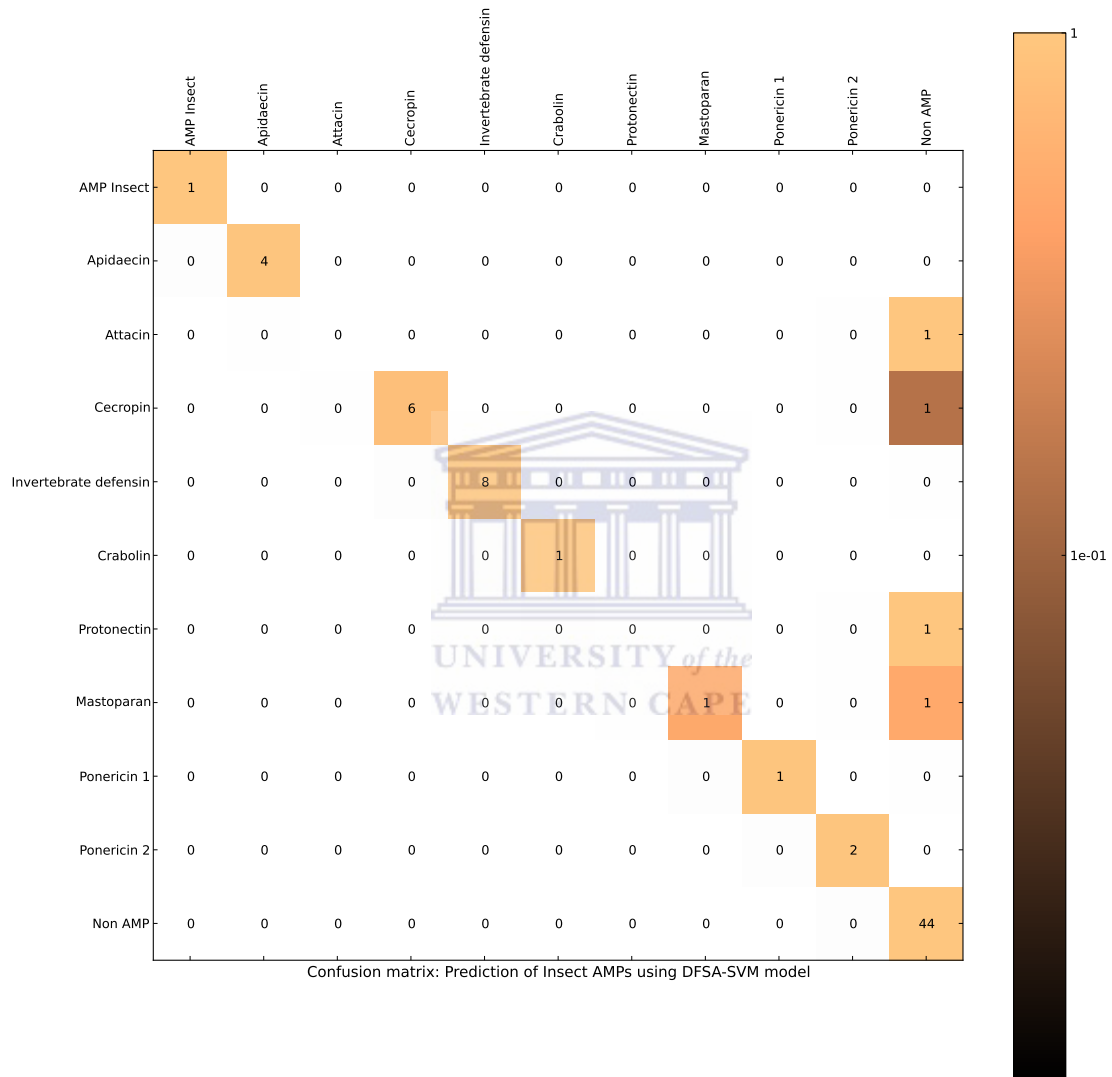


Figure 4.3: Confusion matrix for prediction of insecta AMPs using SAPS model

For each prediction, without well-defined semantics for the SVM raw scores assigned for each prediction of AMPs, it is difficult for users to design downstream experiments. Several methods have been employed to endow raw scores with statistical confidence measure. Some of these methods include Bonferroni correction and *E*-value.

The Bonferroni correction states that if you are aiming for a significance threshold of α but you conduct test m times, then you should adjust your threshold to $\frac{\alpha}{m}$. It is a simple method to implement but it does not only reduce the number of false positive, but also reduce the number of true discoveries (false negative). On the other hand, the *E*-value is the converse of the Bonferroni correction. Instead of dividing the significance threshold (α) by the number of tests performed (m), the *E*-value is the product of α and m . It is anticonservative because this number is too large and hence the false positive rate is too liberal. Between this two extreme points is the false discovery rate. This approach is a relatively recent approach that determines adjusted *p*-values for each test. However, it controls the number of false discoveries in those tests that result on a discovery (significant result). Because of this, it is less conservative than the Bonferroni approach and has greater power to find truly significant results. Another way to look at the difference is that a *p*-value of 0.05 implies that 5% of all tests will result in false positives. An FDR adjusted *p*-value (or *q*-value) of 0.05 implies that 5% of significant test will result in false positives (Käll et al., 2008b; Noble, 2009).

The *q*-value is a measure of significance in terms of the false discovery rate (FDR) rather than the false positive rate. The false positive rate is the rate that truly null examples are called significant whereas the FDR is the rate that significant examples are truly null examples. For instance, a false positive rate of 5% means that on average 5% of the truly null examples in a particular study will be called significant. On the other hand, a FDR of 5% means that among all examples called significant, 5% of these are truly null on average (Storey and Tibshirani, 2003).

A *p*-value threshold of 5% yields a false positive rate of 5% among all null features in the dataset, whereas a *q*-value $\leq 5\%$ means FDR of 5% among the significant features. In the light of definition of the false positive rate, a *p*-value cutoff says little about the content of features actually called significant. The *q*-values directly provide a meaningful measure among the features called significant. Because significant features will likely undergo some subsequent biological verification, a *q*-value threshold can be phrased in practical terms as the proportion of significant features that turn out to be false leads (Noble, 2009; Storey and Tibshirani, 2003).

In this study, a collection of 1284 peptides, i.e., FIXME sequence discussed in section 4.3 was ana-

lyzed. For each peptide, its equivalent SVM score using the insect model is calculated. Figure 4.4 shows the resulting distribution of mixed and null svm scores. An SVM classifier assigns negative examples negative scores and positive examples positive scores. In the figure, the null peptides receive score that are almost entirely negative, however, the mixed peptide distribution has a large set of negative score and a smaller set of positive scores. This observation is consistent with a model in which the set of mixed peptides is comprised of a mixture of correct and incorrect AMPs.

Relationship between q -values and posterior error probability (PEP) for the 284 sequence (mixed) is shown in Figure 4.5. The figure plots the estimated PEP (blue curve) and q -values (red curve) as a function of the SVM score. In this figure, the relationship between PEP and q -value for a real data set, that is, a collection of 1284 classified peptides scores derived from SVM. Setting a PEP threshold of 5% yields 104 significant peptide predictions. Alternatively, setting a threshold of $q = 0.05$, yields 130 significant peptide predictions. Thus for this data set, switching from PEP and q -value yields 25% more identifications.

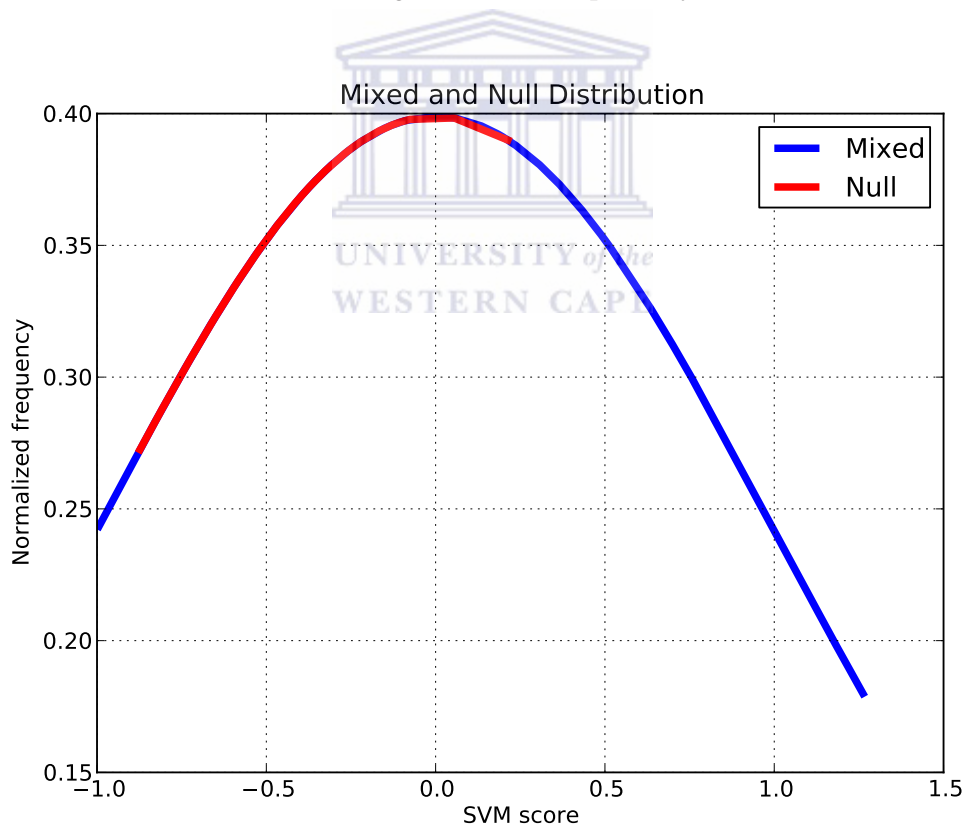


Figure 4.4: The figure represents the distribution of 1284 SVM scores for mixed and null peptides.

