Fingerhut, Legana C. H. W. (2022) ***Identifying antimicrobial peptides in genomes using machine learning.*** **PhD Thesis, James Cook University.**

Identifying antimicrobial peptides in genomes using machine learning

This thesis is submitted for the Degree of Doctor of Philosophy by:

Legana C. H. W. Fingerhut

Current degrees held:

Bachelor of Biomedical Sciences (Honours) Class I

Bachelor of Science in Ecology and Conservation and Zoology

Date: 13/05/2022

College of Public Health, Medical and Veterinary Sciences

Centre for Tropical Bioinformatics and Molecular Biology

James Cook University

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my primary supervisor, Dr Ira Cooke. Thank you for your support, kindness, patience, encouragement and enthusiasm. Thank you for being the best bioinformatics instructor one could ask for. I feel fortunate to have learnt so much from you, and I have no doubt the skills you taught me will benefit me in the rest of my career.

I would also like to thank the rest of my advisory team, Prof Jan Strugnell, Prof Norelle Daly and Prof David Miller.

Special thanks to Brooke, Jl and my peers in the codeR-TSV and marine omics groups for sharing your PhD journeys with mine and for your encouragement, support and fun you all provided.

Toda raba to Rabbi Ari Rubin, for helping me maintain my spiritual health and for reminding me for the big picture.

Thank you to my furbaby Safia, for never leaving my side, and for distracting me when I needed it (and sometimes when I did not) and for ensuring I interrupted long hours at my work desk every day to go for walkies.

A most heartfelt thank you to my husband Jared, for your continuous support, patience and understanding. Your love means more to me than words could ever express.

Finally, to everyone else that has been a part of this incredible journey:

Thank you.

STATEMENT OF CONTRIBUTIONS OF OTHERS

Assistance	Contribution	Name	Affiliation
Intellectual support	Project plan and development	Dr Ira Cooke	James Cook University
	Editorial support	Dr Ira Cooke Prof Jan Strugnell Prof Norelle Daly Prof David Miller	James Cook University James Cook University James Cook University James Cook University
	Statistical and computer programming support	Dr Ira Cooke	James Cook University
Financial support	Stipend	Postgraduate Research Scholarship Higher Degree by Research Covid-19 Student Support Scholarship	James Cook University James Cook University
	Write-up Grant	Doctoral Completion Grant	College of Public Health, Medical and Veterinary Sciences, James Cook University
		Tuition fee sponsorship	James Cook University

PUBLICATION STATUS FOR THESIS CHAPTERS

Thesis Chapter	Published	Planned to publish	Target journal	Author list plan
1	No	No	n/a	
2	Yes: Fingerhut, LCHW, Miller, DJ, Strugnell, JM, Daly, NL & Cooke, IR 2020, ‘ampir: an R package for fast genome-wide prediction of antimicrobial peptides’, *Bioinformatics*, vol. 36, no. 21, pp. 5262–5263, DOI: 10.1093/bioinformatics/btaa653			
3	No	Yes	Briefings in Bioinformatics	Fingerhut, LCHW, Miller, DJ, Strugnell, JM, Daly, NL & Cooke, IR
4	No	Yes		
5	No	No		

A publication has arisen that was based on chapter 2 of this thesis. I was the first author of this publication and contributed the majority of work involved in manuscript writing, software development and data analysis. Co-authors helped edit the manuscript text. My primary supervisor (Dr Ira Cooke) edited the manuscript text and contributed a small amount of code to the ampir package. I plan to adapt chapters 3 and 4 to submit for publication in the Briefings in Bioinformatics journal.

ABSTRACT

Introduction

Antimicrobial peptides (AMPs) are part of the innate immune system in animals and plants, and are produced by almost all life forms. They defend against pathogens such as bacteria, viruses and fungi, and are also thought to play a role in regulating the microbiome. Due to the broad-spectrum activity AMPs possess, and their low potential to induce antimicrobial resistance, AMPs are of great interest as new drug candidates. As the genomes of more and more organisms have been sequenced the potential to discover novel AMPs from predicted protein sequences in public databases has grown. To facilitate this discovery-process many machine learning based AMP prediction tools have been developed, however, many are not fit for the purpose of scanning realistic whole-genome datasets. This thesis investigated the problems and potential solutions associated with identifying AMPs on a genome-wide scale. It begins by developing a new machine learning based AMP predictor with the computational speed and programmable interface required for genome-wide scanning (chapter 2). In chapter 3 it investigates the datasets and summary statistics most appropriate for benchmarking machine learning based AMP predictors in a genome-wide scanning context. Finally it investigates the premise that machine-learning predictors are superior to another widespread approach (homology) and how the answer to this assumption depends on the taxon of interest (chapter 4).

Methods

This thesis used curated AMP databases and UniProt to generate the training and testing datasets. The machine learning based AMP predictor, ampir, developed in chapter 2 is based on the support vector machine (SVM) approach, and has been wrapped up in an R package. Chapter 3 evaluated the training and test sets of nine machine learning based AMP predictors and compared them to representative whole-genome datasets (proteomes) from a highly studied plant (*Arabidopsis thaliana)* and an animal (*Homo sapiens)*. Chapter 3 also assessed the performance of AMP predictors using these two proteomes as benchmark data. In chapter 4, BLAST was compared to a machine learning based AMP approach to predict AMPs in the proteomes of nine organisms spanning a wide taxonomic range. To assess the effect of taxonomic distance on the performance in both methods, a novel metric was formulated to measure the degree to which an organism is represented by AMPs from closely related organisms in an AMP database.

Results

In chapter 2, the machine learning based AMP predictor ampir was shown to outperform other AMP predictors, especially when evaluated using metrics most relevant for AMP discovery from a proteome and when using the proteomes of real organisms (*Arabidopsis thaliana* and *Homo sapiens*) as benchmarks. Chapter 3 highlighted that the training and test sets in the majority of current AMP predictors contain biases that limit their ability to predict AMPs based on realistic proteome input data. Chapter 4 revealed that out of the two AMP finding methods, machine learning AMP predictors and BLAST, only BLAST was significantly positively correlated with taxonomic distance. This suggests that (as expected) machine learning based AMP predictors are indeed better than homology-based searches to discover AMPs in taxonomically distant organisms.

Conclusions

The outcomes of this thesis improve the overall understanding of AMP finding methods using homology-based methods and machine learning via scrutinisation of the underlying data ans assumptions these methods employ. It demonstrates that this is a very challenging, and arguably unsolved problem, but that there is considerable scope for progress if methods for training and evaluating predictors pay close attention to proteome-wide scanning as an intended use case. In addition, it identifies gaps in current AMP datasets that should be filled to realise this goal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: General Introduction

1.1 Antimicrobial peptides

Antimicrobial peptides (AMPs), also known as “host defence peptides” are small proteins that inhibit or kill microbes. Their best known function is as part of the innate immune system where they provide protection against pathogens (Yeaman and Yount, 2003). The earliest studies on AMPs focussed on those with antibacterial activity (Boman, 1995), however, in addition to bacteria, AMPs have also been found that protect against viruses (Real *et al.*, 2004), fungi (Hancock, 1997) and even protists (Xiao *et al.*, 2013). This broad activity is likely due to the diversity of AMPs and their respective structures (Schmitt, Rosa and Destoumieux-Garzón, 2016). AMPs are often able to interact with multiple targets rather than a specific receptor due to their amphipathic structure and charge which interacts with the microbial membrane (Pasupuleti, Schmidtchen and Malmsten, 2012). The main interactions include the disruption of the microbial cell membrane or intracellular processes resulting in cell death (Moravej *et al.*, 2018). Due to the broad scale effectiveness of AMPs as antimicrobial agents and their low antimicrobial resistance potential these peptides are of great interest for antibiotic-like therapeutic drug development (Baltzer and Brown, 2011; Moretta *et al.*, 2021), and multiple AMPs have already been commercialised for this purpose (Chen and Lu, 2020).

AMPs are found throughout the tree of life. They are produced by animals (Zasloff, 2002), plants (Tam *et al.*, 2015), fungi (Essig *et al.*, 2014), protists, e.g. amoeboids (Andrä, Herbst and Leippe, 2003), archaea (Charlesworth and Burns, 2015) and bacteria (Riley and Wertz, 2002). Across this taxonomic spectrum AMPs are likely to have diverse roles. In higher organisms such as animals and plants they are likely to have dual roles defending against pathogens and (as explained below) in modulating the microbiome (Mergaert, 2018). AMPs produced by bacteria, fungi and archaea are likely to mediate competitive interactions between microbes (Hibbing *et al.*, 2010; Essig *et al.*, 2014; Besse *et al.*, 2015).

Over the last decade, interest in the immune system has shifted from the view of a single host organism defending against invading pathogens to a multi-organismal assemblage that is maintained through complex interactions between its members (Eberl, 2010; Hooper, Littman and Macpherson, 2012; McFall-Ngai *et al.*, 2013). This shift reflects the view that the immune system is adapted to mould the host and microbes to co-exist to form a superorganism (Gill *et al.*, 2006; Eberl, 2010), or holobiont (Margulis, 1993), which is a collective of microbes and the host. Under this new viewpoint, the immune system is a dynamic system which acts to maintain an equilibrium or homeostatic environment within the holobiont. Evidence is now emerging that AMPs are key components of the immune system that maintain this equilibrium by regulating the abundance and composition of microbes (Mergaert, 2018). A diverse repertoire of AMPs is needed to shape the microbiome (Salzman *et al.*, 2010; Bosch, 2013) and it is therefore likely that there are many AMPs left undiscovered.

Microbes are important in maintaining the health of most animals and plants. For example, in mice, which are used as models to study human health, microbial communities are linked to gut health where it has been shown that microbes potentially lower chronic gut inflammation linked to irritable bowel syndrome or Crohn’s disease (Salzman *et al.*, 2010). Moreover, in a comparison of gut microbial community structure between lean and obese mice, a greater abundance of beneficial microbes was found in lean mice that may affect body size by increasing metabolic efficiency (Ley *et al.*, 2005). In addition to gut health, microbes contribute to the health of the skin through production of AMPs which synergise with human AMPs to kill pathogens such as *Staphylococcus aureus* (Nakatsuji et al., 2017), *Streptococcus* (Cogen *et al.*, 2010) and *Escherichia coli* (Lüders *et al.*, 2003), reducing the risk of skin conditions such as atopic dermatitis (Nakatsuji et al., 2017). Host protection by microbes is optimised when the microbial community is diverse (Fraune *et al.*, 2015) and it is therefore important to understand how the AMP repertoire affects the microbial communities.

Several studies, using a range of taxa, have shown the regulatory effect of AMPs on the microbiome. Fraune *et al.* (2010) demonstrated that a single AMP (periculin1a) drastically changed the microbial abundance and community structure by altering its expression in *Hydra*. Furthermore, Franzenburg *et al.* (2013) discovered that the composition of arminins (a family of AMPs) is species specific in *Hydra* and that four *Hydra* species all acquired distinct microbial communities under identical culture conditions. This discovery, that AMP expression may be linked to control of the microbiome, has also been observed in other taxa. Expression of the α-defensin DEFA5 gene has been linked to the control of microbial gut communities in mice (Salzman *et al.*, 2010). Weevils, like many insects, contain vertically transmitted symbiotic bacteria, which reside in specialised host tissue, and that provide the host with nutrients (Anselme *et al.*, 2008). Through *in vitro* experiments Login *et al.* (2011) found that the weevil coleoptericin-A AMP keeps the bacteria within the specialised tissue by maintaining their growth. Similarly, legume plants contain specialised nodules that house symbiotic bacteria and nodule-specific cysteine rich (NCR) AMPs terminally differentiate the bacteria to maximise their nitrogen fixation ability (Van de Velde *et al.*, 2010). It is likely that the same AMPs can perform dual actions in relation to microbial control. As shown in weevils, low concentrations or expression of coleoptericin AMPs inhibited growth of symbiotic bacteria around the bacteria-containing host tissue whereas highly expressed AMPs in surface tissues kill invading pathogens (Login *et al.*, 2011). This suggests that the function of AMPs depends on their relative expression and location.

Identifying AMPs, and studying their biological roles is challenging because their short sequence length and rapid evolution obscures evolutionary relationships (Pearson, 2013). This means that homology between AMPs in different organisms, especially distantly related organisms, is often difficult to detect. This in turn means that classification of AMPs into families can be challenging. Nevertheless, analysis of large AMP repertoires has led to the identification of some broad classes of AMPs (e.g. based on cysteine arrangement) where evolutionary relationships can be inferred through short taxonomic distance and/or genomic arrangement. For example, small cysteine rich AMPs are divided into multiple classes based on their cysteine arrangement and are widespread in plants (Manners, 2007). Several plant genomes have been examined for small cysteine rich AMPs and a diverse range of candidates were found (Silverstein *et al.*, 2007). Furthermore, the genome of the human bacterial symbiont *Staphylococcus capitis* was assembled and analysed for antimicrobial activity and a series of proteins were discovered in four gene clusters and experimentally verified as AMPs (Kumar *et al.*, 2017a). Gene clusters that encode AMPs were similarly uncovered in genomes of *Streptococcus pneumoniae* which led to the subsequent discovery of novel AMPs (Javan *et al.*, 2018). Interestingly, these clusters were located in specific regions of the genome and Javan *et al.* (2018) indicated that the AMPs within the clusters are arranged by recombination events. This finding could have important implications for the evolution and regulation of AMPs and comparative genomics analysis of closely related species could show synteny between AMP genome regions. Indeed, a comparative genomics investigation of a specific AMP class (β-defensin) between cattle and sheep showed conserved gene arrangement on multiple chromosomes (Hall *et al.*, 2017). Furthermore, the sheep AMP cathelicidin gene family was mapped to chromosomes from cattle, humans and mice which were similarly conserved. In comparison to humans and mice, sheep and cattle contained more cathelicidin genes which indicates diversification in these taxa (Huttner *et al.*, 1998). Diversification of cathelicidin genes have similarly been found in marsupials and monotremes (Warren *et al.*, 2008). This suggests that despite conserved gene clusters, specific AMPs uniquely evolve in different taxa. These studies indicate that AMP evolution is related to the arrangement in the genome. However, in order to study AMP evolution on a broad scale, representations of all classes of AMPs should be included and the method should be generalisable (i.e. not restricted to specific taxa). By using a comparative genomics approach of all AMPs on closely related species, broad scale patterns of AMP evolution can be observed. This includes aspects such as: how AMPs evolve, e.g. tandem duplication or positive/negative selection; which type of AMPs have expanded at what evolutionary timescale; how closely related AMPs are to one another and at what rate they evolve; do different taxa have different types of AMPs; if AMPs are clustered in the genome, are they clustered so as to co-regulate. Answering these questions is ultimately dependent on the ability to identify complete AMP repertoires in the genomes of a range of taxa.

AMPs can be found in all living organisms. To discover which peptides are AMPs, *in vitro* or *in vivo* experiments can be designed. However, it would be unfeasible to sample all living organisms and perform bioassay guided fractionation to discover which peptides have antimicrobial activity. Therefore, *in silico* methods are preferred to perform initial scans for AMPs on a large-scale basis prior to experimental verification. Previous attempts have been made to identify AMPs purely on their sequence, however none are designed to scan whole genomes, and their performance in this context remains a bottleneck for novel AMP identification as well as studying the evolution of AMPs.

1.2 Finding AMPs

Around three thousand AMPs have so far been discovered among many more that are likely to exist across the tree of life. Since this thesis is concerned with the topic of discovering new AMPs a good starting point is to ask how the AMP discovery process currently works. To address this, I searched the literature for recent research papers (between 2018-2021) which reported novel experimentally verified AMP discoveries (see Table S1.1), focussing on the role that computational tools played in identifying candidates for experimental verification. This search revealed 29 papers and a survey of these papers revealed a wide range of workflows for AMP discovery (summarised in Figure 1.1), but found that a high proportion (18/29) of papers used some variant of what I will describe in this thesis as a “genome-scanning” or ‘omics-based approach. In its general form this approach uses the target genome and associated protein predictions (called a proteome) as the search space for novel AMPs. Candidate AMPs are then identified by a homology-based search, using the basic local alignment tool (BLAST) (Altschul *et al.*, 1990), machine learning prediction or some combination of the two.

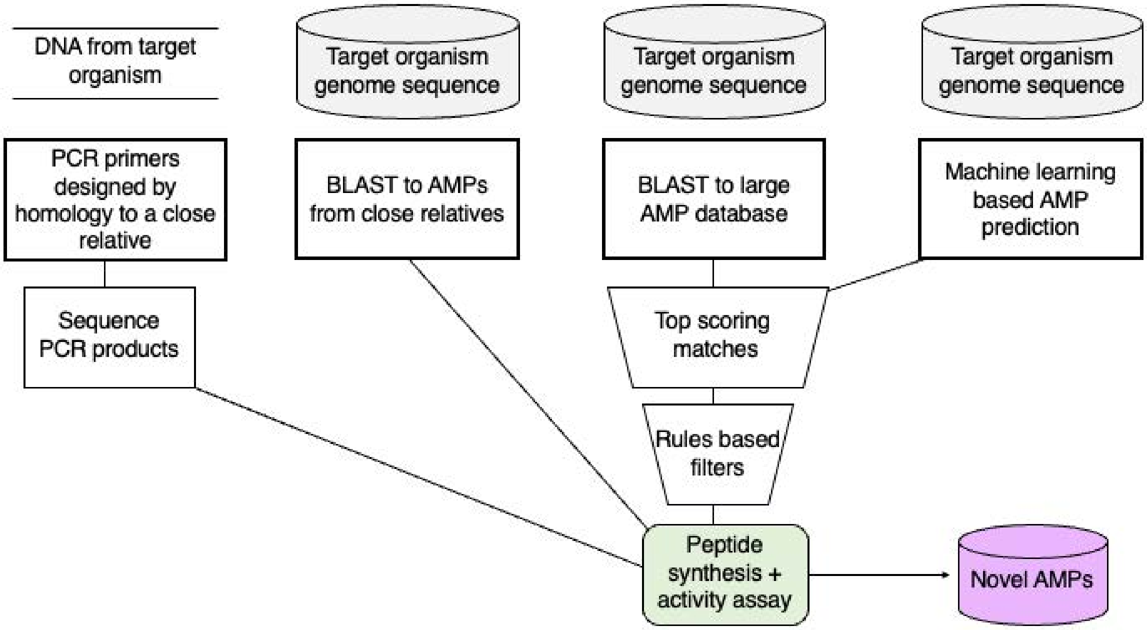

DNA from target organism
Target organism genome sequence
Target organism genome sequence
Target organism genome sequence
PCR primers designed by homology to a close relative
BLAST to AMPs from close relatives
BLAST to large AMP database
Machine learning based AMP prediction
Sequence PCR products
Top scoring matches
Rules based filters
Peptide synthesis + activity assay
Novel AMPs

Figure 1.1: Summary of workflows for AMP identification used by 29 recent publications reporting novel, experimentally verified AMPs.

Of the 29 papers surveyed, 18 identified AMPs based on very close homology to AMPs in a closely related organism (see Figure 1.1). This overall approach is not the focus of this thesis, however, as it is still widely used, I will summarise it briefly here. Two such workflows were identified. The first was a PCR amplification method based on a primer that generally targets a highly conserved region of known AMP sequences, e.g. a sequence within the signal peptide or 5’ untranslated regions. This method most commonly uses primers based on AMPs from extremely closely related species and appears to be most popular in frogs (Wu *et al.*, 2018; Li *et al.*, 2019; Gong *et al.*, 2020; Jiang *et al.*, 2020; Wang *et al.*, 2020a; Chai *et al.*, 2021) but has also been used in fishes *Salmo trutta* (Huang *et al.*, 2019) and *Bostrychus sinensis* (Shen *et al.*, 2021), the fox *Vulpes lagopus* (Li *et al.*, 2021b), the tick *Dermacentor silvarum* (Li *et al.*, 2021a) and scorpion *Chaerilus tricostatus* (He et al., 2021). Since PCR amplification can tolerate a small number of sequence mutations in the primer region this method is essentially a homology-based approach and is therefore analogous to BLAST (described below) but is restricted to only the very closest of homologs. Among the papers surveyed it typically resulted in amplification of a single novel AMP which is likely to be a direct ortholog of the AMP used for primer design. The advantage of the PCR-based approach is that it does not require the genome of the target organism to be known.

The second close-homology-based workflow, shown in Figure 1.1, involved the use of BLAST to scan the proteome of the target organism for AMPs in closely related species, at times even within the same AMP family. For example, a small number of novel cathelicidin AMPs were identified in the frog *Hoplobatrachus rugulosus* transcriptome (Chen et al., 2021) and in the goose *Anser cygnoides* genome (Xiao *et al.*, 2020) and koala *Phascolarctos cinereus* (Peel et al., 2021) using cathelicidin sequences of closely related organisms. A similar approach and result was obtained with β-defensin AMPs in the transcriptome of a fish, *Scophthalmus maximus* (Zhuang et al., 2021) and by using a single AMP sequence from a closely related species to find homologues in the genome of the polychaete *Capitella teleta* (Panteleev *et al.*, 2020). Like the PCR-based approach, these methods were able to provide novel AMP candidates with strong potential, however they have limited capability to identify truly novel AMPs, or AMPs from unrelated organisms.

Of greater relevance to this thesis are workflows in which the proteome of a target organism is scanned for AMPs without reference to a close taxonomic relative. Two basic strategies were employed to perform such scans: (1) homology via BLAST search against a large and diverse database of AMPs and (2) machine learning-based prediction. I found that one paper among the 29 surveyed adopted the first of these approaches (BLAST) (Hayashida and da Silva Junior, 2021), four used machine-learning alone (Yang *et al.*, 2018; González-García *et al.*, 2020; Hassan, Qutb and Dong, 2021; Onime *et al.*, 2021) and six combined both methods (Lee *et al.*, 2020a; Lee *et al.*, 2020b; Dong *et al.*, 2021; Lee *et al.*, 2021a; Lee *et al.*, 2021b; Liscano *et al.*, 2021). Papers that used a combination of homology and machine learning targeted a wide variety of organisms, including the frog *Boana pugnax* (Liscano *et al.*, 2021), fish (Dong *et al.*, 2021), beetle *Psacothea hilaris* (Lee et al., 2020a), mealworm *Zophobas atratus* (Lee et al., 2021a), and butterflies *Papilio xuthus* (Lee et al., 2020b) and

Porphyromonas gingivalis (Lee et al., 2021b*)*. In these papers the outcomes from homology (BLAST) and machine learning predictions were used to filter candidates (e.g. by requiring high scores in both methods). Studies which exclusively used machine learning tended to be applied to non-model organisms for which few homologous AMPs exist in known databases, e.g. in the mollusk *Pomacea poeyana* (González-García et al., 2020), and the shrimp *Litopenaeus vannamei* (Yang et al., 2018). In order to obtain a small list of candidates for experimental verification, candidate sequences were filtered either by using a very high decision threshold (González-García et al., 2020), or by using additional properties of the sequences such as the presence of α-helical structures (Yang et al., 2018).

For the purposes of this thesis, the most important finding from this survey of AMP discovery workflows is that the majority of AMP finding studies start with a transcriptome or genome, and use computational tools (either BLAST or machine learning methods) to obtain a short list of candidates for experimental testing. A key characteristic of this approach is that it involves searching for a relatively small number of AMPs within a very large search space. This has important implications for the design and validation of AMP prediction methods which I discuss at length in chapter 3 of this thesis. Another important discovery from this survey was that simple homology-based methods such as BLAST remain in common use as the primary method for computational AMP prediction, or as an auxiliary method to machine learning-based methods. The relative efficacy of these methods in relation to taxonomic representation in AMP databases forms the basis for chapter 4.

1.2.1 Machine learning models to predict AMPs

The number of published machine learning models designed to predict AMPs has rapidly expanded during the past decade, with much of that growth occurring during the time that research was undertaken for this thesis (2018-2022). This includes many models with accompanying software that are currently available for use (see Table 1.1), as well as many others that have been designed, but which did not include standalone software (Torrent *et al.*, 2011; Fernandes, Rigden and Franco, 2012; Khosravian *et al.*, 2013; Ng, Rosdi and Shahrudin, 2015; Pane *et al.*, 2017; Veltri, Kamath and Shehu, 2017; Wang *et al.*, 2017; Liu *et al.*, 2018; Yoshida *et al.*, 2018; Zhang *et al.*, 2021b). Some of these AMP predictors contain specialisations for antiviral (Thakur, Qureshi and Kumar, 2012; Qureshi, Tandon and Kumar, 2015) or antibacterial peptide (Lata, Sharma and Raghava, 2007; Lata, Mishra and Raghava, 2010) detection, or specific AMP class types, e.g. cysteine rich AMPs (Porto, Pires and Franco, 2012), or linear cationic AMPs (Vishnepolsky and Pirtskhalava, 2014; Vishnepolsky *et al.*, 2018).

Most AMP predictors are primarily accessible via a web interface, where the user can paste in a select number of protein sequences in FASTA format and obtain a result which indicates the probability of the sequence being likely to be an AMP. While this practice has been encouraged on the basis that it results in ease of use (Chou, 2011), it generates a long-term server maintenance burden and often leads developers to impose hard limits on the number of sequences that can be analysed. At the time of writing this thesis, three AMP predictor web servers (Fjell, Hancock and Cherkasov, 2007; Wang *et al.*, 2011; Vishnepolsky and Pirtskhalava, 2014) were unavailable due to server problems, and others imposed sequence limits (<1000 sequences). In the past, when genome-scanning approaches to AMP discovery were uncommon, this limit would not be an issue. However, due to advances in technology, sequencing is becoming cheaper and faster and whole genome assemblies are released monthly (Yin *et al.*, 2017). This increase of biological data is expected to continue, leading to a requirement for large scale analysis on high-performance computing platforms (HPC) which are typically accessible via a command-line interface (Yin *et al.*, 2017). The need for such a high throughput tool was the primary motivation behind development of the AMP machine learning predictor ampir, which forms the basis for chapter 2 of this thesis. Although several such tools are now available, at the time that work for this thesis commenced (2018), very few command-line tools were available for general AMP prediction. Those tools that were available were limited in other ways such as their choice of training data, or a required access to proprietary libraries, e.g. AmPEP (Bhadra *et al.*, 2018). The AMP predictor, ampir, was developed as a free, open source, and customisable AMP predictor to address these issues.

Table 1.1: Predictors currently available for use and their respective statistical learning algorithms.