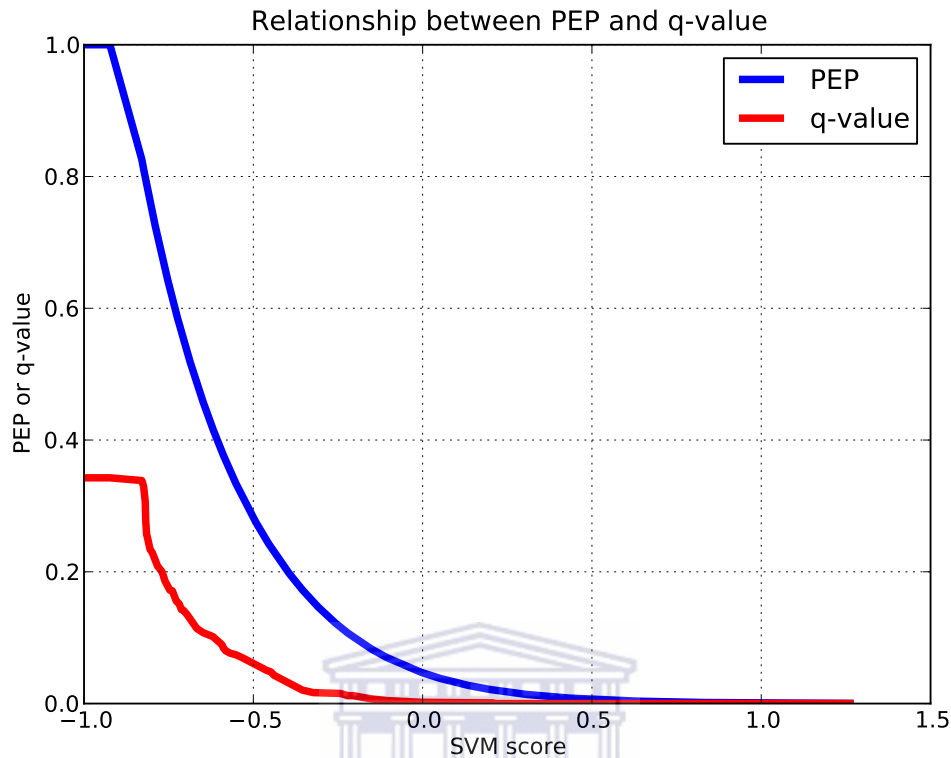


Figure 4.5: PEP and q -values
WESTERN CAPE

Several webservers exist that predict AMPs such as AntiBP2 (Lata et al., 2010) and CAMP (Thomas et al., 2010). AntiBP2 has two limitations, that is,

- the length of the sequence to be analysed should not be more than 100 amino acid long
- there is no statistical confidence measure given for any prediction.

On the other hand, CAMP classifies a query sequence as AMP or not but does not give its respective family.

4.5 Description of the webserver

Since the achieved accuracy is more than 90% in predicting AMP families in haemotophagous insects, it is imperative that these findings provide the research community with a tool that will characterize unknown

peptides by testing membership of an AMP family. For this reason, a webserver for the classification of AMPs into its respective families, under the name of haemotophagous antimicrobial peptide predictor (HAPP) is created. The webserver is hosted by the South African National Bioinformatics Institute and is available at the site <http://apps.sanbi.ac.za/Happ/>.

The user enters the query sequence and choose a particular threshold value and also which hybrid model to use i.e., PS-SVM, GS-SVM or DFSA-SVM (Figure 4.6). Once the user submits the query, the input gets processed at the backend of the webserver, where the raw SVM scores is computed. In addition the complementary statistical measure (PEP) is computed using precomputed values generated from the target and null sequences earlier described. The list in precomputed file consists of SVM scores with its rank ordered PEP and q -values. An example of the results page generated is shown in Figure 4.7 and the output comes in two sections, namely

- **Parameters:** The first section is the header which gives information on what options chosen by the user (SVM raw score threshold, posterior error probability (PEP) threshold and hyperparameter optimization method). In addition, the query sequence is shown.
- **Prediction:** The second section gives the prediction results of the query sequence. The first field indicates the class of the query sequence, be it antimicrobial or not. Then, this is followed by the peptide sub-class of the sequence, in case it is classified as antimicrobial peptide. The last three fields are the SVM raw scores, the estimated posterior error probability (PEP) and the link to the antimicrobial family dataset.

HAPP webserver is the first of its kind for predicting AMPs in haemotophagous insects and has a potential in advancing knowledge of AMPs by providing an interactive way for scientists in the field to quickly determine if a newly sequenced protein is an AMP or not, as well as furnishing with statistical measure for a follow-up assay.

Figure 4.6: The input interface to the HAPP webserver. Users can input the threshold required. Also the user can select from different models generated by GS-SVM, PS-SVM and DFSA-SVM. The query input is a protein.

Class	Peptide family	SVM raw score	PEP	Family http link
Antimicrobial	Apidaecin	1.0274189	0.000753297	Apidaecin(Fasta)

Figure 4.7: The prediction results. The results indicate whether the sequence query belong to the AMP family. If the sequence belongs to the AMP family, it will subsequently indicate the AMP subfamily. The results also display the SVM raw scores, posterior error probability and the link to the AMP family sequence of the prediction results.

4.6 Summary

We predicted AMPs based on models from different families across various taxa rather than using generalized AMP models. Thomas et. al. created a generalized AMP models to predict AMPs. One limitation of their models are not specific and not accurate. On the other hand, the specific models based on AMPs families is more robust and in addition to predicting AMPs with high accuracy, but also classifies them into specific subclasses such as cecropin, defensin, α -defensin etc.

A large-scale test on all of the currently sequenced and publicly available genomes would be useful to ascertain the robust of the methods used to create the webserver. Establishing the possibility that there are more AMPs through *in-silico* as compared to those discovered in laboratories, can provide an additional sense of direction for the wet lab scientist by testing a few predicted AMPs that have high confidence level for their activity.

The webserver will be useful to scan the ongoing genomes for potential AMPs in insects such as *Anopheles gambiae*, *Glossina morsitans*, *Phlebotomus logipalpis*, *Culex quinquefasciatus* and *Anopheles funestus*. These insects are vectors that cause diseases such as trypanosiasis, leishmaniasis, yellow fever and malaria.

One limitation of this study is the lack of enough experimentally validated AMPs that hinders the creation of AMP family models. The other limitation is the small number of sequences used in the target and null databases. The *q*-value and PEP measures depends on the size of the database. The larger the number of sequences in the database that you search, the greater the number of false positives, hence more accurate statistical measure. In future the whole insect proteins from UniProt is intended to be used as the target database in order to generate the null sequences.

Chapter 5

Conclusion and future work

This chapter presents the usage of various computational methods to mine knowledge from the antimicrobial peptides (AMPs) dataset. The main objective of this thesis has been to create AMPs model in order to predict new AMPs. The main contributions of the thesis is broken into three subsections as well as their limitations. In the first subsection, the database of antimicrobial peptides is discussed. In the second subsection, the prediction of AMPs using support vector machines is presented. The section section is on the webservice. Finally, the direction for future work is presented.

5.1 Research contribution and limitations

5.1.1 Antimicrobial peptide database

Databases are useful resource for mining and exploration of antimicrobial peptides, allowing users to query complex biological questions and analysis of data. In this thesis, a comprehensive database of antimicrobial peptides called DAMPD was created. DAMPD is a manually curated database populated with 1232 experimentally validated AMPs entries for both prokaryotic and eukaryotic sources. The procedure for creating the database involves data extraction using keywords and data curation.

The creation of DAMPD database was the first step towards a systematic analysis of AMPs. The DAMPD database was successfully developed and is freely accessible for academic and non-profit users at <http://apps.sanbi.ac.za/dampd>. The DAMPD database contains both search and analytical tools that

ease in search and analysis of biological query. In particular, classification of AMPs using profile hidden Markov model has been implemented. The profiles created can be used to classify new AMP families into known AMP families. HMM profiles were created for AMPs based on prior knowledge of the AMP families.

5.1.2 Classification of AMPs using support vector machines

Data modeling is usually a crucial step in data mining and yield ground for prediction purposes. The curated data in DAMPD was used to create AMPS models in various taxa. In chapter 3, an SVM-based machine learning approach coupled with optimization methods have been implemented to aid in classification of AMPs into their respective AMPs families. Global optimization methods such as grid search, pattern search and derivative-free simulated annealing were used to select the hyperparameters of SVM classifier. PS-SVM was the best hybrid method based on classification accuracy.

5.1.3 Creation of haemotophagous antimicrobial peptide predictor webserver

A webserver to predict haemotophagous insect AMPs into their respective families was created. The webserver is freely accessible at <http://apps.sanbi.ac.za/Happ>. This resource is useful to predict AMPs in ongoing genomes.

Some of the future work include

- enriching the database with additional annotation such as information on promoter region and transcription factors for an AMP. The mode of action of AMPs will be added.
- using string kernels such as profile kernel, spectrum kernel and mismatch kernel instead of amino acid composition and physiochemical properties.
- incorporate feature selection in addition to parameter selection of SVM.
- predict AMPs once the genomes for haemotophagous insect are completed.
- use of modified pattern search method that uses perturbed coordinate directions rather than the spanning direction used in PS.

Appendix A

Supplementary material for Chapter 2

ClustalW results page

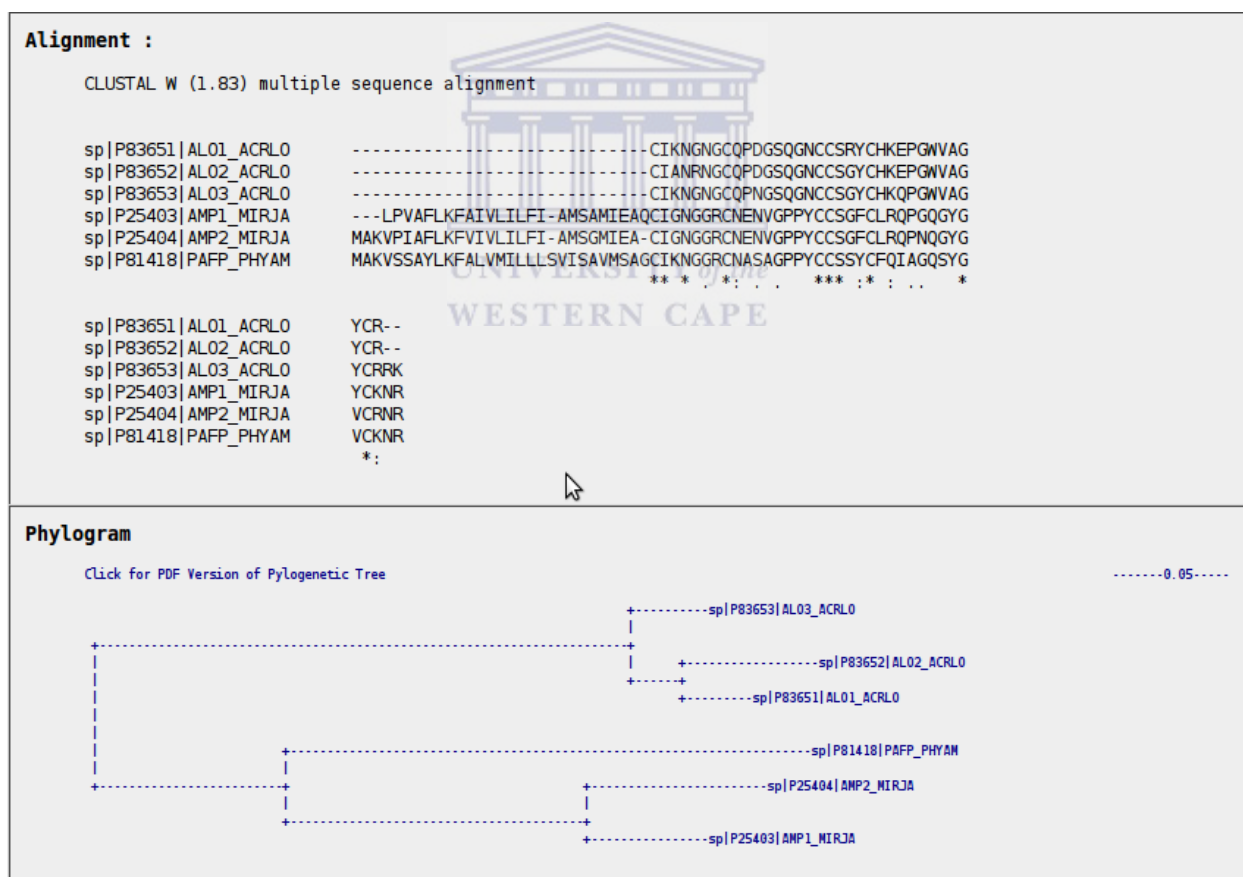


Figure A.1: ClustalW results of AMP family page

HMMER results page

```

Alignment

hmmsearch - search a sequence database with a profile HMM
HMMER 2.3.2 (Oct 2003)
Copyright (C) 1992-2003 HHMI/Washington University School of Medicine
Freely distributed under the GNU General Public License (GPL)
-----
HMM file:                /var/www/dampd/tmp/12225_030411.hmm [12225_030411]
Sequence database:       /var/www/dampd/tmp/12225_030411.query
per-sequence score cutoff: [none]
per-domain score cutoff:  [none]
per-sequence Eval cutoff:  <= 10
per-domain Eval cutoff:   [none]
-----

Query HMM:  12225_030411
Accession:  [none]
Description: [none]
  [HMM has been calibrated; E-values are empirical estimates]

Scores for complete sequences (score includes all domains):
Sequence      Description              Score   E-value  N
-----
sp|Q62715|DEF2_RAT Neutrophil antibiotic peptide NP-2  186.0   1.1e-56  1

Parsed for domains:
Sequence      Domain  seq-f  seq-t    hmm-f  hmm-t    score  E-value
-----
sp|Q62715|DEF2_RAT  1/1      1    94 []      1    92 []    186.0  1.1e-56

```

Figure A.2: Classification results of a query sequence using α -defensin HMM profile.