Predictor name	**Statistical learning algorithm**	**Availability**	**Reference**
AntiBP	ANN, QM, SVM	Web server	Lata, Sharma and Raghava (2007)
AMPer	HMM models	Web server	Fjell, Hancock and Cherkasov (2007)
CAMP	DA, RF, SVM	Web server	Thomas *et al.* (2010)
AntiBP2	SVM	Web server	Lata, Mishra and Raghava (2010)
AVPpred	SVM	Web server	Thakur, Qureshi and Kumar (2012)
CS-AMPPred	SVM	PERL / Linux machines	Porto, Pires and Franco (2012)
ClassAMP	RF, SVM	Web server	Joseph *et al.* (2012)
iAMP-2L	FKNN	Web server	Xiao *et al.* (2013)
AVP-IC_{50}Pred	SVM, RF, IBk, KStar	Web server	Qureshi, Tandon and Kumar (2015)
iAMPpred	SVM	Web server	Meher *et al.* (2017)
DBAASP	DBSCAN	Web server	Vishnepolsky *et al.* (2018)
AMP Scanner	DNN	Web server	Veltri, Kamath and Shehu (2018)

AmPEP	RF	MATLAB	Bhadra *et al.* (2018)
dbAMP	RF	Web server	Jhong *et al.* (2019)
ACEP	DNN	Python	Fu *et al.* (2020)
amPEPpy	RF	Python / Web server	Lawrence *et al.* (2020)
deep-amPEP30	CNN	Web server	Yan *et al.* (2020)
AmpGram	RF	R / Web server	Burdukiewicz *et al.* (2020)
MACREL	RF	Python / Web server	Santos-Júnior *et al.* (2020)
AMPlify	ADL	Python	Li *et al.* (2020)
IAMPE	XGBoost, SVM RF, KNN	Web server	Kavousi *et al.* (2020)
AniAMPpred	SVM	Web server	Sharma *et al.* (2021)

ANN: artificial neural network, ADL: attentive deep learning, CNN: convolutional neural network, DA: discriminant analysis, DNN: deep neural network, DBSCAN: Density-Based Spatial Clustering, FKNN: fuzzy K-nearest neighbour, HMM models: hidden Markov models, IBk and KStar: instance based learner, KNN: K-nearest neighbour, QM: quantitative matrices, RF: random forest, SVM: support vector machine, XGBoost: eXtreme Gradient Boosting.

1.3 Theoretical concepts used in this thesis

Throughout this thesis I use the term ‘AMP prediction’ to refer to the task of classifying a list of amino acid sequences into AMPs and non-AMPs. More specifically, my focus is on the use of supervised statistical learning or deep learning (collectively known as machine learning) approaches to accomplish this. This approach requires a reference database that includes both positive cases (AMP sequences) and negative cases (non-AMP sequences). To convert these data into a form suitable for input to widely used algorithms (see below) each of the amino acid sequences is described via a collection of ‘features’. These features are summary statistics that describe the sequence, and in the case of AMPs they typically include physicochemical properties such as charge and amphiphilicity. The model is then trained on a large subset (typically two thirds) of these data, allowing it to learn (i.e. fit parameters) to distinguish between the two classes (positive and negative). Finally, the leftover smaller subset of data, typically around one third of the total (Kohavi, 1995), is used to test the model to determine the training performance of the model (see Figure 1.2).

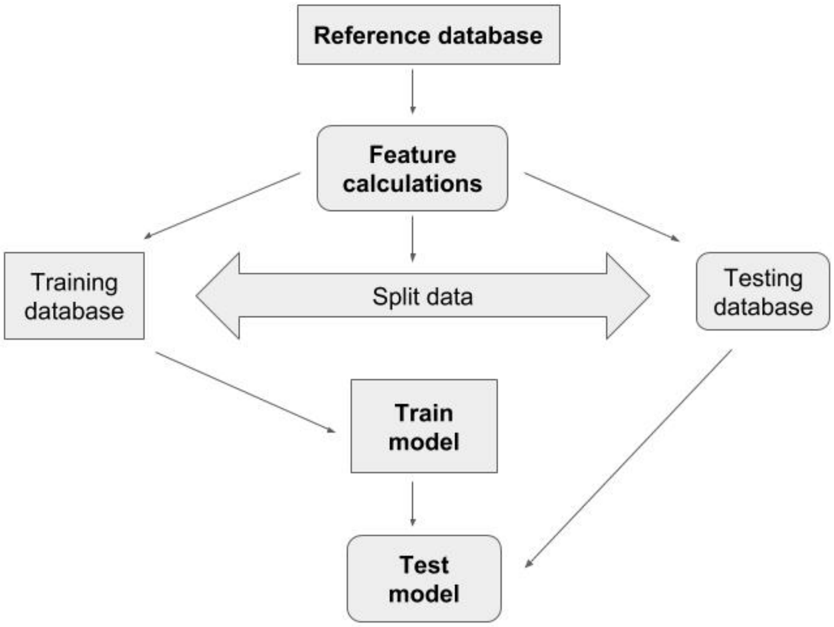


Figure 1.2: Simplified flowchart describing the process of building and testing a statistical classifier for antimicrobial peptide (AMP) prediction using a supervised learning approach. The reference database consists of verified AMPs and general proteins which are used to calculate features from (e.g. physicochemical properties). The calculated features from all proteins are split into a training database to train the

model (using a classification algorithm), and a testing database to evaluate the trained model’s performance.

In the sections below I examine the background theory related to each of the steps in the workflow shown in Figure 1.2, emphasising issues of relevance to AMP prediction. Issues relating to the construction of the reference database are very important and are covered in detail in chapters 2 and 3.

1.3.1 Reference database generation for AMP prediction

Positive (verified AMPs) and negative (non-AMPs) datasets are required to generate the reference database, which is used to train and test the AMP predictor. Since AMPs have been studied for many decades, there are a large number (~3k) of sequences with experimentally verified activity available for use as positive training data. The majority of these are available on the public online annotated protein sequence database Swiss-Prot (Bairoch and Apweiler, 2000). However, there are several protein databases which focus purely on AMP sequences, both natural and synthetic peptides. These AMP databases are often used for positive training sets in AMP predictors (Liu *et al.*, 2017). The five largest AMP databases are: APD (the Antimicrobial Peptide Database (Wang and Wang, 2004), CAMP (Collection of Antimicrobial Peptides) (Waghu *et al.*, 2016), DRAMP (Data Repository of Antimicrobial Peptides) (Fan *et al.*, 2016), LAMP (A database Linking Antimicrobial Peptides) (Zhao *et al.*, 2013) and dbAMP (Jhong *et al.*, 2019). Despite the usefulness of concentrated AMP databases, it is not clear if all databases are regularly updated with AMP sequences. In addition, the AMP databases overlap with one another and collectively contain high overlap with Swiss-Prot (85% - accessed June 2018).

The negative background dataset of AMP predictors is commonly obtained from Swiss-Prot, as there is no database which strictly contains non-AMPs (Liu *et al.*, 2017). Using a diverse range of proteins as a negative dataset will better train the model for realistic situations. This is especially important when scanning genomes, as there are a wide range of proteins present which the predictor would have to assess. However, most previous methods refined the negative dataset by employing “key words” to select contrasting proteins to AMPs. For example, not “antimicrobial”, “secretory”, or “membranous” (Meher *et al.*, 2017; Bhadra *et al.*, 2018; Veltri, Kamath and Shehu, 2018; Jhong *et al.*, 2019). Since this filtering affects both the training and testing dataset it is likely to result in inflated measures of performance because proteins that are difficult to classify (e.g. non-AMP secreted proteins) have been removed. While the negative impacts of filtering secreted proteins are now well known there are many other more subtle ways in which test and training data can accumulate biases compared with realistic input data. These are explored in more detail in chapter 3 of this thesis.

1.3.2 Features used to describe amino acid sequences

The feature calculation and selection steps are crucial elements of any machine learning process involving amino acid sequences. Although the sequences themselves capture all of the information that is available to train the model, they are not in a form that is suitable for use with most statistical approaches. To address this, each sequence is reduced to a numerical vector called a feature vector, ideally consisting of properties of the sequence that distinguish between the AMP and non-AMP classes (Bhadra *et al.*, 2018). This feature vector may be large during the initial model development stage but is often reduced to eliminate features with little predictive value via a feature selection step. Feature selection is an integral part of machine learning as the feature set substantially influences the performance of the model (Chen *et al.*, 2020). The type of features used could both increase and decrease the accuracy of the model as features need to be relevant so as not to add arbitrary information to the model (Hawkins, 2004). In addition, a greater number of features can decrease the computational speed of the model. Therefore, both the type of features and the number of features should be assessed to determine the best fit for the data.

In the context of AMP prediction, features are chosen to reflect compositional and structural properties that reflect AMP activity and can subsequently be used to identify them (Meher *et al.*, 2017). Physicochemical properties influence the structure, function and posttranslational modification process of proteins and are commonly used as feature types in computational protein prediction (Cai and Jiang, 2016). For example, physicochemical properties of the sequence such as hydrophobicity (approximately 50%) and net charge (positive around physiological pH) are often conserved (Yeaman and Yount, 2003) and can indicate the biological action of AMPs (Rončević, Puizina and Tossi, 2019). The combination of these physicochemical characteristics allow AMPs to focus on a wide variety of targets (Leptihn *et al.*, 2010). To facilitate membrane interaction, AMPs form amphipathic α-helix structures which include both a hydrophobic and hydrophilic section (Gallo and Huttner, 1998). This regional structure allows the peptide to bind or penetrate the membrane and can be quantified by the hydrophobic moment (Eisenberg, Weiss and Terwilliger, 1982). The hydrophobic moment has been shown to be useful to distinguish AMPs from non-AMPs (Vishnepolsky and Pirtskhalava, 2014). These physicochemical properties, in addition to isoelectric point, protein aggregation and propensity, have been modelled by Torrent *et al.* (2011) and were found to effectively predict antimicrobial action.

In addition to physicochemical properties, compositional properties such as the amino acid composition and pseudo amino acid composition have been previously included for AMP prediction (Lata, Sharma and Raghava, 2007; Lata, Mishra and Raghava, 2010; Joseph *et al.*, 2012; Thakur, Qureshi and Kumar, 2012; Qureshi, Tandon and Kumar, 2015; Meher *et al.*, 2017; Pane *et al.*, 2017). The amino acid composition of antibacterial peptides and non-antibacterial peptides were shown to be different in antibacterial peptides from non-antibacterial peptides which included differential residue preference (Lata, Sharma and Raghava, 2007; Lata, Mishra and Raghava, 2010). Residue preference and differential prediction of residues was also found in AMPs (Meher *et al.*, 2017). Generally, the amino acid composition calculates the frequency of occurrence of each of the 20 standard amino acids relative to a given protein sequence. However, the sequence order information is not included in the amino acid composition which can restrict models when predicting protein attributes (Chou, 2009). Therefore the pseudo amino acid composition was created by Chou (2001) which improves upon the standard amino acid composition by including a number of correlated factors that estimate the sequence order effect (Chou, 2001). The pseudo amino acid composition has been suggested to be used instead of the amino acid composition in AMP prediction because it loses less sequence information and therefore increases AMP prediction performance (Khosravian *et al.*, 2013; Xiao *et al.*, 2013; Zare *et al.*, 2015) and has been used multiple times to predict AMPs (Wang *et al.*, 2011; Khosravian *et al.*, 2013; Xiao *et al.*, 2013; Zare *et al.*, 2015; Lin and Xu, 2016; Meher *et al.*, 2017).

1.3.3 Model algorithms

The most common classifiers used in AMP prediction are the support vector machine (SVM) and random forest (RF) (see Table 1.1). SVM is a binary classifier that can learn to distinguish between two classes. It does this by mapping feature data points to a multidimensional space and inserting a hyperplane in between the feature datapoints to separate the two classes. The separating hyperplane is placed in a way to maximise the space between the two classes, which results in an optimal classification performance when the model is used on a test set (see Figure 1.3). Further performance improvements can often be achieved through the use of a kernel function, e.g. a radial kernel (Noble, 2006).

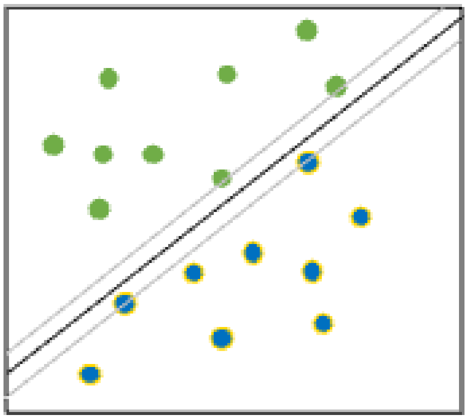


Figure 1.3: A simplified diagram showing the two different classes (indicated by the light and dark coloured points) separated by a hyperplane (dark coloured line), which in two dimensions is represented by a line. The light coloured lines adjacent to the hyperplane represent the width of the space in between the two classes.

RF is a decision tree ensemble classification method that averages multiple random feature subsets to create a classification model (Breiman, 2001). Decision trees use features to separate classes (see Figure 1.4) and RF uses a large number of uncorrelated decision trees to maximise the class prediction result. RFs appeared to be increasing in popularity in AMP prediction as at least four AMP predictors have implemented this classifier within the last two years (Bhadra *et al.*, 2018; Jhong *et al.*, 2019; Santos-Júnior *et al.*, 2020; Burdukiewicz *et al.*, 2020).

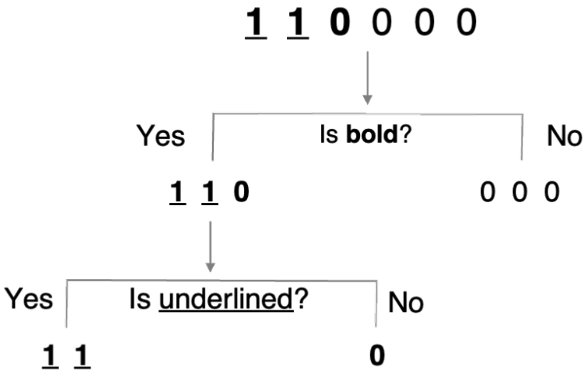


Figure 1.4: Simplified diagram of decision tree logic to separate data based on features.

1.3.4 Model training and optimisation

A crucial step to improve accuracy in AMP predictor models is model optimisation. Model optimisation occurs in the model training stage. During the model training it is customary to try different models with varying model parameters to find the model that best fits the data. This is generally performed using resampling methods which evaluate different model tuning parameters on the performance of the model on different subsets of the training set. One common resampling method is k-fold cross validation where the training data is reshuffled into k-folds of equal size. The typical number of folds (k) recommended is 10, as smaller folds can exhibit bias and larger folds increase the variance in the performance estimation (Kohavi, 1995). In this fold set, one fold is allocated for testing and the remaining nine for training. A model is then constructed on the nine training folds and evaluated on the testing fold. This process is repeated k times, until all folds have been allocated as testing fold once. Finally, the evaluations for each model performance on the test folds are averaged which will provide an overall estimate of model performance (see Figure 1.5). This estimate is considered to be a more accurate measure of the performance of the model compared to an evaluation metric from a model that was only fit once, as it maximises the information present in the data.

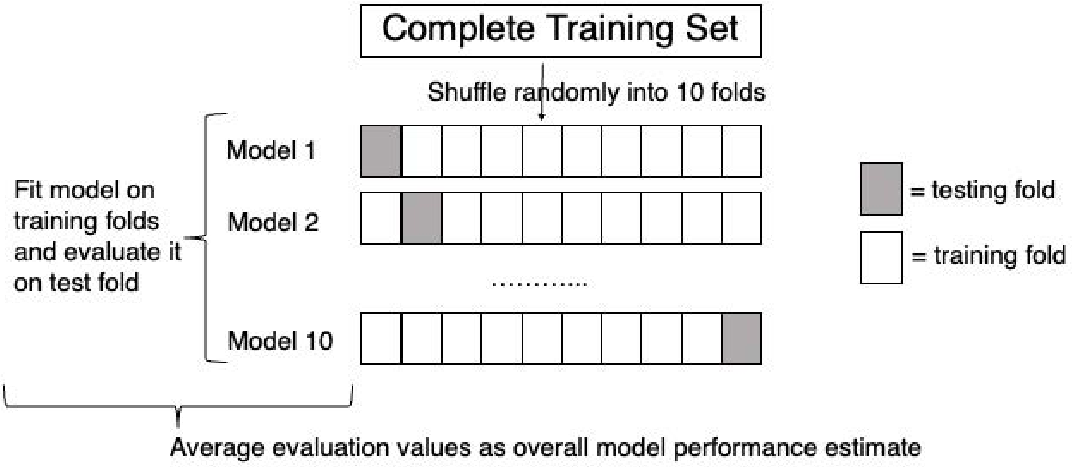

Complete Training Set
Shuffle randomly into 10 folds
Model 1
Fit model on training folds and evaluate it on test fold
Model 2
....
Model 10
= testing fold
= training fold
Average evaluation values as overall model performance estimate

Figure 1.5: k-fold cross validation as a resampling method to increase the accuracy of a model.

The goal of resampling methods is to ultimately improve predictive performance of the model by avoiding overfitting while making maximum use of the available training data. If a model is only able to have exceptionally high performance on the sample data it is trained on, but not when faced with new data, the model is likely to be over-fit, and will consequently have poor performance when used on a different sample dataset. Resampling methods are commonly used in conjunction with the tuning of model parameters. These tuning parameters, which are also referred to as hyperparameters, affect the learning ability of the model and are set prior to the training process of the model. Therefore, the values of these hyperparameters do not change during the model training. Choosing the optimal values for the hyperparameters of the model for the training data used is referred to as tuning the model. Resampling methods are useful as they can iterate through a selection of hyperparameter values and reveal how these values affect the performance of the model. Once the iteration process is complete, the optimal hyperparameter set can be determined by selecting the hyperparameter values that are associated with the best performing model. Finally, the full training data can then be used with the obtained optimal hyperparameter set to train a final model with maximal considered performance (see Figure 1.6).

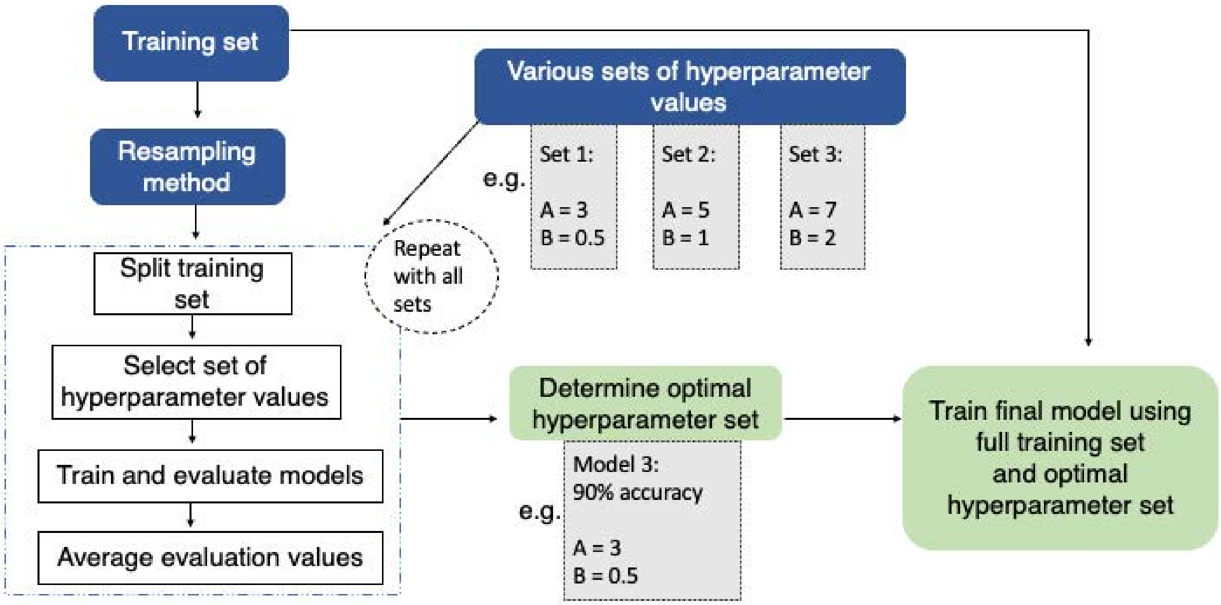
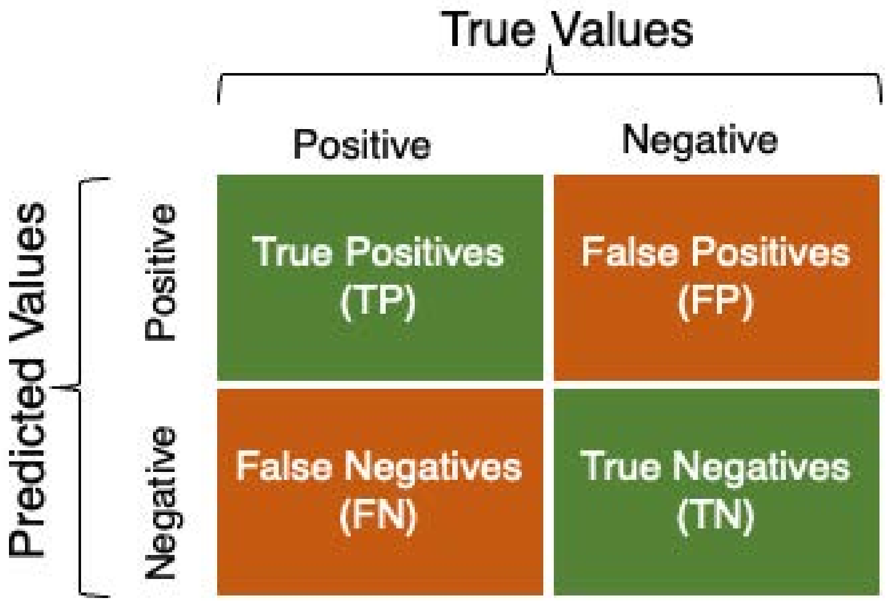

Training set
Resampling method
Various sets of hyperparameter values
e.g.
Set 1:
A = 3
B = 0.5
Set 2:
A = 5
B = 1
Set 3:
A = 7
B = 2
Repeat with all sets
Split training set
Select set of hyperparameter values
Train and evaluate models
Average evaluation values
Determine optimal hyperparameter set
e.g.
Model 3:
90% accuracy
A = 3
B = 0.5
Train final model using full training set and optimal hyperparameter set

Figure 1.6: Tuning the hyperparameters of a model with a resampling method.

1.3.5 Model testing and evaluation metrics

The trained classification model is generally tested on a hold-out test set to determine its performance on an independent dataset. The fundamental basis of model performance evaluation for binary classification models such as AMP predictors is the confusion matrix which captures the four possible outcomes when comparing the prediction results to the actual class values. These four result categories are summarised in Figure 1.7.



		True Values	
		Positive	Negative
Predicted Values	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

Figure 1.7: Confusion matrix showing the predicted class values made by the machine learning model compared to the true class values resulting in true positives, false positives, false negatives or true negatives.

In the context of AMP prediction, the positives refer to the AMPs and the negatives refer to the non-AMPs. Common performance metrics calculated from the confusion matrix are: accuracy, specificity (the true negative rate), recall or sensitivity (the true positive rate), the false positive rate, precision (the positive predictive value), and the F1 score (the balanced measure between precision and recall) (see equation 1.1).

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$False\ \ Positive\ \ Rate = \frac{FP}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F1\ \ score = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

Equation 1.1: Common performance metrics calculated from the confusion matrix.

The true positive rate and the false positive rate can be used to generate a receiver operating characteristics (ROC) curve. ROC curves are able to visualise the performance of machine learning models and subsequently are frequently used by AMP predictors to demonstrate the performance of their models compared to other AMP predictors (Meher *et al.*, 2017; Veltri, Kamath and Shehu, 2018; Fu *et al.*, 2020; Li *et al.*, 2020; Sharma *et al.*, 2021; Xu *et al.*, 2021). A ROC curve depicts a trade-off between the true and false positive rate over a predicted probability range between 0 and 1. This curve can only be generated if machine learning models predict class values across a continuous range of decision thresholds (i.e. between 0 and 1). If a model only predicts in a discrete capacity (generally at a probability threshold of 0.5), only a single point would be visible on the ROC curve (Fawcett, 2006). A discrete model with perfect prediction performance would consist of a point at the position (0,1) (see Figure 1.8). High performing models that predict in a continuous threshold range generally have a curve that angles towards the top left of the ROC plot. In contrast, models with random performance, i.e. those that have learnt no information from the training data, have a diagonal line that spans the middle of the plot (see Figure 1.8).

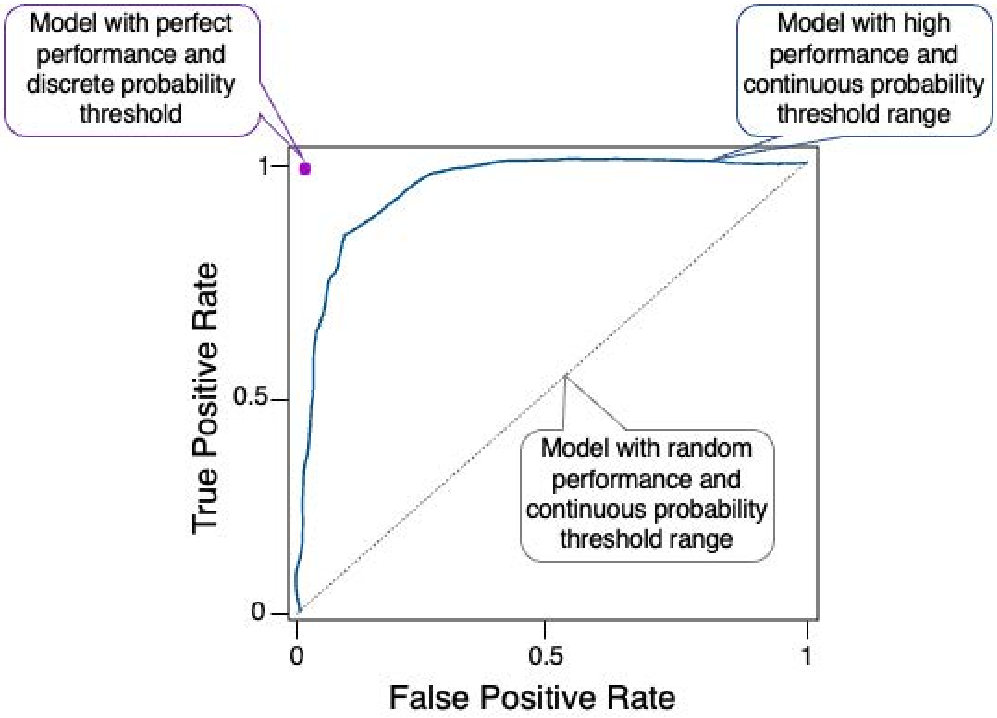


Figure 1.8: Example receiver operator characteristic (ROC) curve for three machine learning models.