Hydrocalculator results page

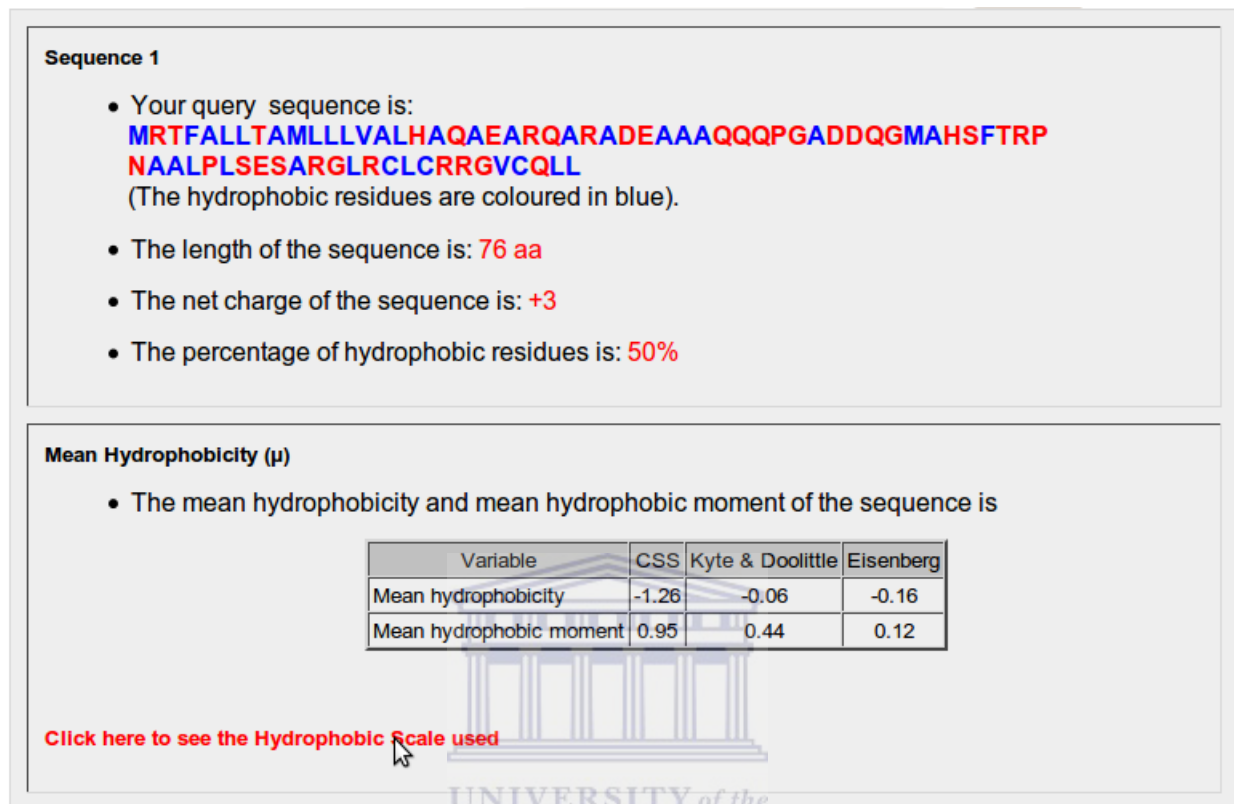
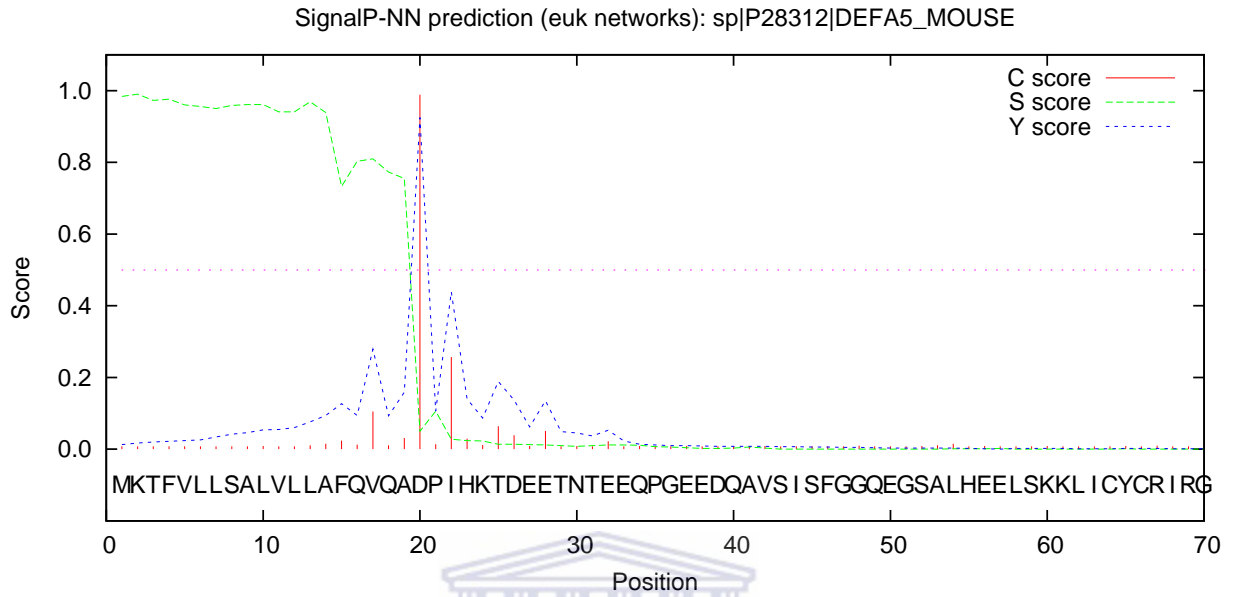
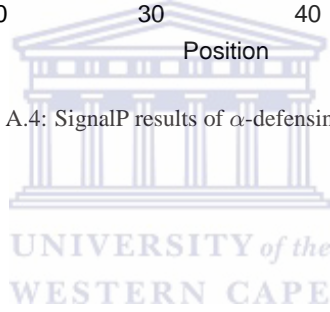


Figure A.3: Hydrocalculator results of α -defensin sequence

SignalP results pageFigure A.4: SignalP results of α -defensin sequence

Appendix B

Supplementary material for Chapter 3

Pattern search method

This example illustrates how the previous Algorithm 1 works in \mathbb{R}^2 . In Figure B.1, x^k is the current iterate at the k^{th} iteration and is represented by the dotted circle \odot . The solid circle \bullet indicates the position of the trial point $p^i \in P^k$ to be examined, where $i = 1, \dots, r$. The small open circle \circ and the circled asterisk \otimes represent unsuccessful and successful trial points respectively of the POLL step. The POLL step begins by evaluating the function value of the trial point $p^i \in P^k$, point by point, where $i = 1, \dots, 4$, as shown in Figure B.1. In Figure 2.2(a), the PS method computes the trial point p^1 by a step of size Δ^k . It computes the function value at p^1 . If $f(p^1) > f(x^k)$ then it examines the next trial point p^2 as shown in Figure 2.2(b). If it is not successful at p^2 , i.e., $f(p^2) > f(x^k)$ then it computes p_3 as shown in Figure 2.2(c). If p^3 is still unsuccessful then the process is repeated until all the trial points in P^k are examined, i.e., until p^4 is computed as shown in Figure 2.2(d). If all the points in the POLL set P^k (i.e., p^1, p^2, p^3 and p^4) are not successful then the step size is reduced by half as shown in Figure 2.2(e), i.e., the next POLL step begins at $x^{k+1} = x^k$ with $\Delta^{k+1} = \frac{1}{2}\Delta^k$. On the other hand, suppose that the trial point p^2 is successful, i.e., $f(p^2) < f(x^k)$ as shown in Figure 2.2(f), then the whole POLL step process starts anew at $x^{k+1} = p^2$ with enlarged step size, i.e., $\Delta^{k+1} = 2\Delta^k$ as shown in Figure 2.2(h). A similar cycle as shown in (a), (b), (c) and (d) of Figure 2.2 will be repeated (if necessary) for the new POLL at x^{k+1} .

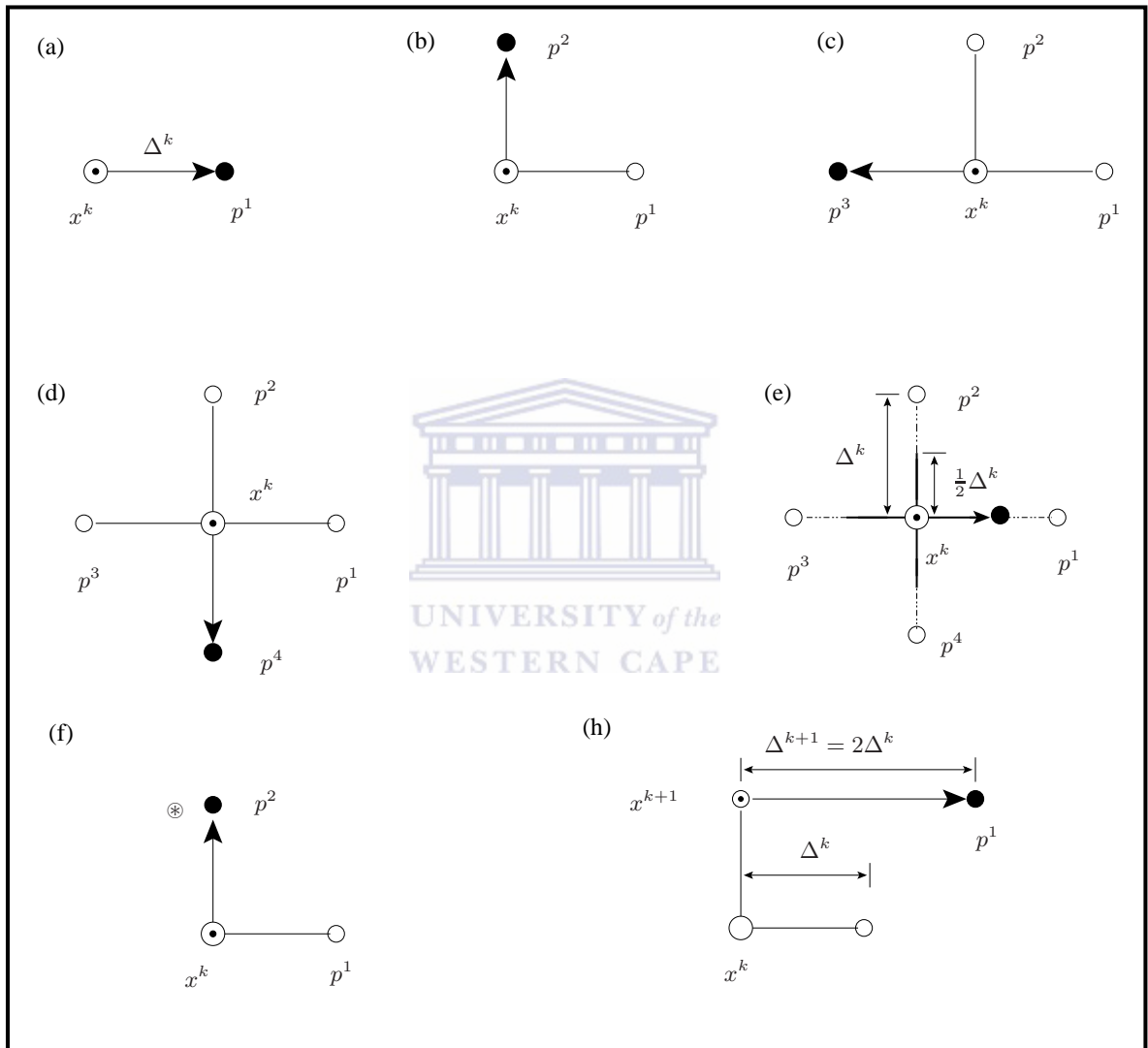


Figure B.1: Figures (a)-(h) shows how the POLL steps works in the PS method.