1.4 Purpose of the thesis

The general assumption in the development of machine learning AMP predictors is that the sample datasets used for training and testing are unbiased and that standard machine learning model development is sufficient to accurately predict AMPs. This means that it is assumed that the data present in the training and testing sets accurately reflect the data present in input datasets that would be encountered in real usage. A major challenge, especially in the context of genomics is addressing the bias that arises due to mismatch between such realistic input data and available training data (Whalen *et al.*, 2022). Choices made by AMP predictor developers to include or exclude specific data can affect this bias. This might mean that AMP prediction needs to be approached with these biases in mind in order to achieve genuine improvements in the utility of AMP predictors as tools for biologists. The purpose of this thesis is to evaluate the sources of these biases and their potential impact on the performance of AMP predictors. Furthermore, this thesis strives to address these biases in the development and evaluation of an AMP predictor. Whilst acknowledging this is difficult to achieve, this thesis shows that these biases are too important to be ignored.

1.5 Aims

The overall aim of this thesis was to identify AMPs in genomes using machine learning methods and to address potential biases in this process. Figure 1.9 depicts an overview of the thesis. Specific aims were:

Aim 1: Develop a machine learning AMP predictor framework and software, ampir, suitable for genome-wide scanning. This aim is addressed in chapter 2.

Aim 2: Using the AMP prediction software developed in chapter 2, as well other available AMP predictors, to benchmark their performance on real proteomes. This aim is addressed in chapter 3

Aim 3: Benchmark machine learning methods (using the AMP prediction framework developed in chapter 2) and homology-based searches (using BLAST) on proteomes of a wide range of organisms, to determine the effect of taxonomic distance on AMP finding methods. This aim is addressed in chapter 4.

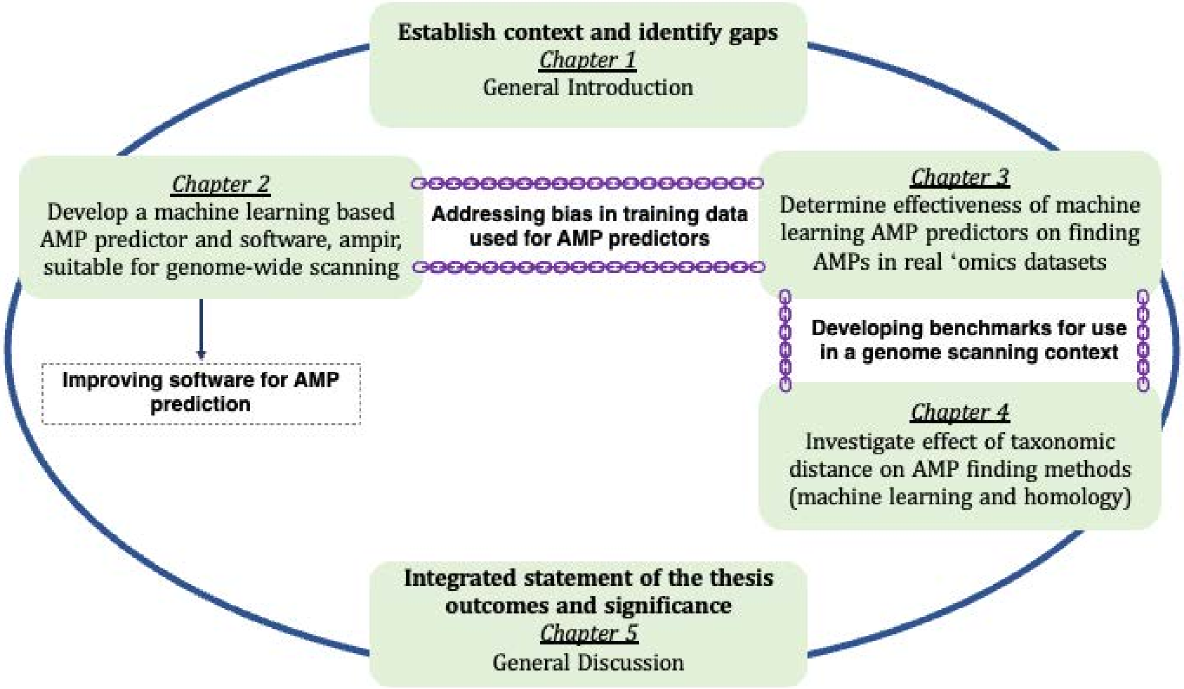

Establish context and identify gaps
Chapter 1
General Introduction
Chapter 2
Develop a machine learning based AMP predictor and software, ampir, suitable for genome-wide scanning
Addressing bias in training data used for AMP predictors
Chapter 3
Determine effectiveness of machine learning AMP predictors on finding AMPs in real 'omics datasets
Improving software for AMP prediction
Developing benchmarks for use in a genome scanning context
Chapter 4
Investigate effect of taxonomic distance on AMP finding methods (machine learning and homology)
Integrated statement of the thesis outcomes and significance
Chapter 5
General Discussion

Figure 1.9: Conceptual thesis figure showing the general thesis structure, chapter objectives and relationships between chapters.

Chapter 2: ampir: an R package for fast genome-wide prediction of antimicrobial peptides

2.1 Abstract

Summary: Antimicrobial peptides (AMPs) are key components of the innate immune system that protect against pathogens, regulate the microbiome, and are promising targets for pharmaceutical research. Computational tools based on machine learning have the potential to aid discovery of genes encoding novel AMPs but existing approaches are not designed for genome-wide scans. To facilitate such genome-wide discovery of AMPs I developed a fast and accurate AMP classification framework, ampir. ampir is designed for high-throughput, integrates well with existing bioinformatics pipelines, and has much higher classification accuracy than existing methods when applied to whole genome data.

Availability and Implementation: ampir is implemented primarily in R with core feature calculation methods written in C++. Release versions are available via CRAN and work on all major operating systems. The development version is maintained at https://github.com/legana/ampir. ampir is also available via a Shiny based web server https://ampir.marine-omics.net/ where users can submit protein sequences in FASTA file format to be classified by either the “precursor” or “mature” model. The prediction results can then be downloaded as a csv file. The full details of the model development process can be accessed on https://github.com/legana/AMP_pub.

2.2 Introduction

Antimicrobial peptides (AMPs) are effector molecules of the innate immune system. They are produced by most forms of life to combat microbial pathogens including bacteria, viruses, fungi, and protists. Their potent activity has led to strong interest in these molecules as targets for pharmaceutical research. Although originally known for their role in defending the host against pathogens (Zasloff, 2002), there is now increasing interest in the regulatory abilities of AMPs on the microbiome (Franzenburg *et al.*, 2013; Bosch, 2014; Mergaert, 2018). Changes in expression of specific AMPs have been linked to changes in microbial composition and abundance during development in basal metazoans (Fraune *et al.*, 2010), nutrient uptake in plants (Van de Velde *et al.*, 2010) and gut health in mammals (Wehkamp *et al.*, 2005). Taken together these studies show that AMPs are sometimes key regulators of the microbiome and as such they may have co-evolved with host microbial partners (Thaiss *et al.*, 2016). However, understanding these co-evolutionary relationships is challenging because the interactions between AMPs and the microbiome are likely to be complex and may involve a range of AMPs across many species.

Studies in which AMP prediction software is applied to the entire complement of protein coding sequences in a genome have the potential to uncover the entire repertoire of AMPs in an organism. Such genome-wide studies might help reveal co-evolutionary patterns between host and microbiome and potentially correlate the diversity of AMPs to the diversity of the microbiome. Previous genome-wide analysis of specific AMPs and AMP families have already revealed multiple selection patterns (Zhang *et al.*, 2019) and gene order conservation (Hall *et al.*, 2017). In addition, genome-wide AMP prediction could aid medical research by revealing novel AMPs useful for potential therapeutics (Kim *et al.*, 2017). In fact, the advantages of using genomes for AMP discovery is already recognised and the genomes of a range of taxa including butterflies (Wang *et al.*, 2021), corals (Shinzato *et al.*, 2021), fish (Zhang *et al.*, 2022; Zhang *et al.*, 2021a), snakes (Kim *et al.*, 2017) and bats (Pérez de la Lastra *et al.*, 2021) have been used to identify AMPs.

Despite intense interest in AMPs, the genes that encode them remain difficult to detect. They evolve rapidly, driven by positive selection coupled with high rates of gene gain and loss (Hanson, Lemaitre and Unckless, 2019) and this, combined with the small size of mature AMP peptides (10-50 amino acids), makes them difficult to detect through homology-based approaches alone. A promising alternative approach to AMP detection is the use of supervised machine learning based on physicochemical properties. Many AMP predictors have been developed using this approach (see chapter 1, Table 1.1)

When searching for an AMP classifier that works well with genome-wide data, important problems with current approaches were noticed. For example in (Meher *et al.*, 2017; Bhadra *et al.*, 2018; Veltri, Kamath and Shehu, 2018; Jhong *et al.*, 2019), the classification models are trained and tested with a negative dataset (i.e. non-AMPs) that is prepared by filtering out sequences that resemble sequences in the positive dataset (i.e. AMPs). The use of contrasting positive and negative datasets makes the prediction model easier and may lead to higher reported performance metrics, however, a model trained with filtered data is unlikely to perform well on a contrasting dataset that is not filtered, i.e. a realistic dataset that contains proteins that are similar to AMPs, such as secreted proteins (see Figure 2.1). This subsequently presents users with an undesirable trade-off. Users either need to pre-filter their data to match the model's training set, in which case they will be removing many valid AMPs, since many are secreted. Alternatively, users need to analyse their data unfiltered, in which case model prediction performance will be poor.

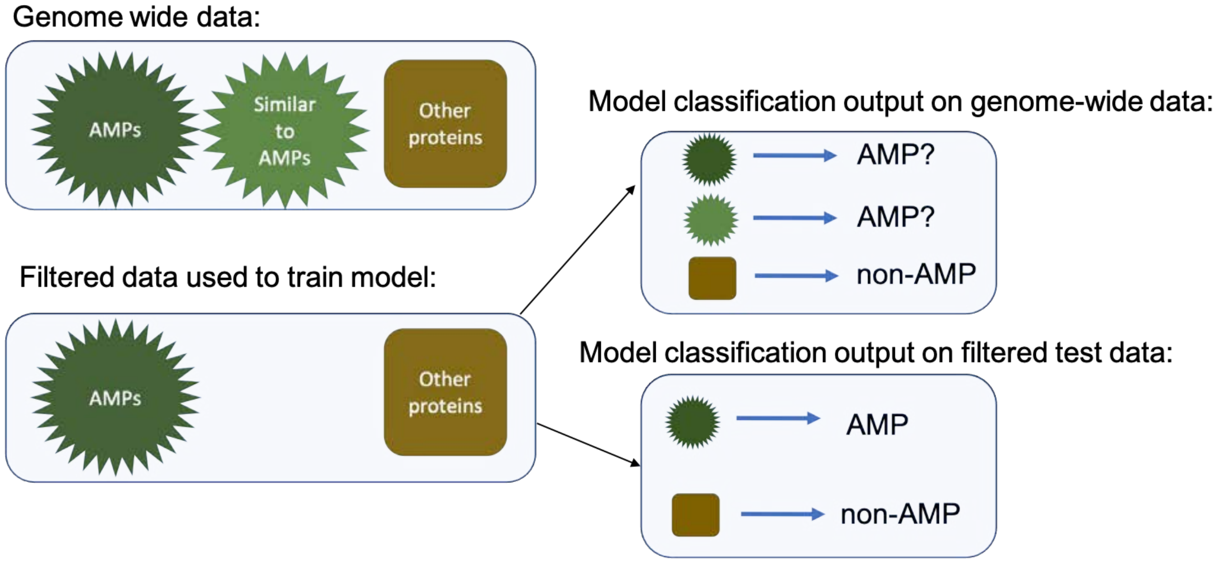

Genome wide data:
AMPs
Similar to AMPs
Other proteins
Model classification output on genome-wide data:
AMP?
AMP?
non-AMP
Filtered data used to train model:
AMPs
Other proteins
Model classification output on filtered test data:
AMP
non-AMP

Figure 2.1: A diagram indicating that models trained with filtered data (where the proteins that are similar to antimicrobial peptides (AMPs) are removed) perform well

when tested on other filtered data, but are likely to perform worse when used on non-filtered genome-wide data.

A second issue is that most predictors are trained and tested on a large proportion of mature peptides (see section 2.3.1.1) whereas in a genome-scanning context it is much more likely that researchers will be working with full-length precursor protein sequences. Furthermore, the proportion of AMPs in a genome is small (usually less than 1%) (see chapter 3) but test datasets used by existing predictors are either balanced (50% AMPs) or nearly balanced leading to unrealistically low estimates of the false positive rate.

Finally, most current AMP predictors have been designed with an emphasis on ease of use for novice users rather than high-throughput or efficient use by experts. Many have therefore been made available exclusively as web services (Xiao *et al.*, 2013; Meher *et al.*, 2017; Veltri, Kamath and Shehu, 2018), which means that computational speed depends on the external server configuration and load, and limits must be placed on the number of sequences that can be processed in one batch (though in some cases quite high limits are set e.g. (Veltri, Kamath and Shehu, 2018). More importantly, most AMP predictors lack an application programming interface (API) and are therefore difficult to integrate into bioinformatic pipelines which is an essential requirement for comparative genomics.

The focus of this research was not to create a fundamentally new approach for AMP prediction, but to optimise existing methods for whole genome input and implement these in a software package designed to satisfy the needs of a genome scan. The R package ampir (antimicrobial peptide prediction in R) was created, written primarily in the R (R Core Team, 2021) programming language with C++ integration for speed. ampir is available on all major operating systems and can be installed via the Comprehensive R Archive Network (CRAN).

2.3 ampir’s design

Most AMP predictors implement a supervised machine learning approach to evaluate the antimicrobial probability of a given protein, and ampir is no exception. This approach involves various choices such as the selection of training data, features and model algorithm. In addition, the final model should ideally be easily and efficiently utilised by anyone searching for AMPs in their datasets. Therefore, the computational implementation, distribution and user interface needs to be considered. The following sections cover these choices made, which led to the development of ampir.

2.3.1 Training data

2.3.1.1 Training data sources

A key goal of ampir was to optimise model predictions for a situation where the input would consist of full-length predicted proteins for an organism. For most non-model organisms these gene models (obtained by gene modelling on software such as Augustus) are all that is available. A survey of training data used in existing predictors (outlined below) revealed that many included a large proportion of mature peptides. Mature peptides represent protein products after enzymatic cleavage of the full-length precursor protein (see Figure 2.2). As such their sequences are difficult to infer from genomic data alone and in most genome scanning contexts it can be assumed that they would be unknown. In the section below the prevalence of likely precursor versus mature proteins in protein databases frequently used as a training data source is investigated. This information is later used to inform the choice of training data for ampir.

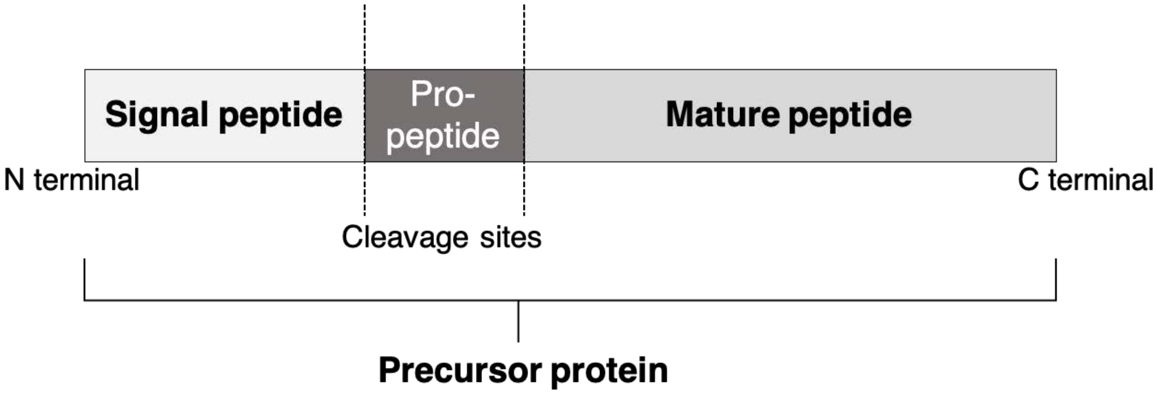


Figure 2.2: Simplified diagram of the constituents of a precursor protein. The constituents are not drawn to scale.

Positive and negative datasets to train the models are commonly obtained from online protein databases. UniProt is a large general protein database, which includes AMPs, is very well annotated and contains a lot of metadata about the sequences it contains. UniProt is divided into two separate database sections, Swiss-Prot and TrEMBL, based on their annotation level. Swiss-Prot contains high grade, non-redundant, manually curated reviewed proteins. TrEMBL is based on computationally analysed information reinforced with automatic annotation using tools such as Interpro (Mitchell *et al.*, 2019) which classify protein sequences into their respective families and predict functional domains (UniProt Consortium, 2019). Validated AMPs can be found within the Swiss-Prot database using the search term “keyword:Antimicrobial [KW-0929]”. In addition to the general Swiss-Prot database, there are also specialised AMP databases. AMP databases are manually curated online databases that specifically contain protein sequences with antimicrobial activity, largely sourced from the literature as well as from UniProt and the National Center for Biotechnology (NCBI) (NCBI Resource Coordinators, 2018). These AMP databases are often used for positive training sets in AMP predictors. The five largest AMP databases are: APD (the Antimicrobial Peptide Database (Wang and Wang, 2004), CAMP (Collection of Antimicrobial Peptides) (Waghu *et al.*, 2016), dbAMP (Jhong *et al.*, 2019), DRAMP (Data Repository of Antimicrobial Peptides) (Kang *et al.*, 2019) and LAMP (A database Linking Antimicrobial Peptides) (Zhao *et al.*, 2013). These AMP databases generally contain a high degree of manual curation for the sequences they contain. Therefore the sequences in these AMP databases are likely to be true AMPs. However, there is a lot of overlap in sequences between the AMP databases (Liu *et al.*, 2017) and Swiss-Prot. The three most recently updated AMP databases, APD, DRAMP and dbAMP, and only the naturally occurring AMPs in those databases, i.e. non-synthetic, as well as the UniProt database were used in this study (accessed April 2020, see Table 2.1).

Table 2.1: The number of antimicrobial peptides present in four protein databases.

Protein database	**No. of AMPs**
APD	3,177
DRAMP	4,394
dbAMP	4,213
UniProt*	3,221 (Reviewed) 19,288 (Unreviewed)

* AMPs in UniProt were found using the search term “keyword:Antimicrobial [KW-0929]”.

The APD was first published in 2004 (Wang and Wang, 2004) and has been regularly maintained. It is a well-known AMP database and has been used for reviews of the AMP prediction landscape (Wang, Li and Wang, 2016). The APD originally contained sequences that were smaller than 100 amino acids but this was changed to 200 amino acids in the 2016 update to encompass more AMPs (Wang, Li and Wang, 2016).

DRAMP was first published in 2016 (Fan *et al.*, 2016) and updated to DRAMP 2.0 in 2019 (Kang *et al.*, 2019). DRAMP focuses on mature peptides and the subsequent criteria for data collection include the removal of sequences if they contain precursor or signal regions or are larger than 100 amino acids (Fan *et al.*, 2016).

dbAMP was published in 2018 and contains a collection of AMPs obtained from a large number of AMP databases (including CAMP and LAMP) and protein databases like UniProt and NCBI (Jhong *et al.*, 2019). No sequence length restriction was evident in the criteria of their AMPs. However, as dbAMP is built on other databases that do impose sequence length restrictions, it is likely biased toward short proteins.

The sequence length restriction becomes apparent in Figure 2.3. It is clear that APD and DRAMP focus on mature peptides as they primarily contain sequences that are short (mostly < 50 amino acids). dbAMP and Swiss-Prot similarly contain these mature peptides but they also contain a range of precursor proteins. Swiss-Prot also includes a small number of larger proteins (> 500 amino acids that are listed under the keyword ‘Antimicrobial’ but are very different from classical AMPs. These include some large viral proteins (e.g. EXLYS_BPDPK, a 2,237 amino acid long peptidoglycan hydrolase that degrades cell components during virus entry) which show evidence of antibacterial activity but their mode of action and vast difference in size make them outliers from the point of view of building a machine learning model. Interestingly, two distinct peaks were noticeable in the sequence lengths of AMPs in the Swiss-Prot database. These peaks likely comprise mature peptides (peak to the left of the dashed vertical line, between the sequence length 0 and 50) and precursor proteins (peak to the right of the dashed vertical line, between 50 and 100).

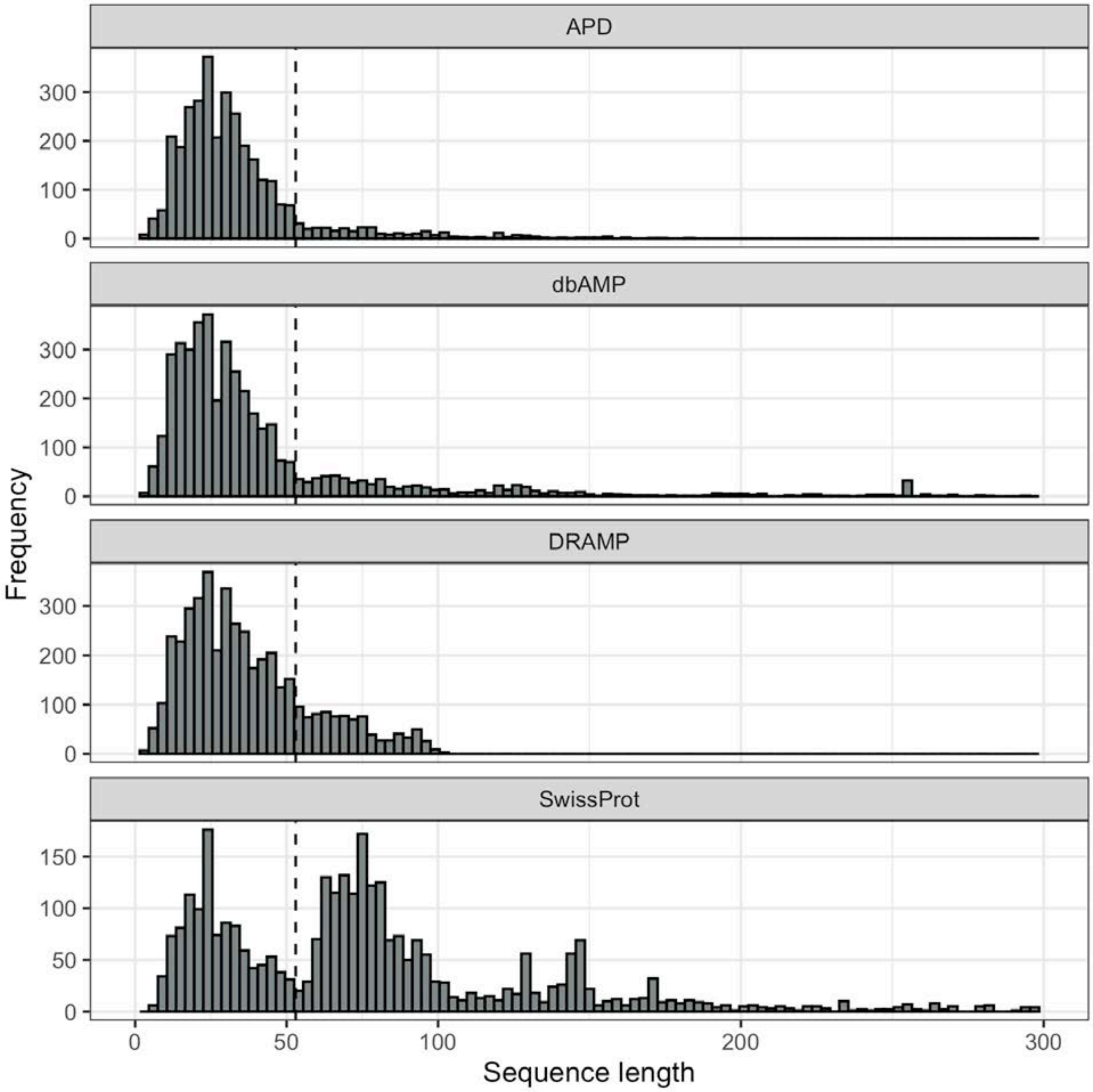


Figure 2.3: Sequence length distributions of AMPs in four protein databases. The vertical dashed line indicates the potential threshold of mature versus precursor proteins.

The Swiss-Prot database provides a peptide field that allows distinguishing between entries for mature peptides and precursors. This in turn can be used to check my hypothesis that dual peaks in the length distribution arise due to mature and precursor proteins respectively. If the peptide length is the same as the total length, it is a mature peptide. Proteins were considered to be precursor proteins if they were not mature peptides, for entries where the peptide information was available. If no peptide information was available, proteins were classified as Unknown. For the reviewed AMPs in Swiss-Prot, there are a total of 768 mature peptides, 806 precursors with peptide annotation information, and 1647 reviewed AMPs without peptide information. The Unknown category has a notably broader distribution of lengths reflecting the possibility that it includes a mix of both types (see Figure 2.4). The mature and precursor curve shapes of Figures 2.4 and 2.5 appear highly similar, indicating that these peaks are likely to comprise mature and precursor proteins, respectively.

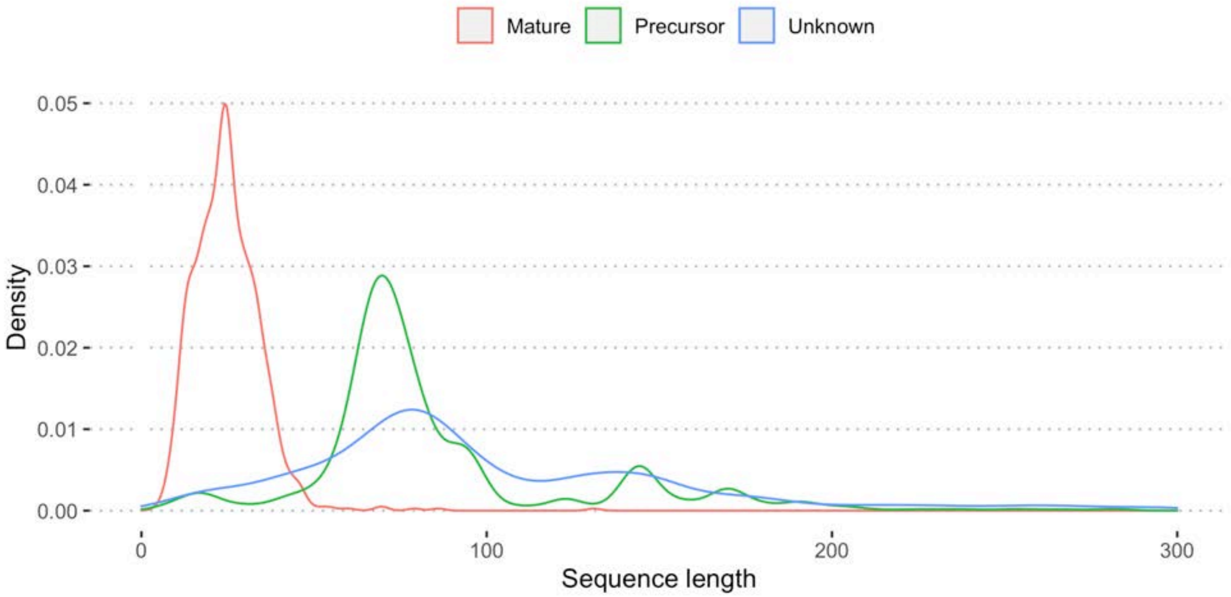


Figure 2.4: The sequence length distribution of mature AMPs and precursor AMPs in Swiss-Prot. Unknown refers to the AMPs that lacked peptide information.