Grid search method

A grid search tries values of each parameter across the specified search range using geometric steps. Grid searches are computationally expensive because the model must be evaluated at many points within the grid for each parameter. For example, if a grid search is used with 20 search intervals and the svm three parameters (c, σ) then the model must be evaluated at $20 \times 20 = 400$ grid points. If cross-validation is used for each model evaluation, the number of actual SVM calculations would be further multiplied by the number of cross-validation folds. For large models, this approach may be computationally infeasible.

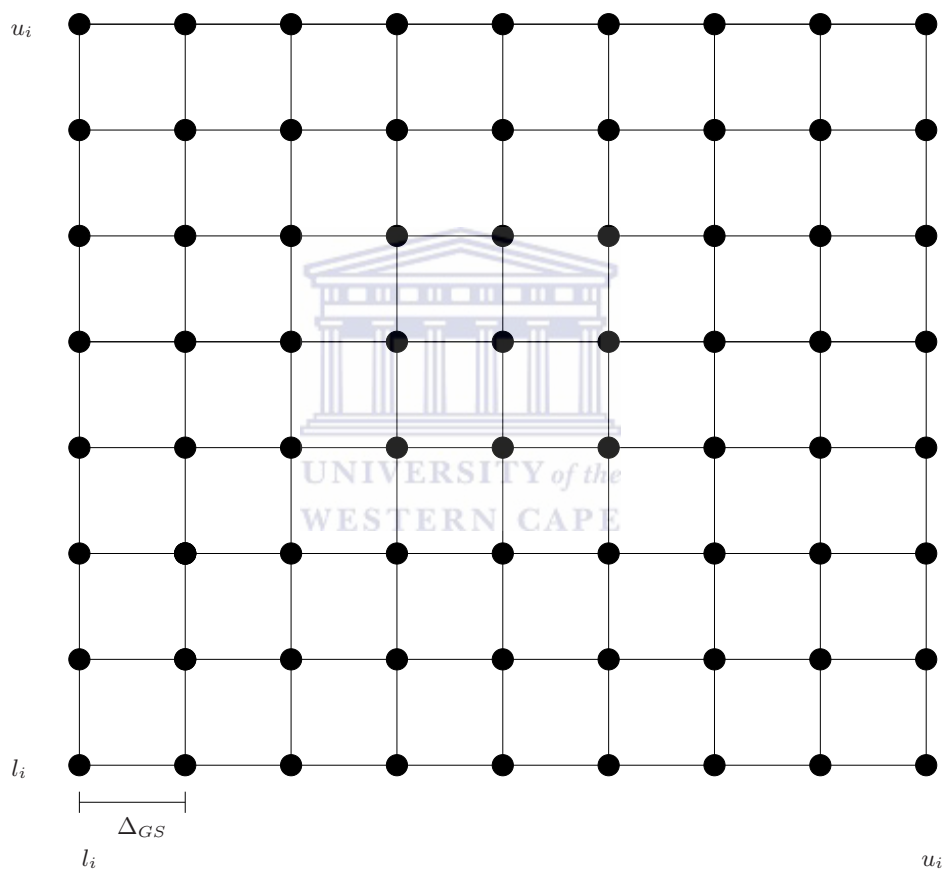


Figure B.2: Figure shows how the Grid Search works in a two dimensional optimization problem

Negative sets

The negative set was downloaded from UniProt using the keywords below:

- actinopterygii (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 100]) AND taxonomy:"Actinopterygii [7898]"
- amphibian (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 100]) AND taxonomy:"Amphibia [8292]"
- arachnida (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 100]) AND taxonomy:"Arachnida [6854]"
- bacteria (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 100]) AND taxonomy:"Eubacterium [1730]"
- crustacea (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 100]) AND taxonomy:"Crustacea [6657]"
- insecta (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 100]) AND taxonomy:"Insecta [50557]"
- mammalia (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 70]) AND taxonomy:"Mammalia [40674]"
- plant (((golgi or cytoplasm or endoplasmic reticulum or mitochondria) NOT antimicrobial) AND length:[0 TO 100]) AND taxonomy:"Viridiplantae [33090]"

Features indices

- **EISD840101 Consensus normalized hydrophobicity scale Kawashima et al. (2008)**

A: 0.25, R: -1.76, N: -0.64, D: -0.72, C: 0.04, Q: -0.69, E: -0.62, G: 0.16, H: -0.40, I: 0.73, L: 0.53, K: -1.10, M: 0.26, F: 0.61, P: -0.07, S: -0.26, T: -0.18, W: 0.37, Y: 0.02, V: 0.54

- **HOPT810101 Hydrophilicity value Kawashima et al. (2008)**

A: -0.5, R: 3.0, N: 0.2, D: 3.0, C: -1.0, Q: 0.2, E: 3.0, G: 0.0, H: -0.5, I: -1.8, L: -1.8, K: 3.0, M: -1.3, F: -2.5, P: 0.0, S: 0.3, T: -0.4, W: -3.4, Y: -2.3, V: -1.5

- **VELV850101 Electron-ion interaction potential Kawashima et al. (2008)**

A: .03731, R: .09593, N: .00359, D: .12630, C: .08292, Q: .07606, E: .00580, G: .00499, H: .02415, I: .00000, L: .00000, K: .03710, M: .08226, F: .09460, P: .01979, S: .08292, T: .09408, W: .05481, Y: .05159, V: .00569

- **ZIMJ680101 Hydrophobicity Kawashima et al. (2008)**

A: 0.83, R: 0.83, N: 0.09, D: 0.64, C: 1.48, Q: 0.00, E: 0.65, G: 0.10, H: 1.10, I: 3.07, L: 2.52, K: 1.60, M: 1.40, F: 2.75, P: 2.70, S: 0.14, T: 0.54, W: 0.31, Y: 2.97, V: 1.79

- **ZIMJ680102 Bulkiness Kawashima et al. (2008)**

A: 11.50, R: 14.28, N: 12.82, D: 11.68, C: 13.46, Q: 14.45, E: 13.57, G: 3.40, H: 13.69, I: 21.40, L: 21.40, K: 15.71, M: 16.25, F: 19.80, P: 17.43, S: 9.47, T: 15.77, W: 21.67, Y: 18.03, V: 21.57

- **ZIMJ680103 Polarity Kawashima et al. (2008)**

A: 0.00, R: 52.00, N: 3.38, D: 49.70, C: 1.48, Q: 3.53, E: 49.90, G: 0.00, H: 51.60, I: 0.13, L: 0.13, K: 49.50, M: 1.43, F: 0.35, P: 1.58, S: 1.67, T: 1.66, W: 2.10, Y: 1.61, V: 0.13

- **JURD980101 Modified Kyte-Doolittle hydrophobicity scale Kawashima et al. (2008)**

A: 1.10, R: -5.10, N: -3.50, D: -3.60, C: 2.50, Q: -3.68, E: -3.20, G: -0.64, H: -3.20, I: 4.50, L: 3.80, K: -4.11, M: 1.90, F: 2.80, P: -1.90, S: -0.50, T: -0.70, W: -0.46, Y: -1.3, V: 4.2

References

- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–8.
- Abramson, M. A., Audet, C., and Dennis, J. (2004). Generalized pattern searches with derivative information. *Mathematical Programming*, 100:3–25.
- Acharya, U. R., Ng, E. Y. K., Tan, J.-H., Sree, S. V., and Ng, K.-H. (2011). An integrated index for the identification of diabetic retinopathy stages using texture parameters. *J Med Syst*.
- Akuffo, H., Hultmark, D., EngstÄm, A., Frohlich, D., and Kimbrell, D. (1998). Drosophila antibacterial protein, cecropin a, differentially affects non-bacterial organisms such as leishmania in a manner different from other amphipathic peptides. *Int J Mol Med*, 1(1):77–82.
- Ali, M. M. and Gabere, M. N. (2010). A simulated annealing driven multi-start algorithm for bound constrained global optimization. *J. Comput. Appl. Math.*, 233(10):2661–2674.
- Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–10.
- Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–402.
- Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Hermjakob, H., Hulo, N., Jonassen, I., Kahn, D., Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N. J., Oinn, T. M., Pagni, M., Servant, F., Sigrist, C. J., Zdobnov, E. M., and InterPro Consortium (2000). Interpro—an integrated documentation resource for protein families, domains and functional sites. *Bioinformatics*, 16(12):1145–50.