Another likely indicator that a protein is an AMP precursor is the presence of a signal peptide. From the 806 precursors identified, 706 show well defined signal peptide sequences. Figure 2.5 shows that only precursors longer than about 60 amino acids are likely to have a signal peptide. Manual inspection of precursors without signal peptides revealed that many are annotated with a pro-peptide indicating that even in this group there is some post-translational processing to produce a mature product.

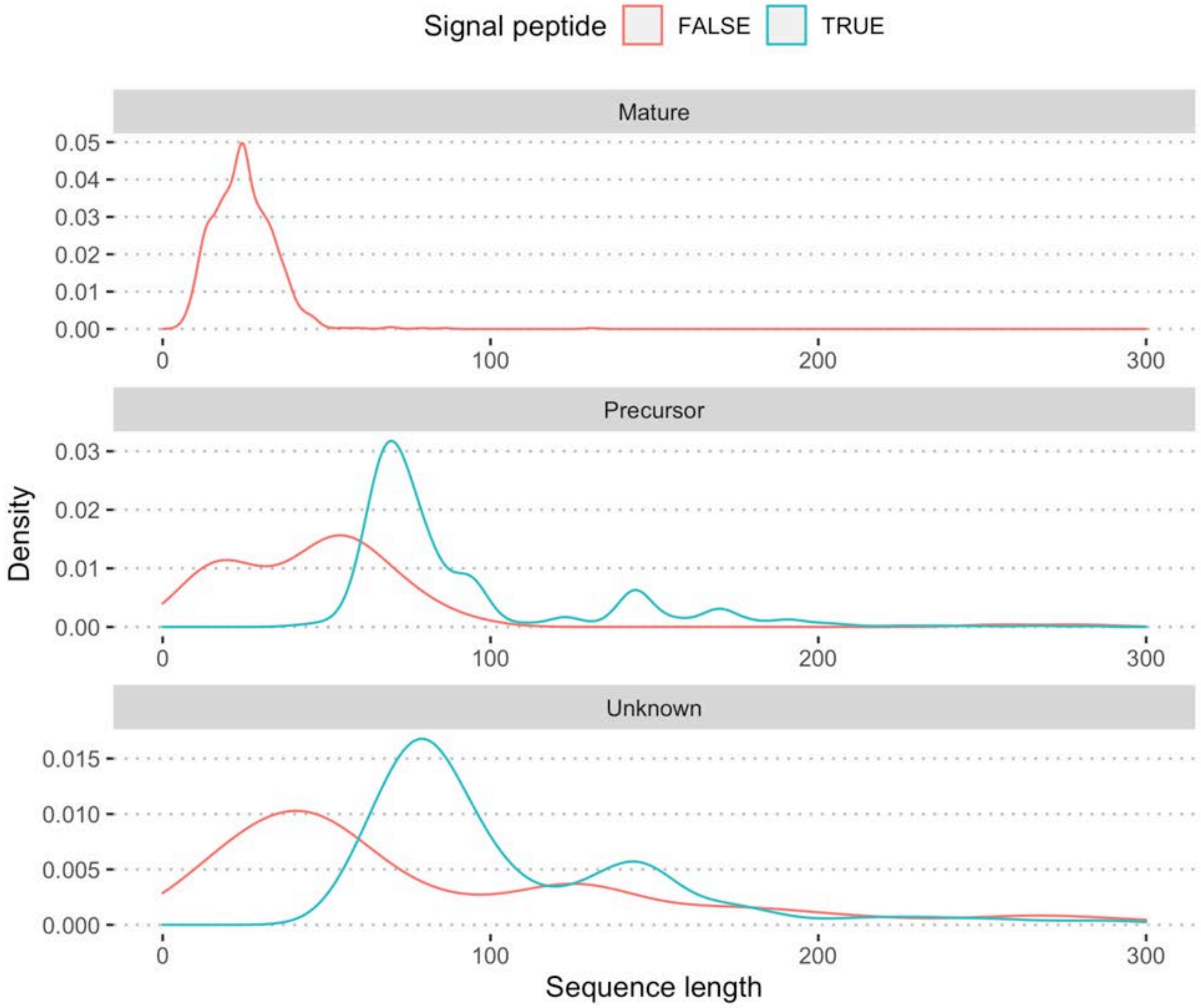


Figure 2.5: Sequence length distribution of mature AMPs and secreted/non secreted precursor AMPs in Swiss-Prot. Unknown refers to the AMPs that were not annotated as mature or precursor.

From the survey of AMP databases, it can be concluded that sequence length (mature peptides tend to be much shorter compared to precursor proteins) and the presence of a signal peptide (present at the beginning of a protein and indicative that the protein is secreted) are likely to be good proxies for whether an amino acid sequence is a mature peptide or precursor protein. Based on the length distributions, it appears that from all of the AMP databases, Swiss-Prot includes the largest number of precursor proteins. However, some are most likely present in other databases, especially in DRAMP. Furthermore, a sequence length of around 50 amino acids is likely effective for distinguishing most precursor proteins from mature peptides. This length cut-off can subsequently be used to guide the selection criteria for inclusion of sequences in ampir’s training data.

2.3.1.2 Training datasets for ampir’s models

As shown in the section 2.3.1.1, protein databases used for AMP prediction can be divided into precursor proteins and mature peptides. Therefore it was decided to create two models for ampir, one for precursor proteins and one for mature peptides. These models are anticipated to serve different needs in the community of potential users. The mature model can be used by peptide chemists wanting to check mature sequences. Whereas the precursor model would be more useful in genome-wide scanning contexts, such as when using transcriptomes or proteomes as input data.

As the two different models serve different purposes, the positive (AMPs) and negative (non-AMPs) datasets that constitute the training datasets varied in composition. This section will describe the positive and negative dataset selection for both the ampir precursor and mature peptide models.

Since the goal for ampir’s precursor model was to obtain the maximum possible utility or genome-wide scans, the training dataset consisted entirely of precursor proteins (sequences longer than 50 amino acids). In typical genome-scanning operations this is the only information available. To achieve the positive dataset, AMPs from UniProt (listed under the “Antimicrobial” keyword), both reviewed (Swiss-Prot) and unreviewed (TrEMBL) sequences were downloaded. In addition, naturally occurring AMPs from the AMP databases APD, DRAMP and dbAMP were obtained. However, only the sequences which were also present in UniProt were retained. Although this removes a small number of proteins from custom AMP databases, remaining protein sequences then include extensive metadata that is provided for proteins in UniProt. The remaining unreviewed sequences (which were not present in the AMP databases) were removed. Furthermore, sequences were removed if they were: longer than 500 amino acids, contained nonstandard amino acids, or duplicated. This resulted in a database containing 2,061 sequences of which 61 were unreviewed. Finally, the program CD-HIT (Li, Jaroszewski and Godzik, 2001) was used to cluster sequences to 90% sequence identity, keeping only a single representative sequence for each cluster. This groups sequences that are more than 90% similar together, while keeping only a single representative sequence (the longest one) for each cluster or group. This subsequently removes highly similar sequences and reduces redundancy. This resulted in 1,483 AMPs used as the final positive dataset for the ampir precursor model. The final positive dataset contained 535 organisms. Out of those organisms, *Arabidopsis thaliana*, mouse, human and rat contained the majority of annotated AMPs (more than 50, see Table 2.2).

Table 2.2: Summary table of the ampir precursor model positive dataset showing organisms that had more than 10 antimicrobial peptide clusters (n90) out of their respective total number of antimicrobial peptides (n).

Organism	**n90**	**n**
Arabidopsis thaliana (Mouse-ear cress)	282	289
Mus musculus (Mouse)	77	96
*Homo sapien*s (Human)	61	84
Rattus norvegicus (Rat)	52	59
Bos taurus (Bovine)	34	43
Gallus gallus (Chicken)	19	23
Sus scrofa (Pig)	18	27
Drosophila melanogaster (Fruit fly)	17	22
Escherichia coli (Bacteria)	12	13
Pan troglodytes (Chimpanzee)	12	32
Dictyostelium discoideum (Slime mould)	11	12
Ornithorhynchus anatinus (Duckbill platypus)	11	11
Caenorhabditis elegans (Nematode)	10	11

Macaca mulatta (Rhesus macaque)	10	23
Oryctolagus cuniculus (Rabbit)	10	13

The ampir precursor negative dataset started out using all proteins in the Swiss-Prot database, clustered to 90% identity with CD-HIT as the foundation for a background dataset. The goal was to use fairly minimal filtering on these so that the sequences have roughly the same composition as a typical set of non-AMP proteins in a genome. Sequences were removed if they were: present in the positive dataset, contained non-standard amino acids or were shorter than 50 amino acids or longer than 500 amino acids in length. The remaining background dataset was still very large (>300,000 sequences). A large negative dataset likely provides a better representation of genome diversity and a more accurate coverage of the feature space, which subsequently could improve the learning of the model. However, using all available non-AMP sequences would be computationally expensive and increase the size of the model. Therefore, the background dataset was randomly sampled to obtain a subset of sequences so that the AMP:non-AMP ratio was 1:10. This resulted in 14,830 sequences used as the negative dataset for the ampir precursor model. This use of an imbalanced dataset allows more data to be used to train the model. However, unless the imbalance is accounted for it will lead to an inaccurately trained model as the model will favour the data more abundantly represented and therefore will perform poorly on the minority class (AMPs). One way to circumvent this is by balancing the dataset via synthesis of additional data using an approach such as Synthetic Minority Over-sampling Technique (SMOTE) (Chawla *et al.*, 2002) or by randomly removing cases from the overrepresented set (He and Garcia, 2009). However, as mentioned earlier, the increase of data in a model increases the computational expense. An alternative approach to imbalanced data is to make it more expensive to misclassify a positive case during training (Weiss, 2004). This approach was implemented in ampir's precursor model via the weights parameter in the caret R package (Kuhn, 2008). During the training of the ampir precursor model weights were set to be inversely proportional to the number of sequences in each class. This weighted approach allows much more data to be used for training without causing the model to be biased towards the majority class.

A mature AMP prediction model was added to provide users the opportunity to analyse their mature peptide sequences. In general these analyses are low throughput use-cases as they rely on users knowing the mature sequence through mass spectrometry or related techniques. The positive dataset for the mature model contained AMP sequences from the APD, DRAMP, dbAMP and Swiss-Prot protein databases that were between 10 and 60 amino acids long. Identical to the precursor dataset, sequences that contained non-standard amino acids or were duplicated were removed. This resulted in 4,983 sequences. Like with the precursor dataset, the resulting sequences were clustered to 90% with CD-HIT. The final positive dataset for the ampir mature peptide model contained 3,232 AMP sequences.

The ideal negative dataset for a mature AMP model would be non-AMP peptides taken from the Swiss-Prot Peptide field. However, unfortunately there are very few such peptides verified and the available peptides tend to be neuropeptides and toxins. It is better for negative datasets to contain a wide range of proteins, to simulate a more realistic dataset. Therefore, a length filter was applied to extract non-AMPs sequences from the Swiss-Prot dataset that were 10 to 40 amino acids long. This 10-40 length filter is more stringent than the 10-60 length filter used in the positive dataset because the positive dataset consisted exclusively of mature peptides. However, the negative dataset contains both mature and precursor proteins and it was important to exclude any potential short precursor proteins that may be present in the dataset. Identical to the positive dataset, sequences that contained nonstandard amino acids were removed and the remaining dataset was clustered to 90% identity with CD-HIT. The final negative dataset for the ampir mature model contained 3,321 sequences.

2.3.2 Feature selection

ampir uses a suite of features commonly used in AMP prediction such as physicochemical properties (Meher *et al.*, 2017; Bhadra *et al.*, 2018) and Chou’s pseudo amino acid composition (Meher *et al.*, 2017). Initial tests revealed that calculating features from amino acid sequences was by far the most computationally intensive step when running predictions from a trained model. Therefore, to eliminate this bottleneck the pseudo amino acid composition calculation from the protr R package (Xiao *et al.*, 2015) was rewritten in C++. Physicochemical properties were calculated with the Peptides R package (Osorio, Rondon-Villarreal and Torres, 2015). Recursive feature elimination in the caret R package was performed to select a minimal feature set for ampir that would still accurately predict AMPs and avoid overfitting. This feature set included the pseudo amino acid composition, isoelectric point, net charge, hydrophobicity, molecular weight and hydrophobic moment.

2.3.2.1 Visualising individual features

As an indication of features that are likely to be useful for classification, the distribution of features was plotted for both the background and target dataset. In these plots, features that show a clear separation in distribution between positive and negative cases are likely to be most useful for classification. In addition, it should be noted that these feature distributions are sometimes heavily influenced by background filtering. This is at least partly because my large protein cut-off (500 amino acids) removes a small number of very large proteins that cause skewness in the Mw and Charge distributions. Higher order lambda values from the pseudo amino acid composition seem to show little difference between the positive and negative datasets (Figure 2.6) (this is not true of low order values though). Xc1 and Xc2 are output values from the pseudo amino acid composition where Xc1 values refer to specific amino acids and Xc2 values refer to lambda values. The lambda value is a parameter in the pseudo amino acid composition function which is specified by the user. For both precursor and mature peptide models I used all physicochemical predictors, all Xc1 predictors and the first two Xc2 predictors.

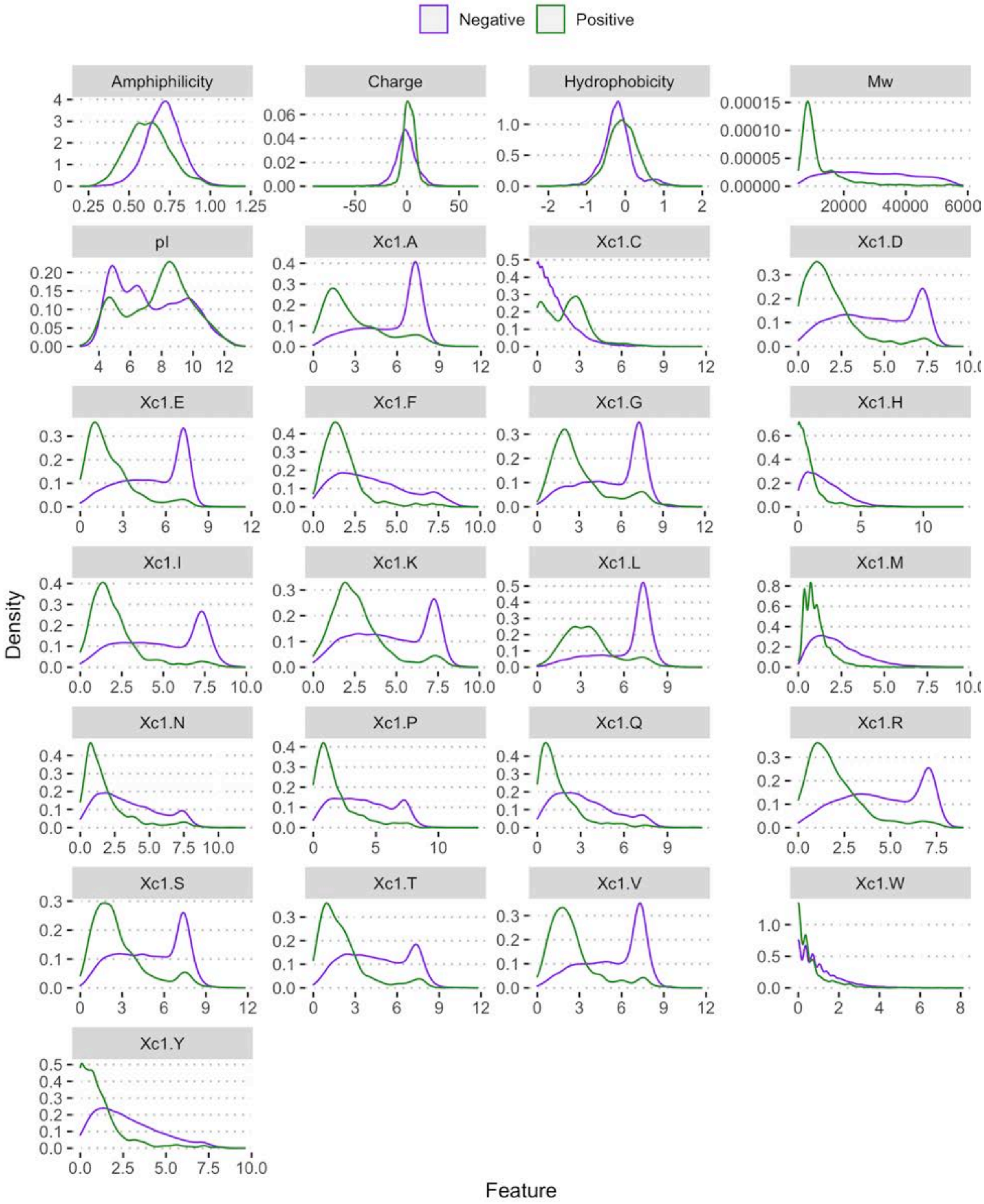


Figure 2.6: The feature distribution for the precursor training data of the hydrophobic moment (amphiphilicity, net charge, hydrophobicity, molecular weight (Mw), isoelectric

point (pI) and pseudo amino acid composition in the negative (background) and positive (target) dataset.

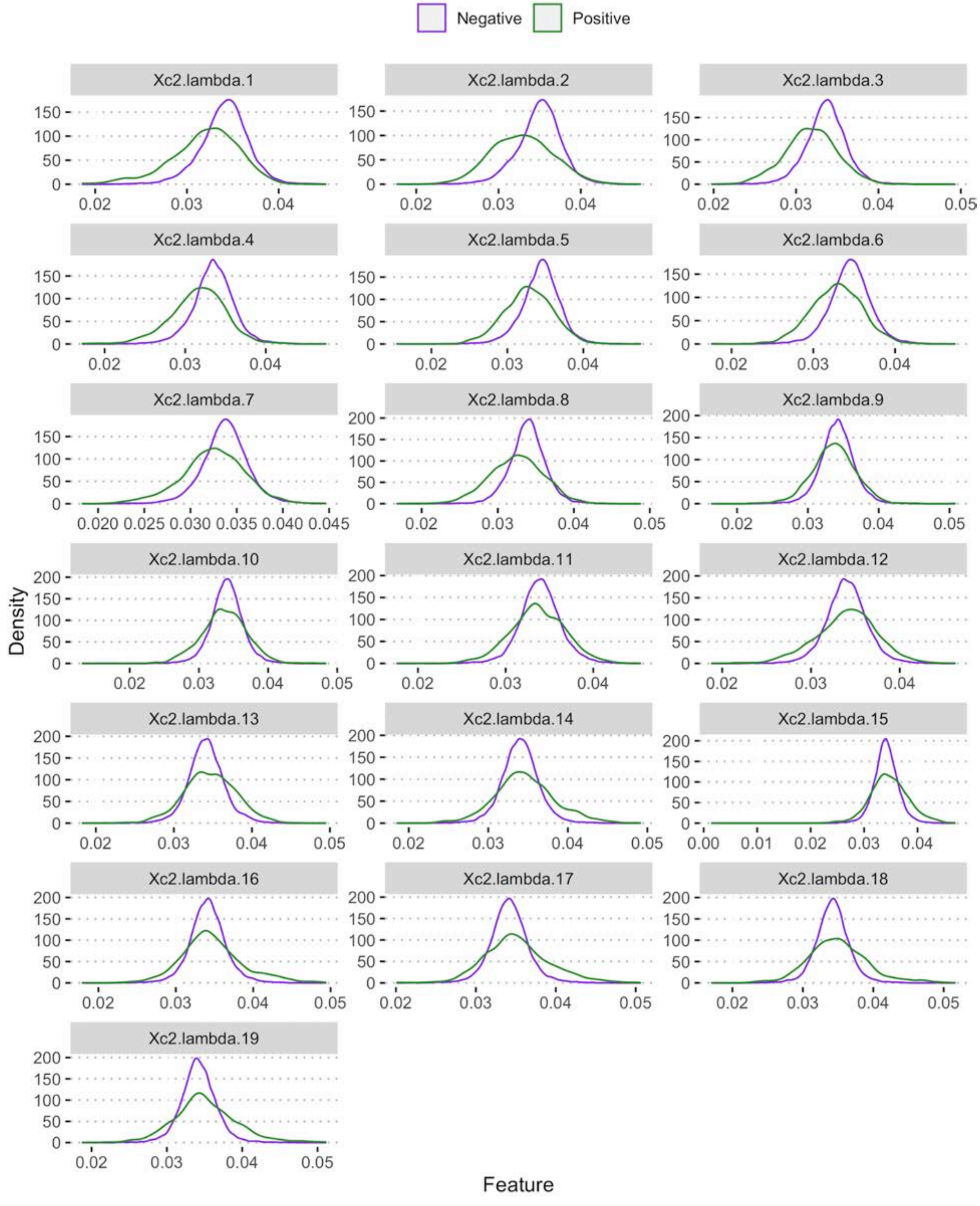


Figure 2.7: The feature distribution for the precursor training data for the lambda values of the pseudo amino acid composition in the negative (background) and positive (target) dataset.

2.3.2.2 Feature differences between mature and precursor proteins

As shown in Figure 2.4 and 2.5, AMP precursors have a different sequence length distribution from mature peptides and are therefore likely to be more distinct from each other. The propeptide regions of AMPs have previously been shown to be more conserved in comparison to the mature peptides (Nicolas, Vanhoye and Amiche, 2003; Fjell, Hancock and Cherkasov, 2007; Rončević *et al.*, 2018). Furthermore, the physicochemical properties, hydrophobicity and net charge, were found to be lower in propeptides (n = 223) in contrast to mature peptides (n = 970) (Fjell, Hancock and Cherkasov, 2007) based on AMPs from the UniProt database. To determine whether physicochemical properties of full-length precursor proteins and mature peptides differ, five physicochemical properties, amphiphilicity, net charge, hydrophobicity, molecular weight and isoelectric point, were calculated on the 806 precursor AMPs and 768 mature AMPs obtained from Swiss-Prot (see training data section) (see Figure 2.8).

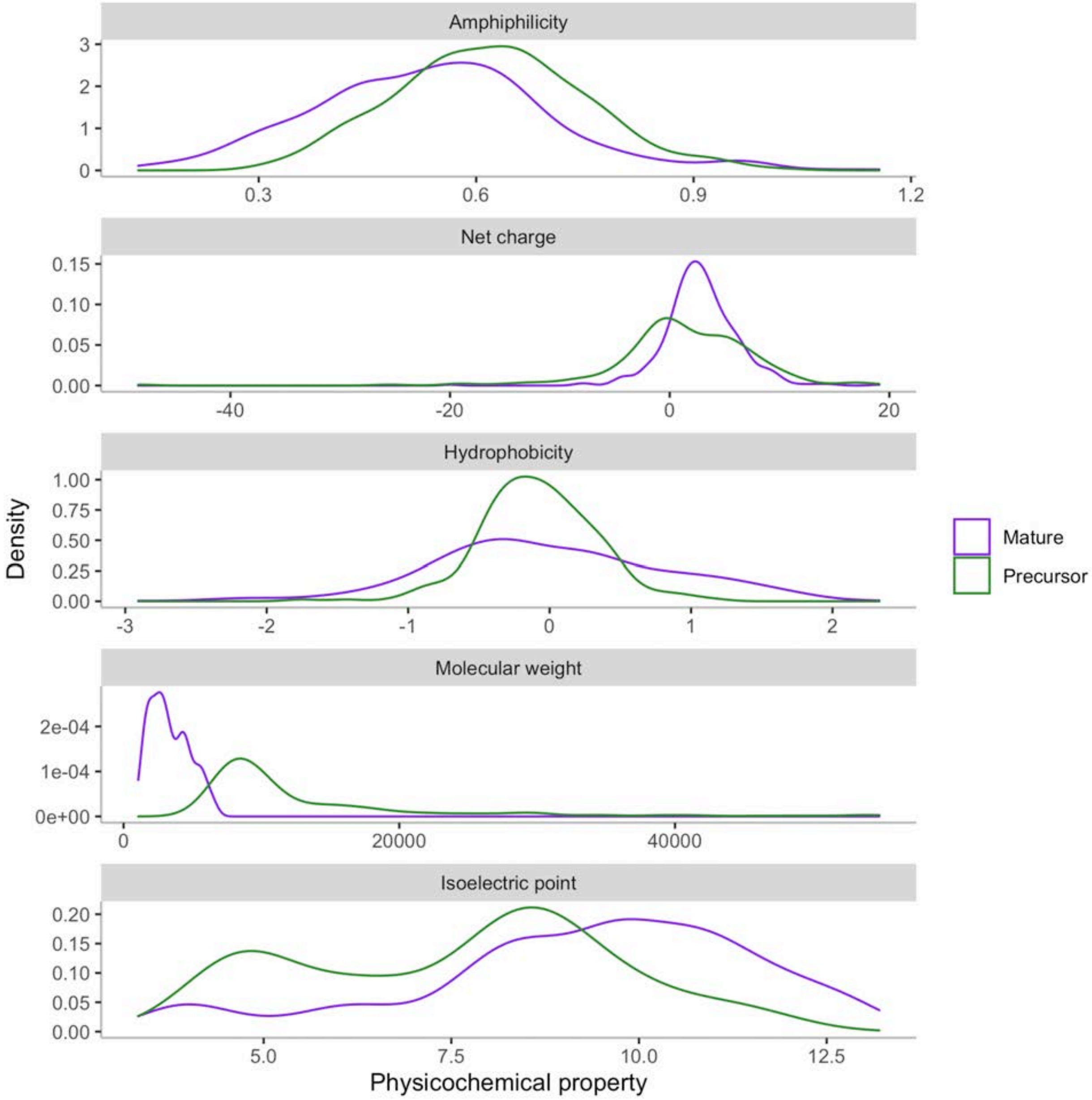


Figure 2.8: Density plots showing the distribution of five physicochemical properties in mature and precursor protein sequences.