- Audet, C., Jr., J. E. D., and Le Digabel, S. (2008). Parallel space decomposition of the mesh adaptive direct search algorithm. *SIAM J. Optim.*, 19(3):1150–1170.
- Bairoch, A. and Apweiler, R. (2000). The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic Acids Res*, 28(1):45–8.
- Barutcuoglu, Z., Schapire, R. E., and Troyanskaya, O. G. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22:830–836.
- Bellamy, W., Takase, M., Wakabayashi, H., Kawase, K., and Tomita, M. (1992). Antibacterial spectrum of lactoferricin b, a potent bactericidal peptide derived from the n-terminal region of bovine lactoferrin. *J Appl Bacteriol*, 73(6):472–9.
- Ben-Hur, A. and Brutlag, D. (2003). Remote homology detection: a motif based approach. *Bioinformatics*, 19 Suppl 1:i26–33.
- Bendtsen, J. D., Nielsen, H., von Heijne, G., and Brunak, S. (2004). Improved prediction of signal peptides: Signalp 3.0. *J Mol Biol*, 340(4):783–95.
- Bjellqvist, B., Basse, B., Olsen, E., and Celis, J. E. (1994). Reference points for comparisons of two-dimensional maps of proteins from different human cell types defined in a ph scale where isoelectric points correlate with polypeptide compositions. *Electrophoresis*, 15(3-4):529–39.
- Boisvert, S., Marchand, M., Laviolette, F., and Corbeil, J. (2008). Hiv-1 coreceptor usage prediction without multiple alignments: an application of string kernels. *Retrovirology*, 5:110.
- Boman, H. G. (2000). Innate immunity and the normal microflora. *Immunol Rev*, 173:5–16.
- Boser, B. E., Guyon, I. M., and Vapnik, N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory, COLT '92*, pages 144–152, New York, NY, USA. ACM.
- Boulanger, N., Brun, R., Ehret-Sabatier, L., Kunz, C., and Bulet, P. (2002). Immunopeptides in the defense reactions of glossina morsitans to bacterial and trypanosoma brucei brucei infections. *Insect Biochem Mol Biol*, 32(4):369–75.
- Brahmachary, M., Krishnan, S. P. T., Koh, J. L. Y., Khan, A. M., Seah, S. H., Tan, T. W., Brusica, V., and Bajic, V. B. (2004). Antimic: a database of antimicrobial sequences. *Nucleic Acids Res*, 32(Database issue):D586–9.

- Brahmachary, M., SchÄnbach, C., Yang, L., Huang, E., Tan, S. L., Chowdhary, R., Krishnan, S. P. T., Lin, C.-Y., Hume, D. A., Kai, C., Kawai, J., Carninci, P., Hayashizaki, Y., and Bajic, V. B. (2006). Computational promoter analysis of mouse, rat and human antimicrobial peptide-coding genes. *BMC Bioinformatics*, 7 Suppl 5:S8.
- Breiman, L. (2001). Random forests. pages 5–32.
- Brogden, K. A. (2005). Antimicrobial peptides: pore formers or metabolic inhibitors in bacteria? *Nat Rev Micro*, 3:238–250.
- Bulet, P., Dimarcq, J. L., Hetru, C., Lagueux, M., Charlet, M., Hegy, G., Van Dorsselaer, A., and Hoffmann, J. A. (1993). A novel inducible antibacterial peptide of drosophila carries an o-glycosylated substitution. *J Biol Chem*, 268(20):14893–7.
- Bulet, P., StÄcklin, R., and Menin, L. (2004). Anti-microbial peptides: from invertebrates to vertebrates. *Immunol Rev*, 198:169–84.
- Carter, V. and Hurd, H. (2010). Choosing anti-plasmodium molecules for genetically modifying mosquitoes: focus on peptides. *Trends Parasitol*, 26(12):582–90.
- Chen, W. and Luo, L. (2009). Classification of antimicrobial peptide using diversity measure with quadratic discriminant analysis. *J Microbiol Methods*, 78(1):94–6.
- Cole, A. M. and Ganz, T. (2000). Human antimicrobial peptides: analysis and application. *Biotechniques*, 29(4):822–6, 828, 830–1.
- Colinge, J., Masselot, A., Giron, M., Dessingy, T., and Magnin, J. (2003). Olav: towards high-throughput tandem mass spectrometry data identification. *Proteomics*, 3(8):1454–63.
- Cristianini, N. and Shawe-Taylor, J. (2001). *An introduction to support vector machines and other kernel-based learning methods. Repr.* Cambridge: Cambridge University Press.
- Damaševičius, R. (2010). Optimization of SVM parameters for recognition of regulatory DNA sequences. *Top*, 18(2):339–353.
- de Jong, A., van Heel, A. J., Kok, J., and Kuipers, O. P. (2010). Bagel2: mining for bacteriocins in genomic data. *Nucleic Acids Res*, 38(Web Server issue):W647–51.

- de Jong, A., van Hijum, S. A. F. T., Bijlsma, J. J. E., Kok, J., and Kuipers, O. P. (2006). Bagel: a web-based bacteriocin genome mining tool. *Nucleic Acids Res*, 34(Web Server issue):W273–9.
- Dekkers, A. (1991). Global optimization and simulated annealing. *Mathematical Programming.*, 50:367–393.
- Duan, K., Keerthi, S. S., and Poo, A. N. (2003). Evaluation of simple performance measures for tuning svm hyperparameters. *Neurocomputing*, 51:41–59.
- Dutton, C. J., Haxell, M. A., McArthur, H. A. I., and Wax, R. G. (2002). *Peptide Antibiotics. Discovery, Modes of Action and Applications*. Marcel Dekker, New York, NY, USA,.
- Eddy, S. R. (1998). Profile hidden markov models. *Bioinformatics*, 14(9):755–63.
- Efron, B. (2004). Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *J. Am. Stat. Assoc.*, 99(465):96–104.
- Ferre, R., Badosa, E., Feliu, L., Planas, M., Montesinos, E., and Bardají, E. (2006). Inhibition of plant-pathogenic bacteria by short synthetic cecropin a-melittin hybrid peptides. *Appl Environ Microbiol*, 72(5):3302–8.
- Finn, R. D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J. E., Gavin, O. L., Gunasekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E. L. L., Eddy, S. R., and Bateman, A. (2010). The pfam protein families database. *Nucleic Acids Res*, 38(Database issue):D211–22.
- Fjell, C. D., Hancock, R. E. W., and Cherkasov, A. (2007). Amper: a database and an automated discovery tool for antimicrobial peptides. *Bioinformatics*, 23(9):1148–55.
- Gabere, M. N. (2007). Simulated annealing driven pattern search algorithms for global optimization. Master’s thesis, School of Computational and Applied Mathematics, University of the Witwatersrand, Johannesburg.
- Ganz, T. (2003). Defensins: antimicrobial peptides of innate immunity. *Nat Rev Immunol*, 3(9):710–20.
- Garcia-Olmedo, F., Molina, A., Alamillo, J. M., and Rodriguez-Palenzuela, P. (1998). Plant defense peptides. *Biopolymers*, 47(6):479–91.
- Garrido, C., Roulet, V., Chueca, N., Poveda, E., Aguilera, A., Skrabal, K., Zahonero, N., Carlos, S., Garc a, F., Faudon, J. L., Soriano, V., and de Mendoza, C. (2008). Evaluation of eight different bioin-

- formatics tools to predict viral tropism in different human immunodeficiency virus type 1 subtypes. *J Clin Microbiol*, 46(3):887–91.
- Gasteiger, E., Hoogland, C., Gattiker, A., Duvaud, S., Wilkins, M. R., Appel, R. D., and Bairoch, A. (2005). *Protein Identification and Analysis Tools on the ExPASy Server*. The Proteomics Protocols Handbook, Humana Press.
- Ghosh, J. K., Shaool, D., Guillaud, P., Cic aron, L., Mazier, D., Kustanovich, I., Shai, Y., and Mor, A. (1997). Selective cytotoxicity of dermaseptin s3 toward intraerythrocytic plasmodium falciparum and the underlying molecular basis. *J Biol Chem*, 272(50):31609–16.
- Giangaspero, A., Sandri, L., and Tossi, A. (2001). Amphipathic alpha helical antimicrobial peptides. *Eur J Biochem*, 268(21):5589–600.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., and Bloomfield, C. D. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, pages 531–537.
- Guani-Guerra, E., Santos-Mendoza, T., Lugo-Reyes, S. O., and Ter an, L. M. (2010). Antimicrobial peptides: general overview and clinical implications in human health and disease. *Clin Immunol*, 135(1):1–11.
- Gueguen, Y., Garnier, J., Robert, L., Lefranc, M.-P., Mougnot, I., de Lorgeril, J., Janech, M., Gross, P. S., Warr, G. W., Cuthbertson, B., Barracco, M. A., Bulet, P., Aumelas, A., Yang, Y., Bo, D., Xiang, J., Tassanakajon, A., Piquemal, D., and Bach ere, E. (2006). Penbase, the shrimp antimicrobial peptide penaeidin database: sequence-based classification and recommended nomenclature. *Dev Comp Immunol*, 30(3):283–8.
- Hammami, R., Ben Hamida, J., Vergoten, G., and Fliss, I. (2009). Phytamp: a database dedicated to antimicrobial plant peptides. *Nucleic Acids Res*, 37(Database issue):D963–8.
- Hammami, R., Zouhir, A., Ben Hamida, J., and Fliss, I. (2007). Bactibase: a new web-accessible database for bacteriocin characterization. *BMC Microbiol*, 7:89.
- Hancock, R. E. and Diamond, G. (2000). The role of cationic antimicrobial peptides in innate host defences. *Trends Microbiol*, 8(9):402–10.