With the exception of molecular weight, no clear distinction between the physicochemical properties of mature and precursor proteins were observed. The change in molecular weight between precursor and mature peptides was expected, however, it is interesting that hydrophobicity and net charge do not appear to be higher in mature peptides as was observed between propeptides and mature peptides (Fjell, Hancock and Cherkasov, 2007). This could be affected by the relatively low number of propeptides used in Fjell, Hancock and Cherkasov (2007) but it is more likely that in this study, the full protein precursor was used of which the propeptides and mature peptides are a part of and therefore it is more difficult to differentiate between the two as the physicochemical properties are combined. Despite the lack of a clear distinction between precursor proteins and mature peptides, there appears to be a detectable difference in some cases. Perhaps with additional AMP sequences and by including enough relevant features together, this could inform statistical classifiers of the difference between precursor proteins and mature peptides. Nevertheless, there is a clear length and molecular weight distinction between the mature peptides and the precursors which could bias a model if both mature peptides and precursors are used to train the model. This supports the choice to implement two separate models (one trained on precursor proteins and one on mature peptides) into ampir.

2.3.3 Model details

During development of ampir I explored two machine learning approaches widely used in AMP prediction (Liu *et al.*, 2017), 1) the support vector machine with radial kernel (SVMr) and 2) a random forest (RF) algorithm. Both algorithms were used to train classification models with the caret R package. Data preprocessing and model training details were as follows: data were centred and scaled for normalisation and three repeats of 10 fold repeated cross validation were used to train each model. Prediction probabilities were calculated in each model and the models were tuned via a grid search of hyperparameter values as selected by caret. SVMr performed marginally better than RF, therefore SVMr was chosen as the final classification method to be implemented into ampir.

2.3.4 ampir as a framework

In addition to providing built-in classifiers for mature peptide sequences or full-length precursor proteins, ampir provides a framework to allow researchers to easily build custom models and use them for fast genome-wide prediction. Specifically, ampir provides computationally efficient methods (implemented in C++ with multicore support) for calculating features commonly used in AMP prediction including physicochemical properties (Meher *et al.*, 2017; Bhadra *et al.*, 2018) and Chou's pseudo amino acid composition (Xiao *et al.*, 2013). In genome-scanning contexts such custom models will be especially important since they allow for optimisation (a) within a restricted taxonomic range, or (b) with restricted or biased input data (e.g. only secreted proteins). Researchers can take full advantage of the caret framework to optimise for these contexts on the basis of training data, feature selection and underlying machine learning approach. The resulting models can then be provided directly to ampir for prediction.

2.4 Models' performance

The performance of ampir's models were tested against three recently published AMP predictors: iAMPpred by Meher *et al.* (2017), AmPEP by Bhadra *et al.* (2018) and AMP Scanner by Veltri, Kamath and Shehu (2018). These AMP predictors were selected because at the time, they were the most recently published methods that were able to analyse several thousand sequences at a time. A more recent AMP predictor, dbAMP by Jhong *et al.* (2019) was released, however this predictor focuses on classifying AMPs to different taxa which is outside of the design scope for ampir and therefore was not included in benchmarking analysis.

The trained ampir models were evaluated on their respective hold-back test sets, not included in the training data. The ampir_mature test set contained 1,310 sequences and the ampir_precursor test set contained 3,262 sequences. Three other AMP predictors (AMP Scanner, amPEP and iAMPpred) were also evaluated against the ampir test sets as a benchmark using a range of performance metrics (Table 2.3 and 2.4). Although this evaluation method is likely biased towards the ampir test sets, this practice of comparing performance of multiple predictors using a hold-back set from a specific predictor is common in the AMP prediction literature, e.g. see Veltri, Kamath and Shehu (2018).

Table 2.3: Evaluations of model performance tested on the ampir_mature test set.

AMP models	Acc	Rec	Sp	Prec	F_1	AUROC
ampir_mature	86	84	88	87	86	92
ampir_prec	60	26	94	81	39	68
AMP Scanner	75	92	58	68	78	81
AmPEP	76	95	58	69	80	90
iAMPpred	70	88	53	64	75	77

Acc: accuracy, Rec: recall, Sp: specificity, Prec: precision, F_1: F_1 score, AUROC: area under the receiver operating characteristics curve. Units are in percentage.

Table 2.4: Evaluations of model performance tested on the ampir_precursor test set.

AMP models	Acc	Rec	Sp	Prec	F_1	AUROC
ampir_prec	88	77	99	87	82	97
ampir_mature	50	100	0	09	17	85
AMP Scanner	70	89	51	15	26	82
AmPEP	46	07	85	05	06	52
iAMPpred	47	90	03	09	16	50

Acc: accuracy, Rec: recall, Sp: specificity, Prec: precision, F_1: F_1 score, AUROC: area under the receiver operating characteristics curve. Units are in percentage.

The area under the receiver operating characteristics curve (AUROC) value is shown in a visual representation in a ROC curve plot (see Figure 2.9) for both the precursor and mature peptide test set. Model performance improves the closer the curve is towards the top left corner. For both test sets and respective ampir models, ampir's curves are the closest to the top left and subsequently also have the best area under the curve (AUC) value (92% for the ampir_mature model and 97% for ampir_precursor). The main thing that can be observed is that all models, with the exception of ampir_precursor, perform very well on the ampir_mature test set. However, the performance of these models drastically decreases in the ampir_precursor test set, which consists of full-length proteins. This clearly shows that models trained with a large proportion of mature sequences are not suitable on datasets that contain full-length proteins.

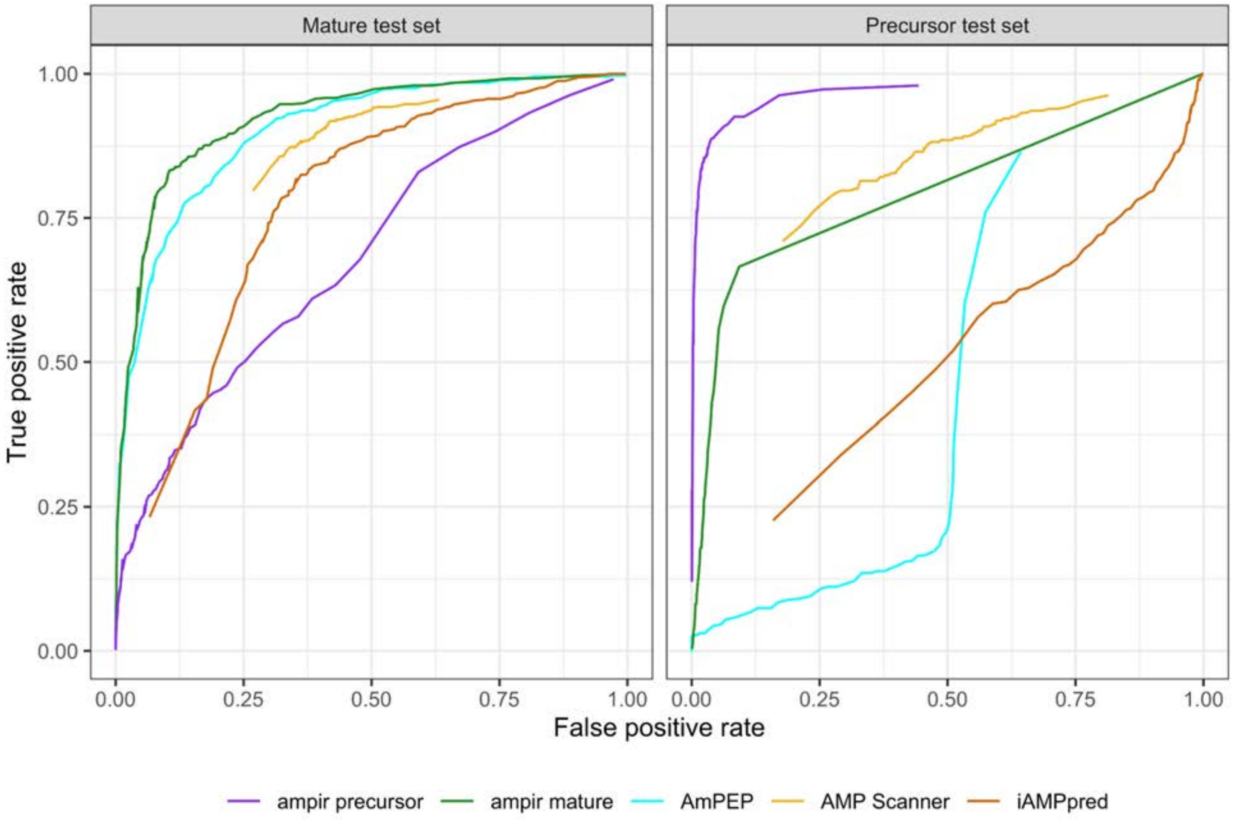


Figure 2.9: Performance of a range of AMP predictors against the ampir_mature and ampir_precursor test set.

Another important thing to note is that AmPEP, AMP scanner and iAMPpred perform well on the ampir_mature test set, likely because their training data consisted of many mature sequences. However, the reported results for each AMP predictor were different, and much better, in their respective papers because they used their own test set (see Table 2.5, note that the four performance metrics common in all papers were used for comparison). The AMPs used in ampir's test set likely overlap with their positive dataset, as the AMP dataset used in AMP predictors does not vary a great deal as there are only a limited number of AMPs. However, there are many different types of proteins that can be used for the negative dataset, and as previously mentioned, these AMP predictors filter the negative dataset to remove AMPs and sequences that are similar to AMPs. Filtering out these sequences therefore has substantial consequences on the performance of models when analysing diverse biological data expected to be present in genomes. These consequences are exacerbated when models are trained with a large proportion of mature peptides, which could compositionally differ from precursor proteins, and are not present in their mature form in most datasets derived from genomes.

Table 2.5: Performance evaluation results from existing AMP predictors as obtained from their references.

AMP models	**Acc**	**Rec**	**Sp**	**AUROC**	**Reference**
AMP Scanner	91	90	92	97	Veltri, Kamath and Shehu (2018)
AmPEP	96	95	97	99	Bhadra *et al.* (2018)
iAMPpred	94	93	95	98	Meher *et al.* (2017)

Acc: accuracy, Rec: recall, Sp: specificity, AUROC: area under the receiver operating characteristics curve. Units are in percentage.

2.5 ampir’s software engineering practices

Modern software engineering practices such as version control, continuous integration, adherence to community accepted naming conventions, and an open development process were adopted in the creation of ampir to ensure that it was high-quality software, and so that improvements could be made while minimising the possibility of introducing bugs.

ampir was designed for a bioinformatic purpose, that is, it was developed to aid understanding of biological data using a multi-disciplinary approach that includes biology, mathematics, statistics and software engineering. With increased biological data availability, more and more bioinformatic software is becoming available however, not enough emphasis is placed on software quality. In the context of AMP prediction there appears to have historically be an over-emphasis on designing software to suit novice users leading much investment in graphical interfaces (Kumar and Dudley, 2007). Although this is most certainly useful, it sometimes hinders the ability of the software to be able to be used in an automated bioinformatics pipeline used for large data analysis. Command line software is generally better suited for this task and is also generally easier to use by technically proficient users. Nevertheless, such software must be well documented. In addition to providing guidance on usage, good documentation provides confidence to the user that the software works as intended and is therefore an important aspect to software engineering.

2.5.1 Open development process

The ampir package is not only open source, but is developed in a completely open manner, whereby the complete version history and the ability to issue pull requests is made available on a web server. Open source software means that the code that generated the software is available for anyone to view, edit and redistribute. Open development goes one step further in providing access, and the ability to contribute to, all of the code, including that in current development as well as all previous versions. This allows total transparency how the program output is generated which subsequently contributes to two important aspects: reproducibility and full method disclosure. These allow anyone to reproduce your results and also detect potential code errors. For example, code that is translated from mathematical equations can contain computational errors and result in decreased accuracy. Such an error was discovered in an algorithm that created a large and widely used temperature dataset (Ince, Hatton and Graham-Cumming, 2012). Therefore, it is important that both the software source code and the methodology code is available. A widely used tool in software development that allows easy sharing of code is a distributed version control system (DVCS). In a DVCS the code is potentially present in multiple repositories each of which keeps track of its own changes, or “commits”. The repositories can be accessed and effectively “cloned” by anyone into their “local” repositories based on their computers via a server that stores a “remote” repository. Separate commits in different repositories may be merged in order to synchronise changes made by multiple developers and the commit history shows all the changes made and who made them (Zolkifli, Ngah and Deraman, 2018). Commits on the local repositories can be uploaded, or “pushed” to the remote repository and then downloaded, or “pulled” by local repositories belonging to other software developers or collaborators (see Figure 2.10).

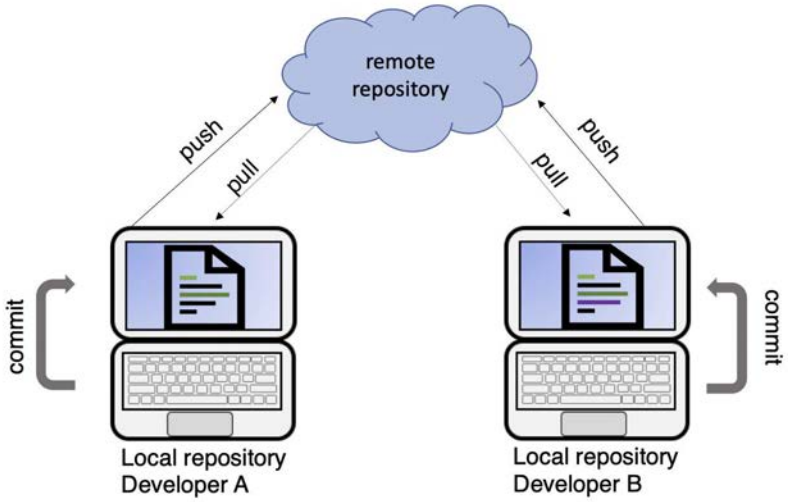


Figure 2.10: Typical software development workflow using a distributed version control system (DVCS). A single remote is shown, however, in theory a DVCS can support multiple remotes as well.

DVCSs are implemented in software like Git (https://git-scm.com/) and Mercurial (https://www.mercurial-scm.org/) which both launched in 2005. These can be hosted on a cloud-based server which allows users to store, manage and share repositories. Until July 2020, Git and Mercurial were both hosted on Bitbucket, however, Bitbucket announced it would drop support for Mercurial and focus on Git alone as Git is most commonly used in software development (Chan, 2020). Git is also hosted by GitHub, a specialised Git cloud-based server. GitHub provides a web graphical user interface for Git repositories which can include source code for software. Ampir has been developed using both Git and GitHub because of their popular and useful features. GitHub allows users of ampir, hosted on GitHub, to easily raise issues such as potential bugs or software improvements. In addition, GitHub also includes useful software engineering features such as code review and continuous integration.

2.5.2 Continuous integration

Continuous integration (CI) refers to a software development process where during the building and maintaining of software, the code is automatically built and tested when new changes are merged by a developer or collaborator (Meyer, 2014). In practice this usually relies on the use of a central cloud-based remote repository that interacts with a separate CI server that tests and builds the software. After the CI server runs tests that the software builds successfully, it notifies the developer team that the build “has passed”. If the CI server encounters build errors, it will also notify the developer team and advise them where the error was encountered so it can be more easily fixed. This testing by the CI server provides confidence to the developers that they can modify the software without breaking it. This subsequently enables collaboration and improvement over time because of the security that the tests provide. Collaboration could come from anyone that wishes to improve the software. The general process for this is that the collaborator clones the software repository to a separate branch, commits the changes they want to make and then creates a pull request. A pull request lets the developers know what changes were committed by the collaborator, which they can then review. The pull request is also automatically tested with the CI server to ensure the changes do not break the build (see Figure 2.11). If the build passes and the developers agree with the code changes made by the collaborator, the developers can then merge the code into the main software repository.

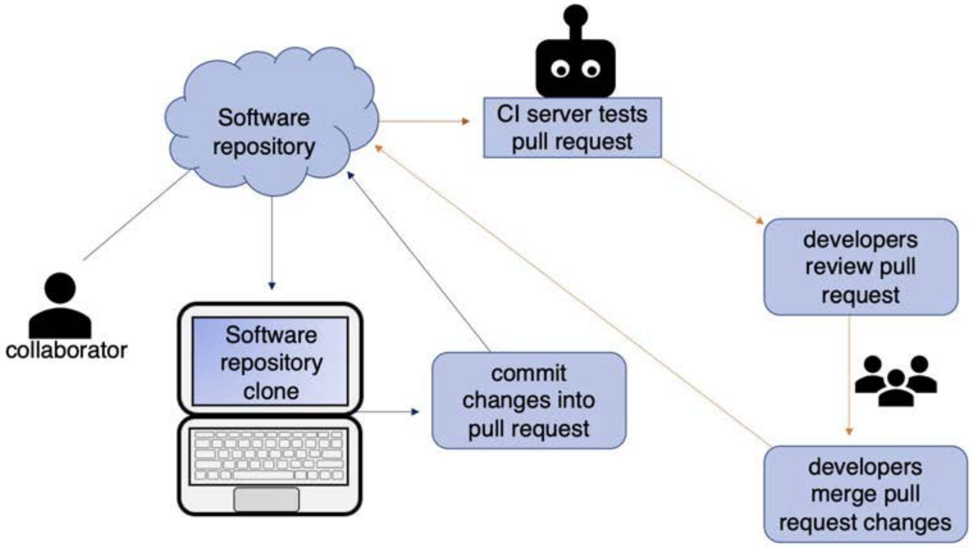

Software repository
CI server tests pull request
developers review pull request
collaborator
Software repository clone
commit changes into pull request
developers merge pull request changes

Figure 2.11: A diagram of the general pull request process. A collaborator clones the software repository and commits changes. The collaborator then requests a pull request from the original software repository which is automatically tested by the CI server to see if the changes made do not break the software. The CI server alerts the developers after it finishes the tests who then review the pull request and then, if the developers agree with the software changes, merge the pull request into the original software repository.

There are a variety of CI servers available such as Buildbot (https://buildbot.net/), CircleCI (https://circleci.com/), Jenkins (https://www.jenkins.io/) and TravisCI (https://travis-ci.org/). Software developers may choose a CI server that more closely suits their needs. For example, developers may want a CI server that is easy to use, or has fast builds, or is customisable, or has high security (Hilton *et al.*, 2017). TravisCI was chosen as ampir’s CI server because it integrates well with Git and GitHub, is easy to use, supports R and C++, and provides good documentation.

2.5.3 Code tests

The tests that the CI server runs to test the build are integrated within the server. However, the developers themselves can also write tests to test the functionality of their code. These tests make the code more robust and help ensure that the software functions as intended. One way to write a test is to match the expected type of an input or output to a known value. This may seem simple but it is especially important when software is written in dynamic languages such as R that do not automatically check expectations around data types. Errors that arise when software expects a certain data type to be used as input but is in fact provided with different data type can be difficult to debug because they are disconnected from the real source of the issue. For example, ampir's main function `predict_amps` expects an object of type `data.frame` as input with sequence names in the first column, and amino acid sequences in the second column. In R, a `matrix`, `data.frame` and `tibble` all look like a table with columns and rows and can store the same data. However, for an old version of `predict_amps`, if a `matrix` was used, `predict_amps` errored with the following message:

```
Error: $ operator is invalid for atomic vectors.
```

This error refers to code written within the `predict_amps` function and is therefore not at all informative to the user. An alternative, and perhaps more detrimental error mode is the silent error. Silent errors occur when something goes wrong in the software but nothing gets reported. For example, in an old version of `predict_amps`, if a `tibble` was used as input, `predict_amps` ran without an error but provided the wrong output. The expected output should have contained a third column with probability values but instead, it contained a third column filled with `NA`. By writing tests that examine the input and output, these errors were found during the development of ampir. These additional tests are also integrated with the CI and software management system and therefore get tested with every change made to the software.

It may not always be obvious for what code tests should be written and it is easy to forget to write additional tests for additional code lines. There are tools such as Codecov (https://codecov.io/) that can be integrated with the CI provider that calculate the code coverage in relation to the tests and provide a visual representation of the code coverage. This is a useful indication to the developers which lines of code have or have not been tested. Furthermore, when new code is committed to the repository, Codecov automatically checks the code for its code coverage and alerts the developer to the increase or decrease of the overall code coverage for the project. This is particularly useful when new features are being implemented by either the developers themselves or collaborators via pull requests, as it encourages test writing for new code (Hilton *et al.*, 2017). New features can include new functions or extensions of current functions. Therefore, larger functions that contain more code require a larger number of tests that evaluate this code. It is important to remember that no test is perfect but they do provide some assurance for code quality and are able to reveal further potential problems. ampir 1.0.1 contains tests for each function and its overall code coverage is 98.69%. A list of tests implemented for the various ampir functions can be found in Table S2.1

2.5.4 Optimisation for high-throughput

A key requirement of any genome scanning software is the ability to efficiently process a large number of input sequences. This capability is referred to as high-throughput. There are two main aspects of software that contribute to this: speed and parallel computing. In software development, speed is influenced by the way code is written and by the programming language used. It is easy to write very slow and inefficient code in R unless care and attention is paid to these aspects (Wickham, 2019). R is highly flexible, and there can be several ways of writing code to perform a certain action, however some methods are faster and more efficient than others. Identifying slow code can be difficult, especially for those users of R that are not formally trained in programming (Wickham, 2019). Code profiling can be used to analyse the execution time of code and find potential runtime bottlenecks (Bergel *et al.*, 2012). R provides a code profiling tool called profvis which records the functions being run at frequent intervals and reports on the execution time and memory usage (Wickham, 2019). This tool was used to analyse ampir and helped locate various slow code sections which consequently were rewritten and improved. For example, provfis was run on a function in ampir that calculates Chou’s pseudo amino acid composition (Chou, 2001), on all given protein sequences, `calc_pseudo_comp`. It incorporates a loop that collects all the calculations for each sequence and then combines them all into a single data frame. When this was first written, the combining action of the dataframes occurred iteratively, i.e., the dataframe “grew” with each loop which is slow because it forces R to store data in its memory until the loop has finished. To speed it up, the code was altered to add each calculation into its own separate table in a list inside the loop and once that finishes, all tables in the list are combined into a single dataframe outside of the loop. This is faster because the function used to combine the tables is effectively only called once. For smaller datasets, the speed gained may not be significant. However, for large datasets, such as expected for ampir, these speed ups from using more effective code can greatly decrease the overall runtime. Therefore, analysing software code after it has been written can be a powerful method to improve the performance of the software.

Parallel computing can increase the speed of software by breaking up the software tasks to smaller pieces and using multiple cores to execute these smaller pieces simultaneously. ampir integrates the (R Core Team, 2021) parallel package which can be used on High Performance Computing (HPC) systems to select additional cores to speed up ampir’s main function (see Figure 2.12).

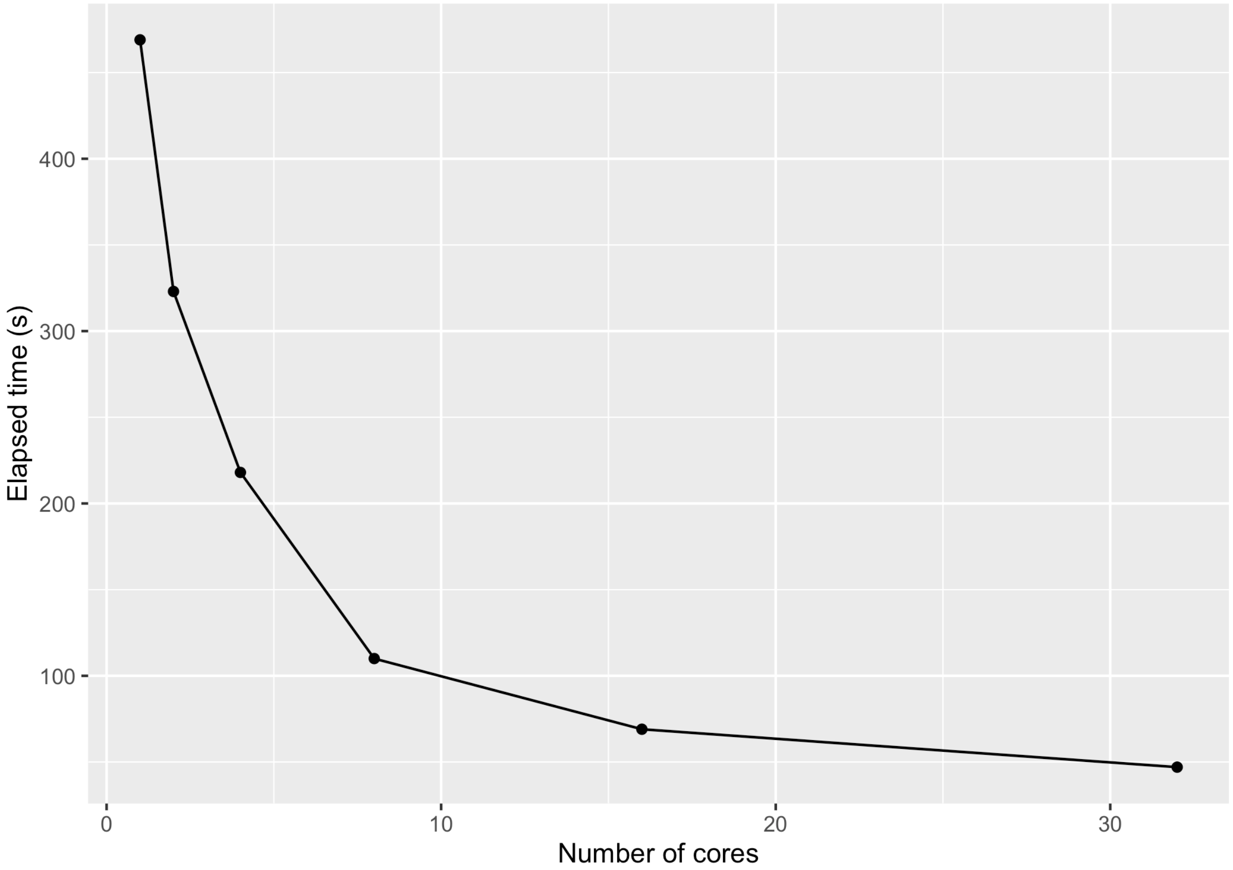


Figure 2.12: Performance of ampir as a function of core count when running `predict_amps` on a dataset of 77,000 proteins.

This means that users of ampir can easily speed up their analysis which is especially useful when implementing ampir on multiple proteomes.

2.5.5 ampir distribution and user interface

ampir is distributed as an R package that includes code, documentation, data and tests (Wickham, 2015) via the Comprehensive R Archive Network (CRAN). CRAN is the primary repository for R and R packages (https://cran.r-project.org/) and provides an easy and standard installation process familiar to R users. During the package submission process, CRAN incorporates strict tests to ensure that the installation process is reliable for all platforms and previous R versions. In addition, CRAN requires packages to conform to extensive instructions related to the structure, documentation, code and functionality of the package (https://cran.r-project.org/doc/manuals/r-release/R-exts.html). CRAN also supports and encourages the writing of additional documentation called ‘vignettes’. The vignette for ampir includes executable examples demonstrating use of the functions in the package and provides explanatory text with context about when to use them.

In addition to the default command line interface R offers for its users, ampir was also developed as a Shiny app. Shiny is an R package that can be used to build graphical user interfaces on web servers (Chang *et al.*, 2021). The ampir Shiny app, available via https://ampir.marine-omics.net/, was designed so users can upload their FASTA file containing amino acid sequences which can then be classified by either the precursor model for full-length proteins, or by the mature model for mature peptide sequences. The prediction results plus original sequence can then be downloaded by the user as a comma separated file for further analysis.

2.6 Conclusion

Antimicrobial peptides are an important part of the innate immune system and help maintain the health of their host organism. Many machine learning methods have been developed to try to identify these peptides *in silico*. However, at the time this chapter was written, all of these methods had shortcomings preventing their use for AMP detection on a genome-wide scale. To facilitate a genome scanning approach to AMP discovery, a new machine learning model to classify AMPs was constructed and embedded in an R package, ampir. ampir was designed with good software engineering practices to be easy and effective to use, and specifically optimised for high-throughput analysis.

Chapter 3: Benchmarking antimicrobial peptide (AMP) machine learning models in a genome-scanning context

3.1 Abstract

Modern pipelines for AMP discovery often begin by using computational tools to identify peptides with putative antimicrobial activity from a large set of candidate proteins. These candidate sets are often derived from the complete set of translations from a transcriptome or annotated genome of an organism. Although many AMP classification tools now exist, their effectiveness on realistic input datasets in the context of ‘omics-based AMP discovery has not been well explored. This chapter explores the training and test data used to build and evaluate a range of recently published AMP predictors. It introduces the idea that complete proteomes from well-studied taxa may be valuable datasets to assess the performance of AMP prediction software in a whole-proteome scanning context. It was found that the test and training data used by most AMP predictors has substantial biases in composition compared with complete proteomes, and that the predictive errors that arise from this are not captured by most performance metrics. Two major sources of compositional bias with impacts on model performance were identified: (1) imbalance between positive and negative classes and (2) selection of training sequences that are unlike those of typical input data. Based on extensive benchmark tests and theoretical analysis, performance metrics best suited to capturing the issue of imbalance were identified. Finally, it was demonstrated that the inclusion of precursor proteins in training datasets results in substantial performance improvements in a genome-scanning context.

3.2 Introduction

Antimicrobial peptides are a ubiquitous feature of most species across all major kingdoms of life. In organisms with well characterised AMP proteomes, such as *Arabidopsis thaliana,* several hundred distinct AMPs are present which suggests that the total diversity of AMPs across all life is likely many million molecules. Despite this, only around three thousand AMPs have been described in Swiss-Prot (UniProt Consortium, 2021). Filling this gap requires efficient methods to discover new AMPs. ‘Omics scanning workflows in which large databases of proteins are scanned for potential candidates prior to experimental screening are a promising method to increase the rate at which new AMPs can be discovered. A key element of the ‘omics scanning workflow is the screening step, and requires “AMP predictors”, computer programs that can predict AMP activity from a peptide sequence. In fact, many such AMP predictors now exist, and there has been an explosion in the number and variety of these programs in recent years. Several reviews (Gabere and Noble, 2017; Liu *et al.*, 2017; Xu *et al.*, 2021), have recently attempted to assess AMP predictors using benchmark metrics and test datasets. However, results presented in this chapter will show that these benchmarking approaches fail to capture the key aspects of model performance that matter in an ‘omics scanning context.

Generally, AMP predictors are benchmarked against a hold-back test set from their own training data and sometimes also against published benchmark sets (Bhadra *et al.*, 2018; Veltri, Kamath and Shehu, 2018; Kavousi *et al.*, 2020; Yan *et al.*, 2020). While these approaches can provide a statistically sound measure of performance within a specific context, the degree to which this reflects real-world usage scenarios is almost never tested. One key issue is that when a hold-back test set is used its statistical composition will be nearly identical to the training data but (as shown in this study) this can be very different to the composition of input datasets used in realistic AMP discovery pipelines. In such situations performance metrics are almost always inflated since the model is likely to perform best on test data that is similar in composition to that used for training.

While testing a predictor on every possible use case is generally not practical, there are several general features of ‘omics datasets used in AMP discovery that should be captured in a test dataset or benchmarking procedure. These include two main aspects: the first is very high data imbalance and a desire for few false positives. Predictors must perform well on input data that is highly imbalanced because AMPs only comprise a small proportion of proteins in the genome (typically < 1%). A well understood consequence of this is that it makes it challenging to construct an unbiased training dataset (Weiss, 2004; He and Garcia, 2009). However, less well appreciated is the influence this has on likely real-world use cases and the choice of benchmarking metrics that best reflect these. In an ‘omics-based AMP discovery pipeline for example, the goal is to obtain a small, high-confidence set of candidate AMPs for synthesis and testing, a scenario that places far greater emphasis on the low false positive regime of performance than typical benchmark metrics such as the Area Under the Receiver Operating Characteristics curve (AUROC), sensitivity and specificity.

A second key issue is that, in genome-scanning applications, sequences are usually only available for precursor proteins rather than mature peptides. These full-length precursor protein sequences usually arise by translating coding sequences from gene models. Since it is not generally possible to accurately deduce the mature sequence from its precursor, this information is typically not available to the user of AMP prediction software. It is also likely that precursor sequences and mature sequences differ substantially in terms of their amino acid composition and physicochemical properties. The performance of a predictor that is predominantly trained on mature peptides is therefore likely to be poor when presented with precursor sequences as input data.

In summary, it is expected that AMP predictor performance is both dependent on 1) the composition (i.e. the types of molecules present) and 2) the balance (i.e., the prevalence of AMPs) of the test dataset. This study explores how these issues affect the appropriateness of training and test data, as well as benchmark metrics used for AMP prediction in an ‘omics AMP discovery context.

3.3 Major sources of compositional bias in training and test data

3.3.1 Methods

All analyses were completed in R version 4.1.2 (R Core Team, 2021) unless stated otherwise, using the RStudio integrated development environment, version 2022.02.0+443 (RStudio Team, 2021) and the tidyverse R package, version 1.3.1 (Wickham *et al.*, 2019).

As representative examples of the type of data that would be used as input in 'omics scanning applications, the complete proteome sets of *Arabidopsis thaliana* (a plant) and *Homo sapiens* (human) were obtained from UniProt proteomes, https://www.uniprot.org/proteomes/ (accessed 23 January 2021). Both organisms have been intensively studied and as a consequence, their reference proteomes are likely to include sequences for the vast majority of protein-coding genes. Functional information for the proteins, including AMPs in these species is among the most complete available, but even for these organisms it is highly likely that some known AMPs have not been identified. For this chapter, it is assumed that these proteomes are completely classified for AMP activity (i.e. every AMP correctly identified).

AMP predictors used in this chapter were predominantly selected based on their ability to cope with high throughput analyses. This is an essential practical requirement for 'omics scanning workflows where one would scan (at minimum) an entire proteome (~30,000 sequences). The AMP predictors used in this chapter are iAMP-2L (Xiao *et al.*, 2013), amPEP (Bhadra *et al.*, 2018), Deep-amPEP30 (Yan *et al.*, 2020), amPEPpy (Lawrence *et al.*, 2020), AMP scanner v2, (Veltri, Kamath and Shehu, 2018), AMPlify (Li *et al.*, 2020), AmpGram (Burdukiewicz *et al.*, 2020) and ampir (Fingerhut *et al.*, 2020).

Most published AMP predictors use a subset of the approximately ~4,000 experimentally verified AMPs as their positive training data. This results in a high degree of overlap between predictors and means that they tend to share compositional biases. I attempted to quantify this bias by comparing these AMP test and training datasets with the composition of proteomes of *A. thaliana* and *H. sapiens.* The idea in making such comparisons is that these real proteomes are representative of real input data (i.e. unbiased), however, it must be acknowledged that they capture only a very small subset of taxonomic diversity and that AMPs may not be fully classified even for these heavily studied species.

3.3.2 Sequence structure of AMPs based on annotated features in Swiss-Prot

Using one simple metric, protein length, it is possible to capture many aspects of compositional bias because there are major differences in length between most non-AMP proteins, AMP precursors, and AMP mature peptides. To demonstrate this, I surveyed precursor sequences for AMPs listed in Swiss-Prot (reviewed proteins in UniProt, found via the keyword “Antimicrobial” [KW-0929], accessed April 2021), revealing the typical sequence structure of AMP precursor proteins. Since the Swiss-Prot database includes the positions of the signal peptide, mature peptide, and C-terminal region for many well characterised AMPs, it was possible to plot the distribution of these features as a function of amino acid position (Figure 3.1). This shows that the signal sequences typically comprise a very short (less than 10 amino acids) sequence at the N-terminus. Sections of the sequence that range between 10 and 60 amino acids are largely mature peptides, and the remaining C-terminal sequences are highly variable in length, with most around 100-200 amino acids in length.

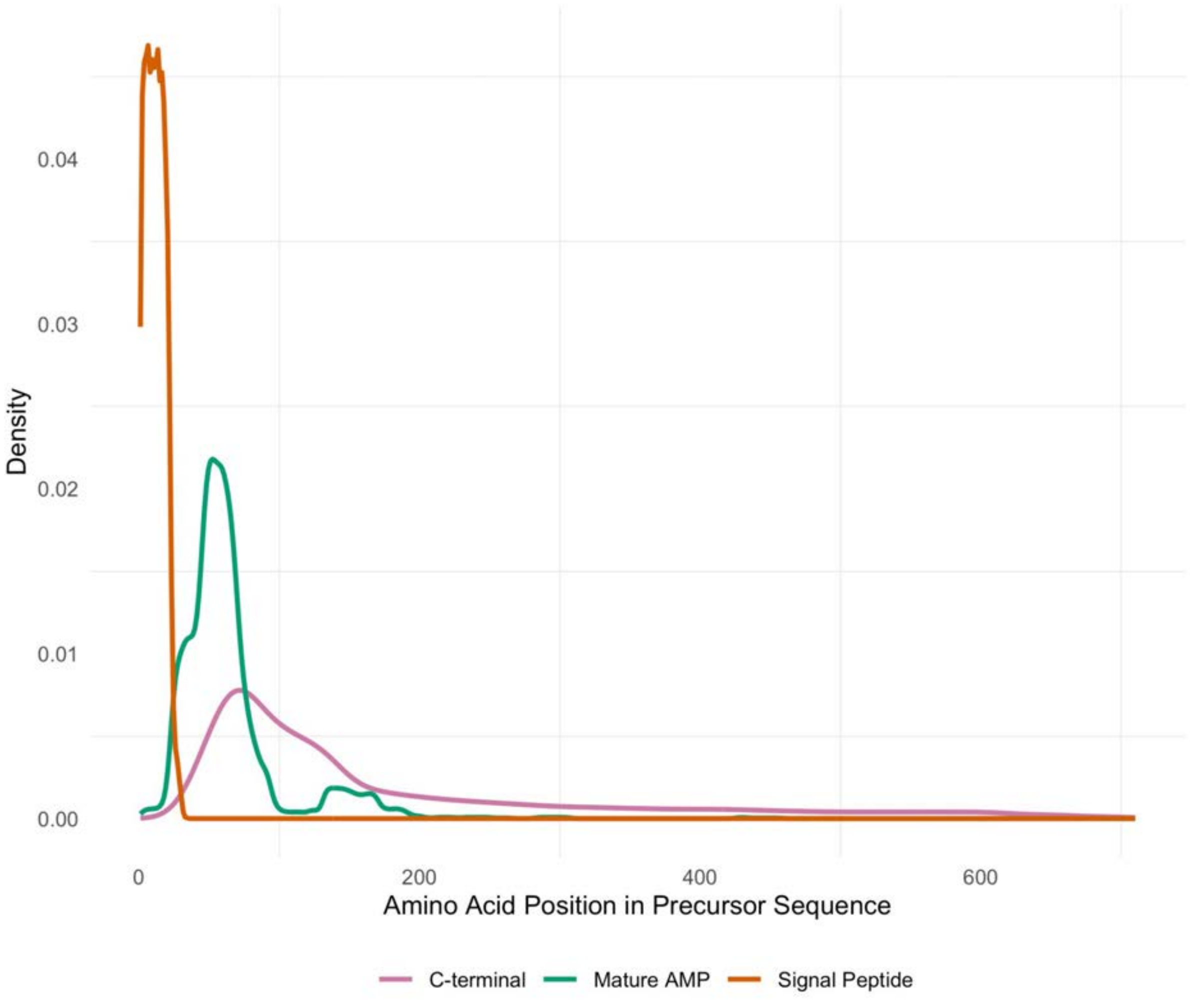


Figure 3.1: Components of a typical AMP precursor sequence as a function of amino acid position based on analysis of 831 AMP sequences with length > 50 in Swiss-Prot.

Understanding the biological significance of components of an AMP precursor sequence means that sequence length can be used to indicate whether a sequence is a mature peptide or a full-length precursor protein. Based on the results shown in Figure 3.1 it can be inferred that mature peptides should have a narrow range of lengths, centred at around 50 amino acids, whereas the lengths of precursor sequences are always longer, but also spread across a broader range.

3.3.3 Sequence length distributions

Examination of the sequence length of AMPs and non-AMPs within the training and test sets of the AMP predictors, revealed significant biases compared with the proteomes of *A. thaliana* and *H. sapiens* (see Figure 3.2). It also revealed differences in the model evaluation approach used by predictors. In most cases the test and training data (i.e. Figures 3.2A versus 3.2B) have near-identical length distributions reflecting the tendency for most predictors to use a randomly held-back portion of the same data used to generate the training set. A notable exception was AmPEP which used a test dataset published by Xiao *et al.* (2013) that has been promoted as an independent benchmark (Meher *et al.*, 2017; Bhadra *et al.*, 2018; Veltri, Kamath and Shehu, 2018; Santos-Júnior *et al.*, 2020). However, as detailed below (see Figure 3.3 and accompanying text) this test set overlaps with training data used by many predictors, which compromises its independence, and as shown in Figure 3.2, the length distributions of both AMPs and non-AMPs in the test data sets of many predictors do not resemble those of the two real proteomes.

Comparing the length distributions of AMPs and non-AMPs in the training and test data to the *A. thaliana* and *H. sapiens* proteomes, it is clear that the majority of predictors include a high density of mature AMPs in their positive dataset. The AMPs for these predictors have a strong peak in sequence length at around 50 amino acids, whereas for the real proteomes (Figure 3.2C) this peak is at a sequence length of 100. Interpreting this in the context of results shown in Figure 3.1, it most likely reflects the fact that most AMP sequences in the proteomes are full-length precursors while those that predominate in training and test data for predictors are mature peptides. The only predictor with a length distribution for test and training data that qualitatively matches that of the proteomes is ampir_precursor. This most likely reflects its deliberate exclusion of mature AMP sequences in favour of precursor proteins.

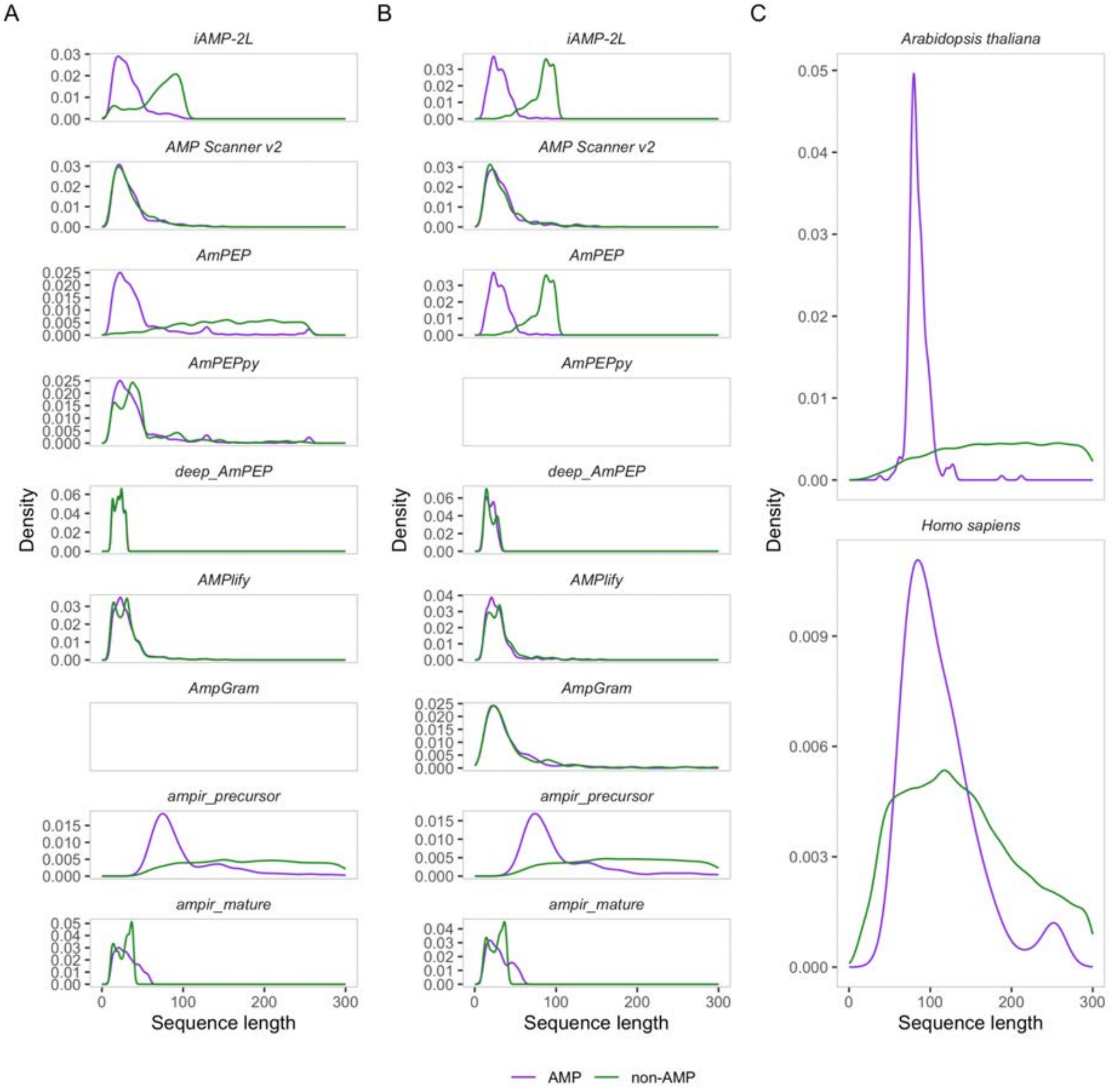


Figure 3.2: Comparison of sequence length distributions for positive (AMP; purple) and negative (non-AMP; green) fractions in training (A) and test data (B) of nine AMP predictors, and for the proteomes of *Arabidopsis thaliana* and *Homo sapiens* (C). The training data for AmpGram and test data for AmPEPpy are blank as these were not available. Sequences longer than 300 are not shown for sequence length distribution clarity.

There are several databases that list AMPs with confirmed activity, including APD3 (Wang, Li and Wang, 2016), DRAMP (Fan *et al.*, 2016), dbAMP (Jhong *et al.*, 2019) and UniProt and most predictors use one or more of these as the basis for constructing their positive AMP dataset. In addition, some AMP databases, e.g. APD3, impose sequence length restrictions for AMPs included in their database (see chapter 2.3.1.1). Due to these shared origins, it is likely that some overlap exists between the positive datasets of AMP predictors. Both the degree of overlap, and the interaction of overlap between the length distribution of AMPs were explored.

3.3.4 Overlap between data used by different predictors

To examine the sequence overlap between databases, the stringdist, v. 0.9.8, R package (Loo, 2014) was used to calculate the Jaro distance between all pairs of positive AMP sequences across all predictor training/test datasets as well as reference proteomes for *A. thaliana* and *H. sapiens*. The Jaro distance was chosen because it is normalised for the length of both sequences and produces a value between 0 (exact match) and 1 (completely dissimilar). Highly similar sequences (Jaro distance <0.2) were considered to be the same as these are likely close homologs at minimum, or near-identical sequences with minor variation. Manual inspection revealed that in many cases matches, at this Jaro distance, occurred between near identical sequences with minor differences, likely due to reporting conventions and/or minor discrepancies between databases. An UpSet plot was created with the ComplexUpset, v. 1.3.3, R package (Krassowski, 2021) to visualise the patterns of overlap. UpSet is a technique to visualise intersections and the frequency of these intersections, commonly by grouping the overlapping sections in a frequency bar plot (Lex *et al.*, 2014). It is an alternative to the well-known Venn diagram that is especially useful when the number of sets is large.

The UpSet plot (Figure 3.3) highlights some key patterns of overlap between datasets and how this relates to their length and precursor protein status (indicated by signal peptide). One key trend is the high degree of uniqueness of sequences used by ampir (first and second columns). Also note that there are several large groups of AMPs shared by many predictors (columns 3 and 4) which appear to be almost exclusively composed of mature sequences (short length distributions). These mature peptides comprise only a very small fraction of the *A. thaliana* and *H. sapiens* proteomes.

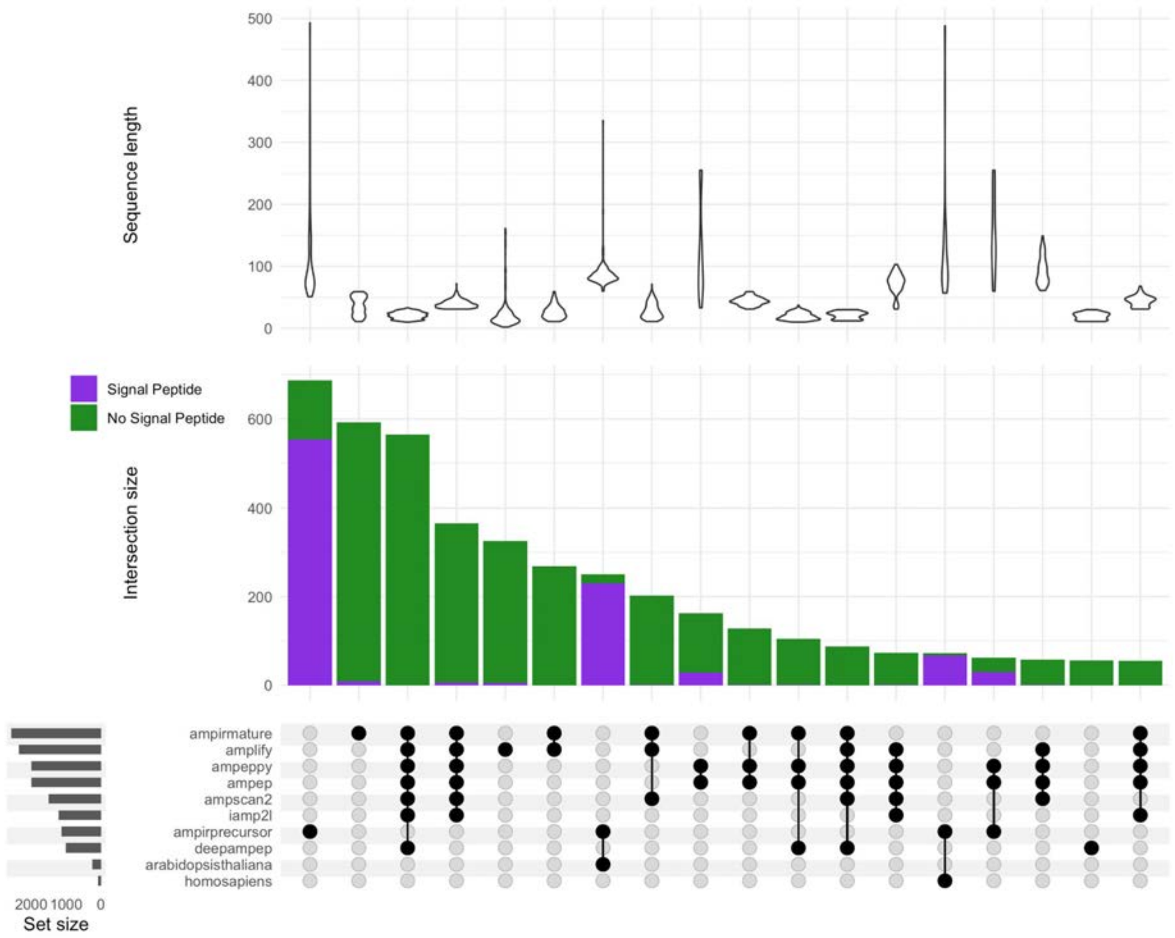


Figure 3.3: UpSet plot showing overlap between positive training data used for eight AMP predictors, and known AMPs within the reference proteomes of *Arabidopsis* thaliana and *Homo sapiens*. Each column in the plot represents proteins that are found in common between multiple datasets (a set intersection) and spans violins (top), vertical bars (middle) and dots (bottom). Dots show the membership of intersections. A dot indicates that all proteins in the intersection are contained within the training data of the corresponding AMP predictor (row: see row labels at left). Only intersections with at least 50 proteins are shown. Vertical bars show the total size of each intersection and its composition in terms of proteins that have a signal peptide or those that do not. The violin plot shows the length distribution for proteins in the intersection. Horizontal bars to the left of the dot plot show the number of proteins in each of the datasets.

As shown in Figures 3.2 and 3.3, there is a substantial divide between the composition of AMP predictor training sets and the proteomes of *A. thaliana* and *H. sapiens*, based on the length distribution of included sequences. This is significant because the majority of AMP predictors use a hold-back set to evaluate the performance of their models, such that compositional bias in the training data will also be reflected in the test data and this in turn will lead to inflated measures of accuracy. This inflated accurary issue will occur irrespective of the test metric used. In addition, with the exception of the ampir_precursor model, all AMP predictors use a balanced test set, i.e., where the number of AMPs equal the number of non-AMPs (see Table 3.1). This balance is unlikely to match the real proportions of AMPs present in a proteome, where it is likely AMPs only comprise a small proportion.

Table 3.1: The number of positive and negative sequences present in the training and test datasets in nine AMP predictors.