- Hancock, R. E. and Lehrer, R. (1998). Cationic peptides: a new source of antibiotics. *Trends Biotechnol*, 16(2):82–8.
- Hancock, R. E. W. and Chapple, D. S. (1999). Peptide antibiotics. *Antimicrobial Agents Chemotherapy*, 43(6):1317–1323.
- Harris, D. J. (2003). Can you bank on genbank? *Trends in Ecology & Evolution*, 18(7):317–319.
- Hedar, A. and Fukushima, M. (2004). Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization. *Optim. Methods Softw.*, 19(3-4):291–308.
- Hoffmann, J. A. and Hetru, C. (1992). Insect defensins: inducible antibacterial peptides. *Immunol Today*, 13(10):411–5.
- Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D. P., Kania, R., Schaeffer, M., St Pierre, S., Twigger, S., White, O., and Rhee, S. Y. (2008). Big data: The future of biocuration. *Nature*, 455(7209):47–50.
- Hu, Y. and Aksoy, S. (2005). An antimicrobial peptide with trypanocidal activity characterized from *Glossina morsitans morsitans*. *Insect Biochem Mol Biol*, 35(2):105–15.
- Huang, J. and Ling, C. X. (2007). Constructing new and better evaluation measures for machine learning.
- Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *J Comput Biol*, 7(1-2):95–114.
- Jin, B., Muller, B., Zhai, C., and Lu, X. (2008). Multi-label literature classification based on the gene ontology graph. *BMC Bioinformatics*, 9:1–15. 10.1186/1471-2105-9-525.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, 1398:137–142.
- Joachims, T. (1999). Making large-scale svm learning practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.
- Käll, L., Storey, J. D., MacCoss, M. J., and Noble, W. S. (2008a). Assigning significance to peptides identified by tandem mass spectrometry using decoy databases. *J Proteome Res*, 7(1):29–34.
- Käll, L., Storey, J. D., MacCoss, M. J., and Noble, W. S. (2008b). Posterior error probabilities and false discovery rates: two sides of the same coin. *J Proteome Res*, 7(1):40–4.

- Käll, L., Storey, J. D., and Noble, W. S. (2008c). Non-parametric estimation of posterior error probabilities associated with peptides identified by tandem mass spectrometry. *Bioinformatics*, 24(16):i42–8.
- Käll, L., Storey, J. D., and Noble, W. S. (2009). Qquality: non-parametric estimation of q-values and posterior error probabilities. *Bioinformatics*, 25(7):964–6.
- Kamysz, W., Okrój, M., and Łukasiak, J. (2003). Novel properties of antimicrobial peptides. *Acta Biochim Pol*, 50(2):461–9.
- Kapetanovic, I. M., Rosenfeld, S., and Izmirlian, G. (2004). Overview of commonly used bioinformatics methods and their applications. *Ann N Y Acad Sci*, 1020:10–21.
- Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M. (2008). Aaindex: amino acid index database, progress report 2008. *Nucleic Acids Res*, 36(Database issue):D202–5.
- Klammer, A. A. and MacCoss, M. J. (2006). Effects of modified digestion schemes on the identification of proteins from complex mixtures. *J Proteome Res*, 5(3):695–700.
- Kolda, T. G., Lewis, R. M., and Torczon, V. (2003). Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482.
- Korn, F., Sidiropoulos, N., Faloutsos, C., Siegel, E., and Protopapas, Z. (2007). Fast nearest neighbor search in medical image databases. pages 215–226.
- Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., and Leslie, C. (2005). Profile-based string kernels for remote homology detection and motif extraction. *J Bioinform Comput Biol*, 3(3):527–50.
- Kyrpides, N. C. (1999). Genomes online database (gold 1.0): a monitor of complete and ongoing genome projects world-wide. *Bioinformatics*, 15(9):773–4.
- Lata, S., Mishra, N. K., and Raghava, G. P. S. (2010). Antip2: improved version of antibacterial peptide prediction. *BMC Bioinformatics*, 11 Suppl 1:S19.
- Lata, S., Sharma, B. K., and Raghava, G. P. S. (2007). Analysis and prediction of antibacterial peptides. *BMC Bioinformatics*, 8:263.
- Lawson, D., Arensburger, P., Atkinson, P., Besansky, N. J., Bruggner, R. V., Butler, R., Campbell, K. S., Christophides, G. K., Christley, S., Dialynas, E., Hammond, M., Hill, C. A., Konopinski, N., Lobo,

- N. F., MacCallum, R. M., Madey, G., Megy, K., Meyer, J., Redmond, S., Severson, D. W., Stinson, E. O., Topalis, P., Birney, E., Gelbart, W. M., Kafatos, F. C., Louis, C., and Collins, F. H. (2009). Vectorbase: a data resource for invertebrate vector genomics. *Nucleic Acids Res*, 37(Database issue):D583–7.
- Lee, S. Y., Kim, S., Kim, S. S., Cha, S. J., Kwon, Y. K., Moon, B. R., and Lee, B. J. (2004). Application of decision tree for the classification of antimicrobial peptide. *Genomics and Informatics*, 2(3):121–125.
- Lehrer, R. I. and Ganz, T. (2002). Defensins of vertebrate animals. *Curr Opin Immunol*, 14(1):96–102.
- Leslie, C., Eskin, E., and Noble, W. S. (2002). The spectrum kernel: a string kernel for svm protein classification. *Pac Symp Biocomput*, pages 564–75.
- Leslie, C. S., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–76.
- Letunic, I., Doerks, T., and Bork, P. (2009). Smart 6: recent updates and new developments. *Nucleic Acids Res*, 37(Database issue):D229–32.
- Li, W. and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–9.
- Li, W., Jaroszewski, L., and Godzik, A. (2002). Tolerating some redundancy significantly speeds up clustering of large protein databases. *Bioinformatics*, 18(1):77–82.
- Liao, L. and Noble, W. S. (2002). Combining pairwise sequence similarity and support vector machines for remote protein homology detection. pages 225–232.
- Lin, S., Lee, Z., Chen, S., and Tseng, T. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing*, 8(4):1505–1512. *Soft Computing for Dynamic Data Mining*.
- Matsuzaki, K. (1999). Why and how are peptide-lipid interactions utilized for self-defense? magainins and tachyplesins as archetypes. *Biochim Biophys Acta*, 1462(1-2):1–10.
- Maxwell, A. I., Morrison, G. M., and Dorin, J. R. (2003). Rapid sequence divergence in mammalian beta-defensins by adaptive evolution. *Mol Immunol*, 40(7):413–21.
- Mika, S. and Rost, B. (2004). Protein names precisely peeled off free text. *Bioinformatics*, 20 Suppl 1:i241–7.

- Momma, M. and Bennett, K. P. (2002). A pattern search method for model selection of support vector regression.
- Moore, R. E., Young, M. K., and Lee, T. D. (2002). Qscore: an algorithm for evaluating sequest database search results. *J Am Soc Mass Spectrom*, 13(4):378–86.
- Mulvenna, J. P., Wang, C., and Craik, D. J. (2006). Cybase: a database of cyclic protein sequence and structure. *Nucleic Acids Res*, 34(Database issue):D192–4.
- Nagarajan, V., Kaushik, N., Murali, B., Zhang, C., Lakhera, S., Elasri, M. O., and Deng, Y. (2006). A fourier transformation based method to mine peptide space for antimicrobial activity. *BMC Bioinformatics*, 7 Suppl 2:S2.
- Noble, W. S. (2006). What is a support vector machine? *Nat Biotechnol*, 24(12):1565–7.
- Noble, W. S. (2009). How does multiple testing correction work? *Nat Biotechnol*, 27(12):1135–7.
- Pan, W. (2002). A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics*, 18(4):546–54.
- Park, I. Y., Park, C. B., Kim, M. S., and Kim, S. C. (1998). Parasin i, an antimicrobial peptide derived from histone h2a in the catfish, *parasilurus asotus*. *FEBS Lett*, 437(3):258–62.
- Passerini, A., Punta, M., Ceroni, A., Rost, B., and Frasconi, P. (2006). Identifying cysteines and histidines in transition-metal-binding sites using support vector machines and neural networks. *Proteins*, 65(2):305–16.
- Pavlidis, P., Cai, J., Weston, J., and Grundy, W. N. Wn: Gene functional classification from heterogeneous data.
- Perrière, G. and Gouy, M. (1996). Www-query: an on-line retrieval system for biological sequence banks. *Biochimie*, 78(5):364–9.
- Pieper, U., Eswar, N., Webb, B. M., Eramian, D., Kelly, L., Barkan, D. T., Carter, H., Mankoo, P., Karchin, R., Marti-Renom, M. A., Davis, F. P., and Sali, A. (2009). Modbase, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res*, 37(Database issue):D347–54.
- Popescul, A., Popescul, R., and Ungar, L. H. (2003). Structural logistic regression for link analysis.

- Pounds, S. and Morris, S. W. (2003). Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. *Bioinformatics*, 19(10):1236–42.
- Prosperi, M. C. F., Fanti, I., Ulivi, G., Micarelli, A., De Luca, A., and Zazzi, M. (2009). Robust supervised and unsupervised statistical learning for hiv type 1 coreceptor usage analysis. *AIDS Res Hum Retroviruses*, 25(3):305–14.
- Rinaldi, A. C. (2002). Antimicrobial peptides from amphibian skin: an expanding scenario. *Curr Opin Chem Biol*, 6(6):799–804.
- Samakovlis, C., Kimbrell, D. A., Kylsten, P., Engström, A., and Hultmark, D. (1990). The immune response in drosophila: pattern of cecropin expression and biological activity. *EMBO J*, 9(9):2969–76.
- Samanta, B., Al-Balushi, K. R., and Al-Araimi, S. A. (2006). Artificial neural networks and genetic algorithm for bearing fault detection. *Soft Comput*, 10(3):264–271.
- Sander, O., Sing, T., Sommer, I., Low, A. J., Cheung, P. K., Harrigan, P. R., Lengauer, T., and Domingues, F. S. (2007). Structural descriptors of gp120 v3 loop for the prediction of hiv-1 coreceptor usage. *PLoS Comput Biol*, 3(3):e58.
- Sasson, O., Vaaknin, A., Fleischer, H., Portugaly, E., Bilu, Y., Linial, N., and Linial, M. (2003). Protonet: hierarchical classification of the protein space. *Nucleic Acids Res*, pages 348–352.
- Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., and Dzeroski, S. (2010). Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, 11(1):2.
- Schoes, A. M., Brown, S. D., Dodevski, I., and Babbitt, P. C. (2009). Annotation error in public databases: misannotation of molecular function in enzyme superfamilies. *PLoS Comput Biol*, 5(12):e1000605.
- Seebah, S., Suresh, A., Zhuo, S., Choong, Y. H., Chua, H., Chuon, D., Beuerman, R., and Verma, C. (2007). Defensins knowledgebase: a manually curated database and information source focused on the defensins family of antimicrobial peptides. *Nucleic Acids Res*, 35(Database issue):D265–8.
- Shai, Y. (2002). Mode of action of membrane active antimicrobial peptides. *Biopolymers*, 66(4):236–48.
- Sigrist, C. J. A., Cerutti, L., de Castro, E., Langendijk-Genevaux, P. S., Bulliard, V., Bairoch, A., and Hulo, N. (2010). Prosite, a protein domain database for functional characterization and annotation. *Nucleic Acids Res*, 38(Database issue):D161–6.

- Simmaco, M., Mignogna, G., and Barra, D. (1998). Antimicrobial peptides from amphibian skin: what do they tell us? *Biopolymers*, 47(6):435–50.
- Simpson, P. K. (1990). *Artificial Neural Systems*. Pergamon Press.
- Skrabal, K., Low, A. J., Dong, W., Sing, T., Cheung, P. K., Mammano, F., and Harrigan, P. R. (2007). Determining human immunodeficiency virus coreceptor use in a clinical setting: degree of correlation between two phenotypic assays and a bioinformatic model. *J Clin Microbiol*, 45(2):279–84.
- Soong, T.-T., Wrzeszczynski, K. O., and Rost, B. (2008). Physical protein-protein interactions predicted from microarrays. *Bioinformatics*, 24(22):2608–14.
- Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., and Levy, S. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, 21(5):631–43.
- Steiner, H., Hultmark, D., Engström, Å., Bennich, H., and Boman, H. G. (1981). Sequence and specificity of two antibacterial proteins involved in insect immunity. *Nature*, 292(5820):246–8.
- Storey, J. D. and Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proc Natl Acad Sci U S A*, 100(16):9440–5.
- Strimmer, K. (2008a). fdrtool: a versatile r package for estimating local and tail area-based false discovery rates. *Bioinformatics*, 24(12):1461–2.
- Strimmer, K. (2008b). A unified approach to false discovery rate estimation. *BMC Bioinformatics*, 9:303.
- Thomas, S., Karnik, S., Barai, R. S., Jayaraman, V. K., and Idicula-Thomas, S. (2010). Camp: a useful resource for research on antimicrobial peptides. *Nucleic Acids Res*, 38(Database issue):D774–80.
- Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–80.
- Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622.
- Torczon, V. (1991). On the convergence of the multidirectional search algorithm.

- Tossi, A., Sandri, L., and Giangaspero, A. (2002). New consensus hydrophobicity scale extended to non-proteinogenic amino acids. *Peptide*, pages 416–417.
- Valanne, S., Wang, J.-H., and R met, M. (2011). The drosophila toll signaling pathway. *J Immunol*, 186(2):649–56.
- van 't Hof, W., Veerman, E. C., Helmerhorst, E. J., and Amerongen, A. V. (2001). Antimicrobial peptides: properties and applicability. *Biol Chem*, 382(4):597–619.
- Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
- Verma, C., Seebah, S., Low, S. M., Zhou, L., Liu, S. P., Li, J., and Beuerman, R. W. (2007). Defensins: antimicrobial peptides for therapeutic development. *Biotechnol J*, 2(11):1353–9.
- Vizioli, J. and Salzet, M. (2002). Antimicrobial peptides from animals: focus on invertebrates. *Trends Pharmacol Sci*, 23(11):494–6.
- Wade, D. and Englund, J. (2002). Synthetic antibiotic peptides database. *Protein Pept Lett*, 9(1):53–7.
- Wang, C. K. L., Kaas, Q., Chiche, L., and Craik, D. J. (2008a). Cybase: a database of cyclic protein sequences and structures, with applications in protein discovery and engineering. *Nucleic Acids Res*, 36(Database issue):D206–10.
- Wang, G., Li, X., and Wang, Z. (2009). Apd2: the updated antimicrobial peptide database and its application in peptide design. *Nucleic Acids Res*, 37(Database issue):D933–7.
- Wang, J., Hu, C., Wu, Y., Stuart, A., Amemiya, C., Berriman, M., Toyoda, A., Hattori, M., and Aksoy, S. (2008b). Characterization of the antimicrobial peptide attacin loci from *Glossina morsitans*. *Insect Mol Biol*, 17(3):293–302.
- Wang, Z. and Wang, G. (2004). : the antimicrobial peptide database. *Nucleic Acids Res*, 32(Database issue):D590–2.
- Wasserman, W. W. and Fickett, J. W. (1998). Identification of regulatory regions which confer muscle-specific gene expression. *J Mol Biol*, 278(1):167–81.
- Weston, J., Leslie, C., Ie, E., Zhou, D., Elisseff, A., and Noble, W. S. (2005). Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–7.

- Whitmore, L., Chugh, J. K., Snook, C. F., and Wallace, B. A. (2003). The peptaibol database: a sequence and structure resource. *J Pept Sci*, 9(11-12):663–5.
- Wilkins, M. R., Lindskog, I., Gasteiger, E., Bairoch, A., Sanchez, J. C., Hochstrasser, D. F., and Appel, R. D. (1997). Detailed peptide characterization using peptidemass—a world-wide-web-accessible tool. *Electrophoresis*, 18(3-4):403–8.
- Wu, Y., Wei, B., Liu, H., Li, T., and Rayner, S. (2011). Mirpara: a svm-based software tool for prediction of most probable microrna coding regions in genome scale sequences. *BMC Bioinformatics*, 12(1):107.
- Yang, Z. R. (2004). Biological applications of support vector machines. *Brief Bioinform*, 5(4):328–38.
- Yassine, H. and Osta, M. A. (2010). Anopheles gambiae innate immunity. *Cell Microbiol*, 12(1):1–9.
- Ye, J., Member, S., Li, Q., and Member, S. (2005). A two-stage linear discriminant analysis via qr-decomposition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27:929–941.
- Yeaman, M. R. and Yount, N. Y. (2003). Mechanisms of antimicrobial peptide action and resistance. *Pharmacol Rev*, 55(1):27–55.
- Yount, N. Y., Bayer, A. S., Xiong, Y. Q., and Yeaman, M. R. (2006). Advances in antimicrobial peptide immunobiology. *Biopolymers*, 84(5):435–58.
- Zasloff, M. (2002). Antimicrobial peptides of multicellular organisms. *Nature*, 415(6870):389–95.
- Zernov, V. V., Balakin, K. V., Ivaschenko, A. A., Savchuk, N. P., and Pletnev, I. V. (2003). Drug discovery using support vector machines. the case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. *J Chem Inf Comput Sci*, 43(6):2048–56.
- Zhang, H., Ling, C. X., and Zhao, Z. (2005). The learnability of naive bayes. In *In: Proceedings of Canadian Artificial Intelligence Conference*, pages 432–441. AAAI Press.
- Zhao, H. W., Zhou, D., and Haddad, G. G. (2011). Antimicrobial peptides increase tolerance to oxidant stress in drosophila melanogaster. *J Biol Chem*, 286(8):6211–8.
- Zheng, L., Li, X., Li, F., Yan, X., Wang, Y., and Wang, Z. (2011). Automatic classification of lip color based on svm in traditional chinese medicine inspection. *Image and signal processing (CISP), 2010 3rd International Congress*, 28(1):7–11.