	Training set		Testing set		Reference
AMP predictor	**AMPs**	**non-AMPs**	**AMPs**	**non-AMPs**	
iAMP-2L	897	2,405	920	920	Xiao *et al.* (2013)
amPEP	3,268	166,791	iAMP-2L test set		Bhadra *et al.* (2018)
Deep-amPEP30	1,529	1,529	94	94	Yan *et al.* (2020)
amPEPpy	3,268	3,268	*Not specified*		Lawrence *et al.* (2020)
AMP scanner v2	1,066	1,066	712	712	Veltri, Kamath and Shehu (2018)
AMPlify	3,338	3,338	835	835	Li *et al.* (2020)
AmpGram	2,216	2,216	247	247	Burdukiewicz *et al.* (2020)
ampir_precursor	1,187	11,864	296	2,966	Fingerhut *et al.* (2020)
ampir_mature	2,586	2,657	646	664	

3.3.5 Balance of classes in training datasets

Training, and testing phases of model development are affected by dataset balance in different ways. One of the reasons that many models adopt a balanced dataset for training is that training on imbalanced data can lead to a model that overemphasises the majority class (Meher *et al.*, 2017). This is a serious problem if (as is the case for AMP prediction) the minority class is of maximum interest. The flip-side to this is that in order to balance data many cases from the majority dataset are discarded, which potentially results in loss of valuable information. Fortunately there are now several statistical approaches that deal with this issue, accommodating training on unbalanced data without over-emphasising the majority class. One category of approaches, e.g. Synthetic Minority Over-sampling Technique (SMOTE) achieves balance by synthesising additional data for the minority class (Bhadra *et al.*, 2018). An alternative approach (adopted by ampir_precursor) is to weight data in inverse proportion to their class abundance (i.e. downweight the majority class). This has been suggested to lead to increased performance of the model (Bhadra *et al.*, 2018), likely due to the presence of additional data which the model can learn from.

Irrespective of whether balanced data are used for training purposes the class imbalance issue must also be addressed when evaluating models. This issue is addressed in detail below (section 3.4).

3.4 Implications of imbalanced data on performance evaluations

As can be seen from Table 3.1 it remains common practice to evaluate AMP model predictors using a balanced testing set. In this section I describe in-detail why this approach is especially problematic when using AMP predictors in an ‘omics scanning context.

3.4.1 Survey of AMP proportions in real proteomes

As can be seen from Table 3.2, AMPs only comprise a small proportion (0.5-2%) of the encoded proteins in the genomes of five well-studied taxa. To obtain these estimates I chose the top four organisms (the mammals *Mus musculus*, *Homo sapiens* and *Bos taurus* and the flowering plant *Arabidopsis thaliana*) by numbers of reviewed AMPs in Swiss-Prot (accessed February 2021), with the addition of the fruitfly *Drosophila melanogaster* as a well-studied invertebrate. For each organism, the proportion of AMPs was calculated by dividing the number of reviewed AMPs by the total number of reviewed proteins in the reference proteome for that organism. Overall, it is clear from Table 3.2 that AMPs only comprise less than 1% of the genome in animals, and slightly more in *A. thaliana* (~2%). The large number of AMPs in *A. thaliana* may be explained by the prevalence of cysteine-rich AMPs in this species (Tam *et al.*, 2015), and in other plants. The AMP proportions for all species shown in Table 3.2 are likely to be underestimates as there are many unreviewed proteins which may include AMPs. As the nearest rounded number for the AMP proportion in most species in this table is 0.01, this number was used as a representative measure for the purposes of exploring issues related to data imbalance.

Table 3.2: The number and proportion of reviewed AMPs in well-studied organisms in their respective proteomes.

Species	**AMPs in proteome**	**Reviewed proteins**	**Proporti on**	**Proteome ID**	**Unreviewed proteins**
Arabidopsis thaliana	291	15,961	0.0182	UP000006548	23,384
Mus musculus	100	17,058	0.0059	UP000000589	38,416
Homo sapiens	99	20,381	0.0049	UP000005640	55,395
Bos taurus	55	6,014	0.0091	UP000009136	31,499

Drosophila melanogaster	24	3,596	0.0067	UP000000803	18,521

3.4.2 Effect of dataset imbalance on test metrics

Despite the fact that AMPs comprise a minority of expressed proteins in a typical genome the issue of test dataset balance remains largely unexplored by the AMP prediction community. For example, the most recent and comprehensive review of AMP predictors evaluated the performance of 30 AMP predictors (Xu *et al.*, 2021) but used a balanced test set for all benchmarks. While balanced test sets have historically dominated the AMP prediction literature the potential for AMP predictors to be used in genome-scanning creates an imperative to account for imbalance in model evaluation (Whalen *et al.*, 2022). In this section I explore the issue of how class imbalance affects standard test metrics and use this to inform recommendations for the evaluating AMP predictors for genome-scanning.

Metrics used to test classification performance are generally based on the four confusion matrix categories, which comprise the true positives, false positives, true negatives and false negatives. Common metrics derived from these fundamental measurements include accuracy, specificity (or true negative rate), recall (also known as sensitivity or true positive rate), precision, and the Matthews Correlation Coefficient (MCC) (see equation 3.1). Of these, the MCC is the only metric that considers all four confusion matrix categories and subsequently only scores highly if a good result is achieved in all four of them. The MCC is considered a valuable metric in classification, as it is more informative in unbalanced datasets, and provides a more realistic sense of performance, compared to accuracy or F1 score, which can be misleadingly overconfident (Chicco and Jurman, 2020; Chicco, Tötsch and Jurman, 2021).

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$
$$Specificity = \frac{TN}{TN + FP}$$
$$Sensitivity = \frac{TP}{TP + FN}$$
$$Precision = \frac{TP}{TP + FP}$$
$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Equation 3.1: Formulae for commonly used performance metrics based on the confusion matrix. MCC: Matthews Correlation Coefficient, TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

Although AMP predictors are binary classifiers, they typically produce a score for each AMP reflecting an estimated probability that it belongs to the AMP class. This score, which I refer to as p (probability), is usually provided as output to the user, and provides the opportunity for users to adjust performance by choosing a different threshold value to the default (usually p=0.5). By choosing a different decision threshold the user will be making a different trade-off between elements of the confusion matrix. These trade-offs can be captured using either a ROC (Receiver Operating Characteristics) curve which plots the true positive rate (y) versus the false positive rate (x), or a Precision Recall curve which plots precision (y: Equation 3.1) versus recall (x: Equation 3.1). Both curve types can also be reduced to a single number by taking the area under the curve to produce the Area Under the Receiver Operating Characteristics curve (AUROC) and Area Under the Precision Recall Curve (AUPRC) metrics. The AUPRC is particularly useful as a performance metric for imbalanced datasets (Davis and Goadrich, 2006; He and Garcia, 2009; Sofaer, Hoeting and Jarnevich, 2018) as these focus on the proportion of actual true positives within the positive predictions (Saito and Rehmsmeier, 2015), rather than including the true negatives, as the false positive rates in the ROC curves do. While the AUPRC and AUROC do not depend on a chosen decision threshold (usually p=0.5), they still attempt to reduce performance of the predictor to a single value across the probability range (from 0 to 1), placing equal emphasis on all parts of this range.

To explore the effect of data imbalance on performance metrics I introduce the variable, α, which represents the proportion of AMPs in a test set. A confusion matrix generated under a balanced test set ($\alpha = 0.5$) can then be rescaled to expected values under an unbalanced dataset ($\alpha \neq 0.5$) using Equation 3.2.

$$TP_{\alpha} = TP_{0.5}\alpha$$
$$TN_{\alpha} = TN_{0.5}(1-\alpha)$$
$$FP_{\alpha} = FP_{0.5}(1-\alpha)$$
$$FN_{\alpha} = FN_{0.5}\alpha$$

Equation 3.2: Expected values of confusion matrix entries for a given value of α, scaled from values measured under a test set where the proportion of AMPs equal 0.5 (denoted by subscript 0.5).

This rescaling equation reveals some important characteristics of performance metrics. Firstly, it can now be shown that the receiver operating characteristic (ROC) curve and associated (AUROC) metric are invariant under α, and therefore less informative than other metrics when dataset imbalance is important (Davis and Goadrich, 2006). This can easily be seen by considering the two axes in a ROC curve (y = TPR, x = FPR), and the way that these scale with α (see Equation 3.3). These equations clearly show that neither the x or y axes of a ROC curve are affected by changes in α which implies that both ROC curves themselves and associated metrics (AUROC) are completely invariant with the dataset balance. In some situations, this may be desirable. However, for ‘omics-scanning where there is a strong requirement for high precision (which does vary with α) ROC-based metrics can provide a misleading estimate of performance.

$$ROC_y = TPR = \frac{TP}{TP + FN}$$
$$ROC_y^\alpha = \frac{\alpha TP}{\alpha TP + \alpha FN} = TPR = ROC_y$$
$$ROC_x = FPR = \frac{FP}{FP + TN}$$
$$ROC_x^\alpha = \frac{(1-\alpha)FP}{(1-\alpha)FP + (1-\alpha)TN} = FPR = ROC_x$$

Equation 3.3: Formulae for the x (FPR, false positive rate) and y (TPR, true positive rate) axis of a receiver operator characteristics (ROC) curve. Axes scaled by α are marked by a superscript α.

Using this same mathematical framework, we can now see that the relationship between precision and recall, and therefore PRC curves and the AUPRC are affected by α (see equation 3.4). This equation shows that recall is unaffected by α, whereas precision will decrease as α decreases.

$$Precision_\alpha = \frac{TP\alpha}{TP\alpha + FP(1-\alpha)}$$
$$Recall_\alpha = \frac{TP\alpha}{TP\alpha + FN\alpha} = \frac{TP}{TP + FN}$$

Equation 3.4: Recall and precision metrics for a given ɑ value. TP: true positives, FP: false positives, FN: false negatives

The effect of test set imbalance on precision-recall curves is shown graphically in Figure 3.4 based on results from ampir. These results were prediction results from the test set of ampir v.0.1 which contained 996 AMPs and non-AMPs (α = 0.5). To implement α, these test results were used as a foundation and expected values for different α were obtained via equations 3.2 and 3.4. Figure 3.4A indicates shifts of the trade-off between the precision and recall with varying proportions of AMPs in a genome over a range of predicted probability values. Note that recall (as expected) does not change, whereas as α decreases a much higher decision threshold is required in order to achieve high precision. Note that for a realistic α value (α = 0.01) the threshold required to achieve a precision of 0.5 is very high (>0.95). This highlights the fact that the relative rarity of AMPs makes them extremely challenging to identify with precision from whole proteome datasets. It also highlights the importance of probability values reported by predictors, as users who desire higher precision will use these values to implement an alternative decision threshold, reflecting their needs. In Figure 3.4B the impact of α on the trade-off between precision and recall can be observed. This highlights the challenge associated with small α. When the α value is 0.5, there is very little need for compromise, as both high precision and high recall can be achieved. In contrast, for α=0.01 it is impossible to achieve high precision without major sacrifices to recall.

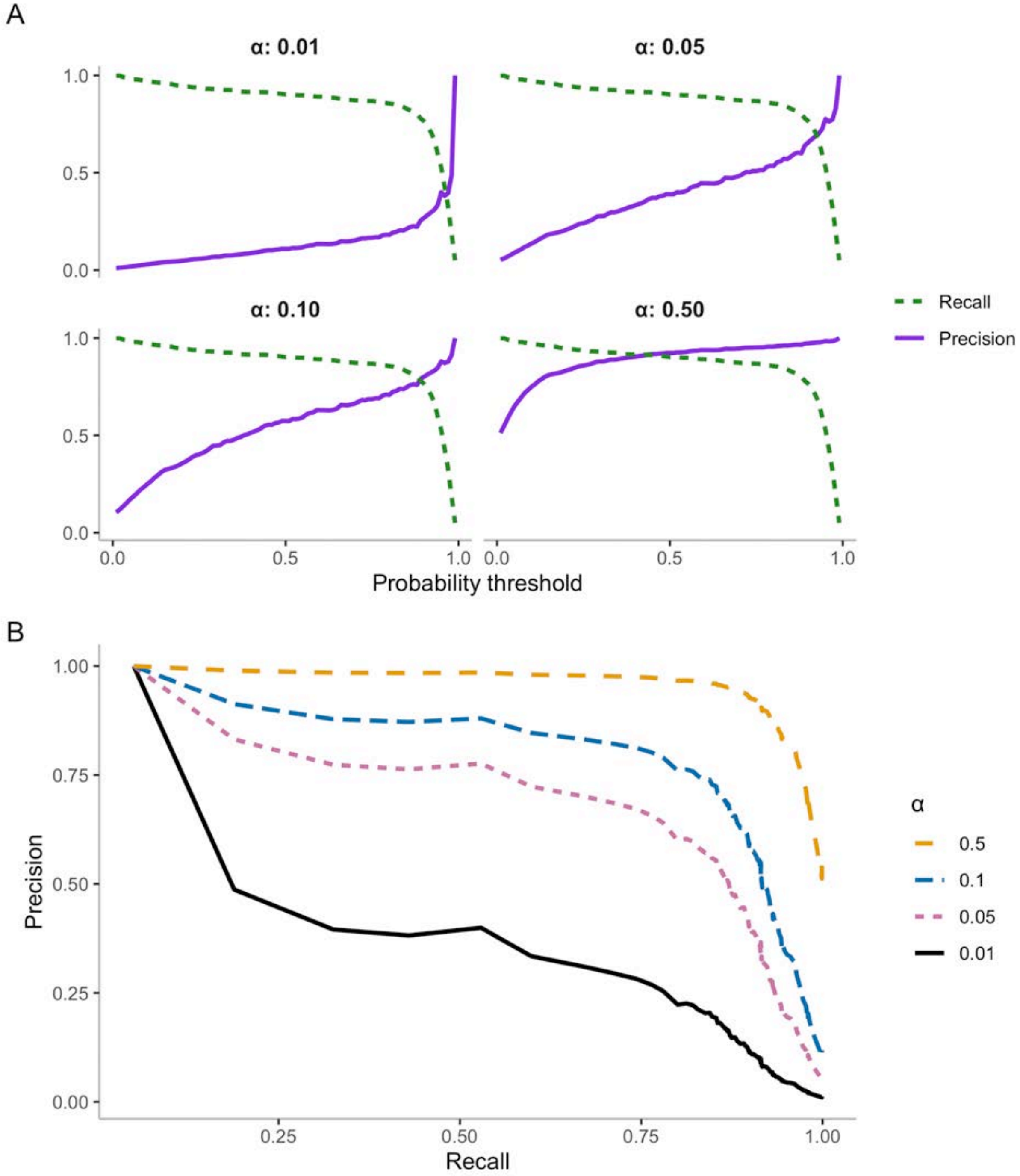


Figure 3.4: A) Calculated precision and recall metrics over predicted probability thresholds and B) A traditional precision-recall curve for four different α values that represent different proportions of AMPs in a genome.

3.5 Benchmarking AMP predictors in a genome-scanning context

To date, performance evaluations for AMP predictors have largely focussed on test datasets that do not reflect the types of inputs that would be used when scanning for AMPs in the genome of an organism, a use-case that I call genome-scanning. The use of balanced test sets is considered standard practice in the field. Even independent studies (i.e. those which did not develop their own AMP predictor) that have sought to benchmark different AMP predictors to determine the best performing predictor utilised a balanced test dataset (Gabere and Noble, 2017; Xu *et al.*, 2021). Similarly, additional reviews that focus on the availability and development of AMP predictors appear not to consider the implementation of AMP predictors on datasets that reflect a genome-scanning use case (Liu *et al.*, 2017; Aronica *et al.*, 2021; Ramazi *et al.*, 2022). This lack of emphasis on benchmarks appropriate for genome-scanning is surprising as this application is common in the AMP discovery field (see chapter 1). Although evaluating the real-world accuracy of AMP predictors is challenging, I demonstrate in this chapter that it is an issue that has serious consequences for the design and evaluation of AMP predictors.

The focus in this chapter is purely on a use case that is defined as “genome-scanning”, in which an AMP predictor is used as part of a novel AMP discovery pipeline to scan an entire ‘omics dataset (e.g. all predicted proteins) with the goal of producing a short list for experimental validation of AMP activity. An ideal realistic test set for ‘omics scanning applications would be the complete set of proteins across a broad range of taxa in which all of the AMPs and their precursor sequences have been correctly classified. Although no such datasets currently exist, a small number of highly studied organisms have proteomes that are thought to reflect translations from almost the entire complement of expressed genes, and for which a high proportion of AMPs have been classified based on close homology and/or experimental assays. In this section, the proteomes from *A. thaliana* and *H. sapiens* are used as representatives of realistic ‘omics scanning input data and subsequently used as a test set for multiple AMP predictors. These test sets have the appropriate balance, i.e. where α equals ~1-2%, and have little overlap with training data used by most predictors (see Figure 3.3).

3.5.1 Methods

The AMP predictors used in this section are outlined in Table 3.3. They include the AMP predictors amPEP (Bhadra *et al.*, 2018), amPEPpy (Lawrence *et al.*, 2020), deep-AmPEP (Yan *et al.*, 2020), AMP Scanner v2 (Veltri, Kamath and Shehu, 2018), AMPlify (Li *et al.*, 2020), AmpGram (Burdukiewicz *et al.*, 2020), ampir mature and precursor (Fingerhut *et al.*, 2020) to classify AMPs in the proteomes of *A. thaliana* and *H. sapiens*. However, the web server provided for deep-AmPEP contained a file size restriction of 1 megabyte. Considering the file size for a proteome is approximately 20 megabytes, deep-AmPEP was excluded from benchmarking analysis. The training data for the ampir precursor model includes precursor protein sequences for both *A. thaliana* and *H. sapiens.* In order to avoid calculating inflated performance metrics for ampir, a specific model, ampir_prec_nb, was created which excluded all *A. thaliana* and *H. sapiens* proteins. The *A. thaliana* and *H. sapiens* proteomes were preprocessed to comply with input requirements of all predictors. Specifically, all sequences that were shorter than 10 amino acids, or which contained non-standard amino acids, were removed. The majority of performance metrics were calculated using custom written functions in R. However, the area under the curve (AUC) for both the receiver operator characteristics (ROC) and precision recall (PR) curves, were calculated using the precrec, v. 0.12.7, R package (Saito and Rehmsmeier, 2017).

Table 3.3: AMP predictors used for proteome benchmarking (accessed January 2021). Further details on each model including the exact number of sequences present in each predictor's training and test set can be found in Table 3.1.

Predictor	**Test Dataset**	**Model algorithm**	**Accessed**	**Link**
amPEP	Balanced	Random forest	MATLAB source code	https://sourceforge.net/projects/

				axpep/
amPEPpy v.1.0	Balanced	Random forest	Python 3.8 via Conda v.4.10.0	https://github.com/tlawrence3/amPEPpy
deep-AmPEP30	Balanced	Convolutional neural network	Web server	https://cbbio.online/AxPEP/
AMP Scanner v. 2	Balanced	Convolutional and recurrent neural network	Web server	https://www.dveltri.com/ascan/v2/ascan.html
AMPlify v.1.0.0	Balanced	Deep learning with attention mechanisms	Python 3.8 via Conda v.4.10.0	https://github.com/bcgsc/AMPlify
AmpGram v.1.0	Balanced	Random forest	R 4.0.3	https://github.com/michbur/AmpGram
Ampir precursor v.1.1.0	Unbalanced	Support vector machine with radial kernel	R 4.0.3	https://github.com/Legana/ampir
Ampir mature v.1.1.0	Balanced			

3.5.2 Self-reported performance of predictors

The aforementioned predictors all reported excellent performance in their respective papers (see Table 3.4). Note that although most predictors evaluated their performance using at least some metrics that are sensitive to data imbalance such as the MCC and

AUPRC, the actual test data used was balanced (ampir_precursor is an exception, see Table 3.1). While this practice allows for a standardised comparison between predictors, the lack of testing with unbalanced data means that the relative performance of predictors listed in Table 3.4 is potentially not representative of how the predictors would perform on an unbalanced dataset.

Table 3.4: Performance evaluation results from existing AMP predictors on their respective test sets.

AMP models	**Acc**	**Sn**	**Sp**	**Prec**	**MCC**	**AUROC**	**AUPRC**
AmPEP	0.96	0.95	0.97		0.90	0.99	0.78
amPEPpy	0.84	0.85	0.82	0.83	0.68	*NA*	*NA*
deep-AmPEP30	0.77	0.77	0.78		0.54	0.85	0.85
AMP Scanner v2	0.91	0.90	0.92		0.82	0.96	NA
AMPlify	0.94	0.93	0.93	*NA*	*NA*	0.98	NA
AmpGram	*NA*	0.85	0.81	0.81	*NA*	0.91	NA
ampir mature	0.86	0.87	0.86	0.86	0.73	0.93	0.93
ampir precursor	0.97	0.73	0.99	0.90	0.79	0.97	0.86

Acc: accuracy, Sn: sensitivity, Sp: specificity, MCC: Matthew's correlation coefficient, AUROC: area under the receiver operating characteristics curve, AUPRC: area under the precision recall curve. Results were taken from each model's associated reference: amPEP (Bhadra *et al.*, 2018), amPEPpy (Lawrence *et al.*, 2020), deep-AmPEP (Yan *et al.*, 2020), AMP Scanner v2 (Veltri, Kamath and Shehu, 2018), AMPlify (Li *et al.*, 2020)

AmpGram (Burdukiewicz *et al.*, 2020), ampir mature and precursor (Fingerhut *et al.*, 2020).

3.5.3 AMP predictor performance on *H. sapiens* and *A. thaliana* test sets

Figure 3.5 shows both the ROC curves (A) and the PR curves (B) for the prediction results from various AMP predictors on the *H. sapiens* and *A. thaliana* proteomes. On a ROC curve plot, an AUC of 0.5 makes a diagonal line from the bottom left to the top right corner. The perfect PR curve is like a mirror image of the perfect ROC curve; it bends at the top right corner, which refers to the model performing with 100% recall and precision. Therefore, the more the PR curves bend toward the top right corner, the better the model is. When comparing multiple curves on the same plot, the curve that is above another curve, is generally assumed to reflect better performance. It is clear that the ROC curves overall show a better performance compared to the PR curves which corroborate the findings of Davis and Goadrich (2006) and Saito and Rehmsmeier (2015) that the AUROC can be misleadingly confident on imbalanced datasets. The AUC values for the ROC curves range between 0.16 - 0.99 for *A. thaliana* and 0.43 - 0.94 for *H. sapiens.* The AUC values for the PR curves are between 0.004 - 0.83 for *A. thaliana* and 0.001 - 0.30 for *H. sapiens*, which are overall much lower compared to the AUROC values. The overall poor performance of amPEP and amPEPy could be attributed to the high proportion of mature peptides in their respective training datasets. This is reflected in Figure 2.9, which shows amPEP performs well on ampir's mature test set, but poorly on the ampir's precursor test set. Interestingly, all models, with exception to amPEP, had higher AUC values for both ROC and PR curves for *A. thaliana* (Figure 3.5). This may indicate the models were better at detecting AMPs in *A. thaliana* compared to *H. sapiens*. The ampir precursor model had the highest AUPRC value on the *A. thaliana* proteome (AUPRC: 0.83). However, the remaining AUPRC values (for both proteomes) are below 0.35. Therefore, according to the AUPRC metric, none of the models (save perhaps the ampir precursor model on *A. thaliana*) performed well in detecting AMPs in the *H. sapiens* and *A. thaliana* proteomes.

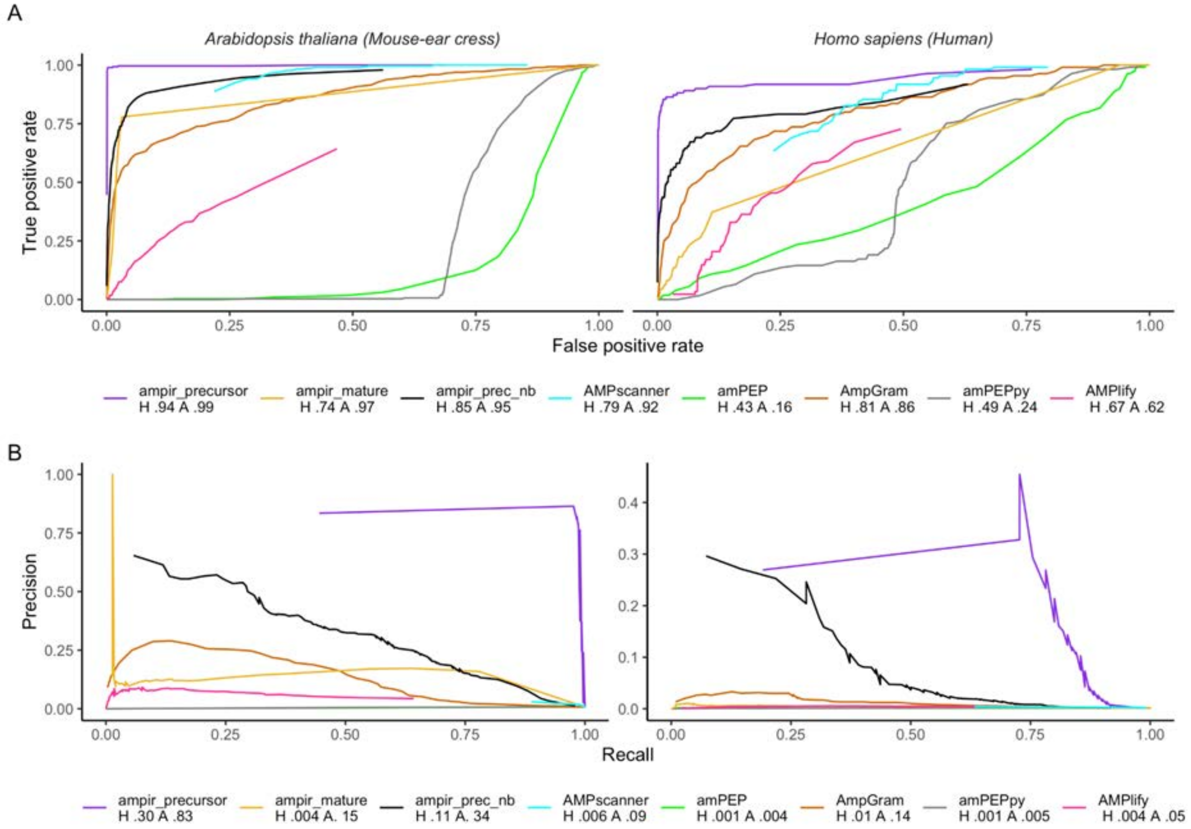


Figure 3.5: Performance of various AMP predictors in classifying whole proteome data for *Homo sapiens* and *Arabidopsis thaliana*. Performance is shown as ROC curves (A) and precision-recall (PR) curves (B). H refers to *Homo sapiens* and A refers to *Arabidopsis thaliana*. The numbers that follow are the respective AUC values for either the ROC or PR curves.

The AUPRC, AUROC, MCC, precision and recall metrics were used to assess the performance of AMP predictors in both proteomes (Figure 3.6). A wider range of performance metrics (including the accuracy, specificity and F1 score) can be found in Table S3.1. From these metrics, the AUPRC and MCC are most informative in imbalanced datasets. Other metrics, such as the AUROC, may be too misleadingly optimistic (Saito and Rehmsmeier, 2015; Davis and Goadrich, 2006) which is apparent in Figure 3.6, as the AUROC values overall are much higher compared to the AUPRC and MCC. Although the AUROC and AUPRC capture performance across the entire range of decision thresholds the values for precision, recall and MCC were calculated using a default decision threshold of p=0.5. With the exception of amPEP, amPEPpy and AMPlify, the AMP predictors were able to find a large proportion of the AMPs (recall > 0.56) present in both proteomes, which likely reflects the decision threshold chosen (p=0.5). However, only ampir_precursor and ampir_precursor_nb were able to classify the AMPs with any precision at this decision threshold in *A. thaliana* (precision = 0.38 and 0.28, respectively). In contrast, the precision values for other predictors and for *H. sapiens* were all less than 0.05 indicating that a much higher decision threshold is likely to be required in practical use-cases. For example, if the goal is to produce a list of candidate AMPs for experimental testing it would be a vast waste of resources to obtain a candidate list containing less than 5% true positives.

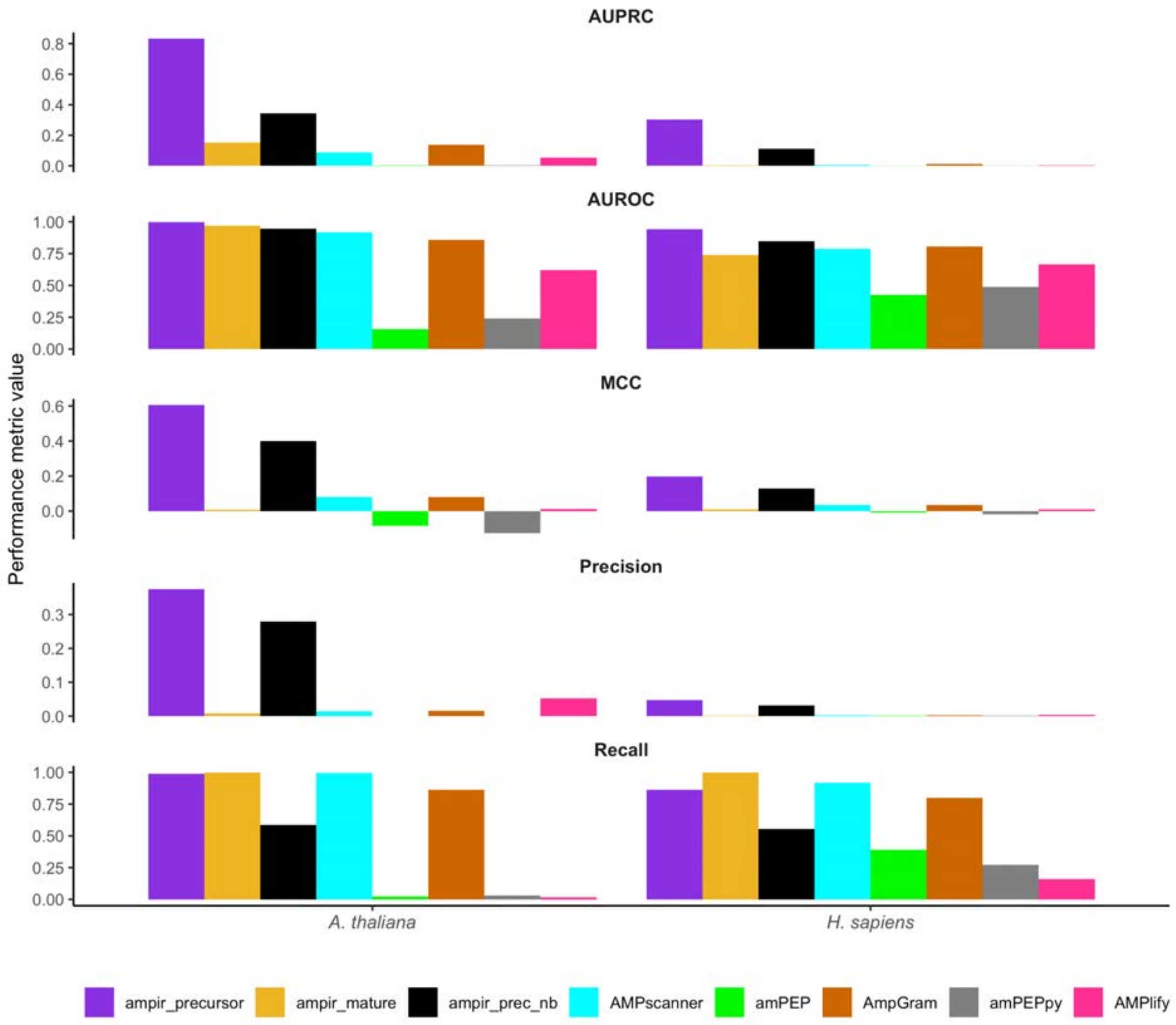


Figure 3.6: Performance of AMP predictors on the proteomes of *Arabidopsis thaliana* and *Homo sapiens* using various performance metrics.

Performance metrics evaluated at a decision threshold of p=0.5, or that capture the entire range of thresholds (e.g. AUPRC and AUROC) may not be very informative when the goal is genome-scanning. Consider the situation in which the goal is to obtain a list of candidates for experimental testing. Recent publications that do this usually synthesise and experimentally test between 10 and several hundred candidate peptides, with several hundred representing a very large investment in time and money (Ma *et al.*, 2022). Precision is therefore by far the most important metric in these cases because it is essential that the candidate lists are rich in true positives (AMPs). To adequately express the real-world performance of predictors in this situation the trade-off between numbers of true and false positives was shown as these are the confusion matrix entries that contribute to precision. In addition, the focus is on the low false positive regime as this matters most in whole proteome scans (Figure 3.7). In these plots the number of false positives represent wasted effort and true positives represent the goal. Importantly, the x-axis is restricted to only show values less than 500 since it is not feasible to test larger numbers of candidate proteins. This range restriction removes some predictors entirely, notably AMP Scanner which was unable to predict less than 500 false positives at any decision threshold (see Figure S3.2). Figure 3.7 shows that the ampir_precursor and ampir_precursor_nb models clearly outperform all other predictors in this regime which likely reflects the fact that the ampir precursor models are the only predictors to incorporate large numbers of precursor proteins in their training data.

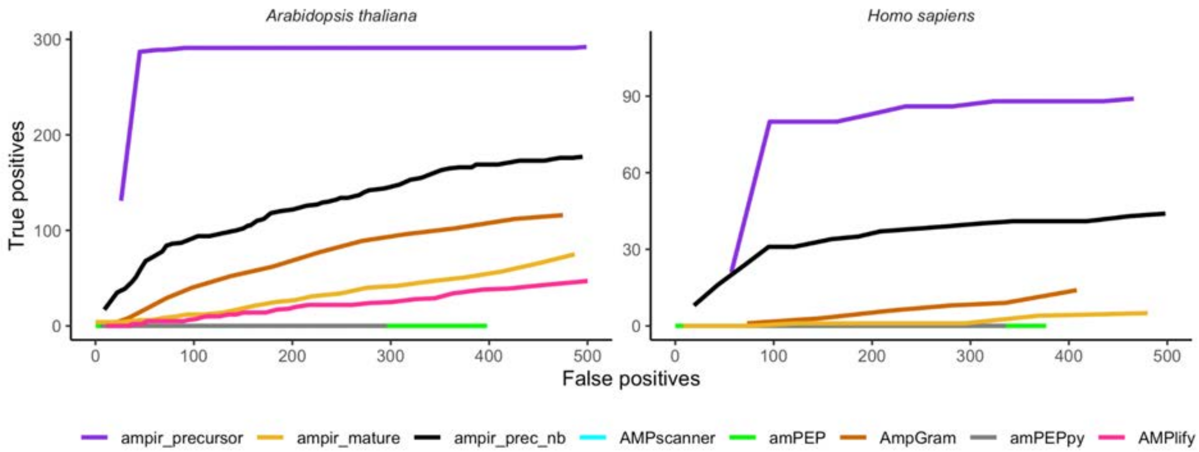


Figure 3.7: The ability of eight AMP prediction models to predict AMPs in the low false positive regime (<500) in the proteomes of *Arabidopsis thaliana* and *Homo sapiens*. It is scaled so that the limits of the y-axis show the full complement of known AMPs in each genome (291 for *A. thaliana*, 110 for *H. sapiens*), and the limits of the x-axis are restricted to emphasise behaviour in the low false positive (FP) regime (FP < 500) because this is most relevant in whole genome scans.

3.4 Discussion

In this chapter I found that the reported performance for published AMP prediction models is generally much higher than their performance on realistic input datasets. This largely reflects the practice of hold-back testing which uses test data compositionally similar to the training data to assess model performance. While hold-back testing works well in situations where the training data is unbiased compared with real-world usage, I found that such biases were prevalent in AMP datasets and that they have a dramatic effect on real-world performance.

In order to characterise these biases, I adopted the novel approach of using complete proteomes from well-studied organisms as representative of real-world datasets. This approach has its own problems (see limitations section below) but enabled me to identify major sources of bias in most AMP datasets used to train and test predictors. Specifically, AMP training and testing data for several widely used models have length distributions far from those seen in the proteomes of *H. sapiens* and *A. thaliana*, which largely reflects differences in filtering strategies on positive and negative datasets. In addition, training and testing data for most predictors was balanced (equal numbers of positive and negative cases), but a survey of AMPs in proteomes of well-studied organisms indicates that the positive fraction is usually less than 1%.

One of the goals in creating AMP prediction models is to reduce the number of false positives obtained from prediction results (Meher *et al.*, 2017). One way to decrease the number of false positives is to maximise precision, by adjusting the decision threshold to a higher value. The results from this chapter illustrate the fact that as the proportion of AMPs in the input dataset (α) approaches a realistic value for ‘omics scanning (~ 1%), it often becomes necessary to choose between precision or recall as it is impossible to maintain acceptable values for both (equation 3.4).

Furthermore, I found that AMP predictor performance in the low false positive regime is essential, as this regime represents the behaviour of the model at decision thresholds most likely to be used for AMP scanning applications. Importantly, this is not entirely captured by the common evaluation metrics used. I explicitly showed performance in the low false positive regime by plotting true versus false positives over a restricted range of false positives. It would be a useful future direction to devise a statistical metric to capture this. One metric that could be explored for AMP prediction is the partial AUC, which is the AUC for a specific region of the ROC curve (Dodd and Pepe, 2003). Other, balanced, metrics such as MCC go some way to achieving this, however, it is necessary to choose a decision threshold at which to evaluate these. For genome scanning the number that is most relevant is the expected yield of true positives among the top X highest ranked candidate AMPs. In this case the difficulty lies in setting X which represents the number of AMPs that is feasible to synthesise and screen for activity. Here, I surveyed values of X ranging from 0-500, which I call the false positive regime. Future work might consider citing numbers such as Y10, Y100, Y1000 representing the expected yield of AMPs where X is 10, 100, and 1000 respectively.

Another thing I observed in this chapter was the prediction result differences between *A. thaliana* and *H. sapiens*. Overall, AMP predictors appeared to exhibit higher performance in *A. thaliana* compared to *H. sapiens*. Although this could reflect the substantially higher number of AMPs present in *A. thaliana* (see Table 3.2), there may be compositional differences in AMPs between the two contrasting taxa, animals and plants, that affect the AMP predictor. This could potentially reflect the composition of the AMP databases, which are used as a fundamental data source in the training dataset of machine learning AMP models. These AMP databases are taxonomically biased, reflected by the majority of known AMPs sourced from certain well-studied taxa, such as amphibians (Wang, Li and Wang, 2016). This taxonomic bias is another source of compositional bias in the training data which may affect the prediction accuracy. However, the contribution of taxonomic bias is more difficult to assess compared to the contributions of the other two factors that contribute to compositional bias, i.e., the AMP to non-AMP balance and precursor versus mature peptide composition. Investigating taxonomic bias is challenging and requires the application of dedicated methods, which are addressed in chapter 4.

To adequately assess the performance of an AMP predictor, it should be tested on real ‘omics datasets. Proteomes are a possible option, as these contain a complete collection of predicted proteins potentially present in organisms. Ideally, proteomes used as test sets should be high quality and annotated with the full complement of

AMPs present in that organism. However, even in well-studied organisms in which a large number (~100) of AMPs have been identified (e.g. *A. thaliana* and *H. sapiens*), this criterion is not met. In fact, for the majority of organisms, including well-studied model organisms such as *Drosophila melanogaster*, substantially fewer AMPs are known (~25) than are likely to be present. Subsequently, there are many prospective AMPs left undiscovered in most species. Therefore, proteomes for many species are likely not adequately annotated, which severely restricts the options currently available for AMP testing.

3.5 Conclusions and Recommendations

This chapter examined the ability of recently published AMP predictors to predict AMPs in proteomes. Most AMP predictors performed poorly in this regard, mostly due to biases in the training data. AMP predictor performance can be increased by performing model training exclusively on precursor proteins. Furthermore, it was discovered that, when using proteome data, the performance of AMP predictors generally fell far short of reported performance metrics. To accurately test the performance of a predictor on a genome-wide scale, the predictor should consider several aspects. First, the actual true proportion of AMPs in a genome is low (~1%). Therefore, performance of AMP predictors on imbalanced data should be included. Additionally, metrics to measure the performance should be robust to imbalanced data, and also reflect the purpose of the task (i.e. to identify correctly classified AMPs). Second, a test set should be included which most closely resembles a genome, to simulate the most realistic scenario in which to test the predictor. This chapter described issues with the composition of training and testing data of AMP predictors. Addressing these issues will improve the performance of AMP predictors in real world situations.

Chapter 4: When are machine learning AMP predictors better than homology for AMP detection in genomes?

4.1 Abstract

The last decade has seen a proliferation in the number and variety of machine or deep learning methods developed for classifying peptides or proteins according to their antimicrobial activity. The inherent justification of these methods is that they must offer some advantage over homology-based searches which are widely used for the same purpose. One potential reason for this is that antimicrobial peptides (AMPs) may exhibit some convergence towards common physicochemical properties independent of their amino acid sequence, and the other is that they are often under strong positive selection, leading to a high sequence diversification rate making sequence similarity searches less effective at greater taxonomic distances. However, little effort has been made to empirically test these assumed advantages. In this chapter the effectiveness of machine learning models and homology-based searches to find AMPs in a variety of organisms was compared. A new metric was devised to measure the degree to which an organism is represented by AMPs from closely related organisms in an AMP database. This metric was used to determine the effect of taxonomic distance on performance of both machine learning and homology methods. It was found that while the performance of homology-based searches declined significantly with taxonomic distance, this was not the case for ampir, a machine learning based AMP predictor. This suggests that machine learning models are indeed more effective at finding AMPs in taxonomically distant organisms than homology-based searches. This study validates the effectiveness of machine learning models to find novel AMPs on a wide taxonomic scale.

4.2 Introduction

Machine learning is frequently employed as a method to discover novel antimicrobial peptides (AMPs). More than 30 machine learning AMP predictors have been developed for this very purpose (Xu *et al.*, 2021). In most papers describing new AMP predictors, or in benchmarks across existing predictors, there are no justification statements as to why a machine learning method is preferred over homology-based searches, or sequence similarity. This is despite the fact that sequence similarity implemented in tools like the basic local alignment search tool (BLAST) (Altschul *et al.*, 1990) remains a common approach to identifying new AMPs as candidates for experimental verification of AMP activity. For example, recent experimentally verified AMPs in a range of animal groups were found using the sequence similarity approach, including in an annelid (Panteleev *et al.*, 2020), crustacean (Du *et al.*, 2019), gastropod (Hayashida and da Silva Junior, 2021), bird (Xiao *et al.*, 2020) and in amphibians (Li *et al.*, 2019; Chai *et al.*, 2021; Liscano *et al.*, 2021), arachnids (Li *et al.*, 2021a; He *et al.*, 2021), fish (Dong *et al.*, 2021; Shen *et al.*, 2021; Zhuang *et al.*, 2021; Zhang *et al.*, 2022), insects (Lee *et al.*, 2020a; Lee *et al.*, 2020b; Lee *et al.*, 2021b; Lee *et al.*, 2021a), and mammals (Li *et al.*, 2021b; Peel *et al.*, 2021).

One advantage machine learning methods may have over sequence similarity, or homology alone, is that they are more flexible. This is because there are many statistical approaches and algorithms, each of which can be optimised for specific use cases by selecting and modifying the training databases, as well as via selecting the most relevant features. AMP predictors are generally trained on physicochemical and compositional properties of proteins, most of which can be calculated directly from the amino acid sequence. These features attempt to capture the structure and function of the protein and in turn, its properties that reflect antimicrobial activity (Lata, Sharma and Raghava, 2007; Lata, Mishra and Raghava, 2010; Thomas *et al.*, 2010; Torrent *et al.*, 2011; Porto, Pires and Franco, 2012; Meher *et al.*, 2017; Bhadra *et al.*, 2018; Veltri, Kamath and Shehu, 2018). The process relies on AMPs having a pattern of physicochemical properties that is distinct from non-AMPs, which the algorithm is able to use to differentiate between the two different classes. Although in theory it is possible to train models using any informative feature, most models so far use a very similar set of features. This might reflect constraints arising from limited information that can currently be gleaned from the amino acid sequence alone. Another major constraint that limits the effectiveness of the machine learning approach is the availability of high-quality training data. This lack of high-quality training data results in AMP prediction models often using subsets of the same or similar training databases because at the present time there are few choices available.

Sequence similarity searches, generally performed with BLAST, compare a query sequence to a (potentially very large) database and identify likely homologs on the basis of sequence similarity. This approach could be viewed as a very simple form of machine learning as it uses a nearest neighbour classifier. However, for the purpose of this chapter, BLAST is considered to be distinct from machine learning. Since the BLAST approach explicitly captures the order and relative conservation weighting of amino acids in the sequence, it might perform better than most machine learning approaches which focus on summary statistics that can be derived from the sequence rather than the sequence itself. When applying homology searches for AMP prediction, the underlying assumption is that since homologous sequences are often also similar in structure and function, homology can be used to infer antimicrobial activity. The weakness in this approach however, is that it becomes more difficult to do this when the sequences being compared are far apart on an evolutionary scale (Pearson, 2013). This can be summarised as the two key principles of homology-based functional assignment: (a) correctly inferring a common function based on homology, and (b) identifying genuine homologues. Both principles break down as taxonomic distance increases. This is partly because (principle a) when sequences diverge, they can acquire different functions (Sangar *et al.*, 2007). Even in well-studied organisms AMP gene families have been shown to rapidly expand which results in greater diversification (Innan and Kondrashov, 2010; Lazzaro, Zasloff and Rolff, 2020).

In addition, in the context of finding AMPs, their fast evolutionary rate and small size make it difficult to identify sequence homologues (principle b) in all but the most closely related organisms (Ohtsuka and Inagaki, 2020). This is because short sequences (like AMPs) provide limited information with which to infer homology resulting in poor performance of most established methods (Santos-Júnior *et al.*, 2020). Aside from their small size and rapid evolutionary rate, AMPs can also arise via convergent evolution (Unckless and Lazzaro, 2016; Lazzaro, Zasloff and Rolff, 2020) which further limits the effectiveness of homology-based searches. Evidence for this comes from several authors who noted that AMPs have little sequence homology, regardless of their shared function, making it difficult to find AMPs using sequence similarity methods (Lata, Sharma and Raghava, 2007; Lata, Mishra and Raghava, 2010; Khosravian *et al.*, 2013). It has been suggested that this lack of sequence similarity among AMPs could indicate AMPs frequently arise via convergent evolution, to optimise their effectiveness against local microbes in the environment in which they are created (Hancock and Chapple, 1999).

A striking feature of the literature on AMP discovery is the general reliance on homology-based searches to produce candidates for experimental verification (Du *et al.*, 2019; Panteleev *et al.*, 2020; Xiao *et al.*, 2020; Chai *et al.*, 2021; Zhang *et al.*, 2022) while at the same time machine learning approaches have proliferated (Xu *et al.*, 2021), presumably under the assumption that these offer advantages as a replacement for, or complement to homology. Despite this, there has been little effort to compare homology-based searches to machine learning methods to find AMPs. From the 39 AMP prediction papers surveyed, only Santos-Júnior *et al.* (2020) and Wang *et al.* (2011) investigated the effectiveness of both methods. Santos-Júnior *et al.* (2020) benchmarked BLAST against their own machine learning based AMP predictor, Macrel, as well two other machine learning based AMP predictors, AMP Scanner (Veltri, Kamath and Shehu, 2018) and iAMP-2L (Xiao *et al.*, 2013) using a test set containing 500 AMPs and 500 non-AMPs which had homologous sequences with a sequence identity of 80% or higher removed. They found the effectiveness of BLAST to be similar to random, and concluded that, aside from immediate homologs, homology is not suitable to classify AMPs. In contrast, Wang *et al.* (2011) found BLAST to be an accurate method for AMP prediction, and even outperformed their machine learning method. The methods were tested on two datasets: one containing 2,752 AMPs and 10,014 non-AMPs, and the other dataset was a subset, where sequences with a sequence identity higher than 70% were removed. Wang *et al.* (2011) stated that even though BLAST had a better overall performance, its performance decreased by approximately 10% on the 70% dataset, suggesting that the presence of close homologs increased the predictive capacity of BLAST.

As machine learning methods do not solely rely on the principles of homology-based functional inference, and target function more directly (via features that reflect this), they are less restricted and more likely to be successful at finding novel proteins especially in cases of convergent evolution or high taxonomic distance. However, the performance of machine learning models is reliant on the training data and at the present time the majority of peptides with experimentally verified AMP activity (positive training cases) are largely restricted to a narrow range of taxonomic groups such as mammals and amphibians (Wang, Li and Wang, 2016). Likewise, homology-based methods are impacted by the same issue since these experimentally verified AMPs represent the set of query sequences that can be used for search.

Therefore, it is expected to be more difficult to find AMPs in taxa that are distant from the taxonomic majority (taxa where the majority of AMPs have been identified) irrespective of the method used. Our hypothesis however is that machine learning models should outperform sequence similarity searches to find these more distant AMPs. To test this, the performance of a machine learning model was compared with sequence similarity (BLAST) searches to find AMPs in a range of organisms. Importantly, these organisms were selected to capture a range of taxonomic distances allowing an assessment of relative degradation in performance along a taxonomic distance scale.

4.3 Methods

All analyses were completed in R version 4.1.2 (R Core Team, 2021), unless stated otherwise, using the RStudio integrated development environment, version 2022.02.0+443 (RStudio Team, 2021) and the tidyverse R package, version 1.3.1 (Wickham *et al.*, 2019).

4.3.1 Construction of AMP databases

The UniProt database (UniProt Consortium, 2021) (accessed 07 July 2021) was used as the fundamental source for the creation of the different databases. The UniProt AMP database was created by selecting all proteins that contained the UniProt keyword "Antimicrobial [KW-0929]" which resulted in 45,497 AMPs. The Swiss-Prot AMP database was a subset of the UniProt database, which contained all reviewed proteins annotated with the "Antimicrobial" keyword, 3,320 AMPs. The final AMP database was more stringent, and it was created using a similar approach to how the positive dataset for ampir was created (see chapter 2.3.1): the UniProt database was used as a starting point and all reviewed AMPs from UniProt and the unreviewed AMPs (found via the "Antimicrobial" keyword), providing the unreviewed AMPs overlapped with the APD3 (Wang, Li and Wang, 2016), DRAMP (Kang *et al.*, 2019) or dbAMP (Jhong *et al.*, 2019) specialised AMP databases (accessed 09 April 2021), which resulted in a total of 3,412 AMPs. This database is henceforth referred to as the AMP database.

4.3.2 Organism selection

Multiple factors were considered in the selection of the organisms. The first factor was that organisms should span a wide taxonomic range. The second factor was that organisms should have enough known AMPs to allow calculation of an empirical false discovery rate. A threshold of 10 AMPs was set for this criterion. The final factor was that organisms should have a designated reference proteome within UniProt (https://www.uniprot.org/proteomes). Therefore, the organism which had the most AMPs within each taxonomic order was considered, providing the organism had more than 10 AMPs and a proteome. The predominant phyla used were Chordata, Arthropoda and Streptophyta, as these were the only phyla that contained multiple organisms with more than 10 AMPs and a proteome. To provide a most distant point in the wide taxonomic range, the bacterium with the most known AMPs was also included. The reference proteomes for the selected organisms were obtained from UniProt proteomes (accessed August 2021). The proteomes were downloaded with one protein sequence per gene to remove potential duplicate protein sequences. See Figure 4.1 for a simplified representation of the construction of AMP databases and organism selection.

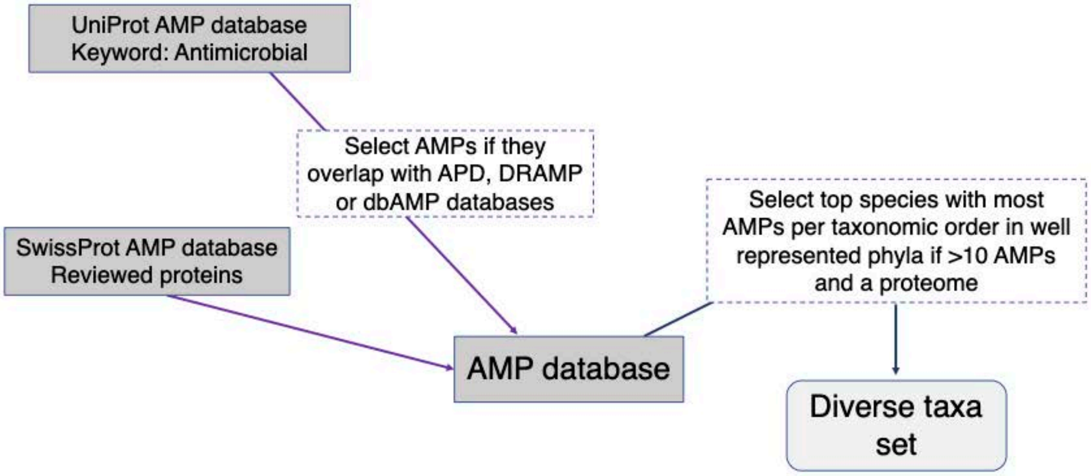

UniProt AMP database
Keyword: Antimicrobial
Select AMPs if they overlap with APD, DRAMP or dbAMP databases
SwissProt AMP database
Reviewed proteins
AMP database
Select top species with most AMPs per taxonomic order in well represented phyla if >10 AMPs and a proteome
Diverse taxa set

Figure 4.1: Simplified diagram of database construction and organism selection.

4.3.3 Test sets creation

The proteomes for the chosen organisms were used as test sets. However, there are likely many uncharacterised AMPs in the proteomes, especially for organisms with less complete AMP coverage. A higher false negative rate in these organisms is therefore expected. Unfortunately this remains an unsolved problem in this field. However, proteomes reflect a real use scenario for AMP discovery finding methods, which is why they were used as test sets in this study. The proteins in the proteomes were labelled as an AMP if these proteins overlapped with the AMP database. Additionally, the mature AMPs that were present in the AMP database for each organism, were matched to the organism’s respective proteome (which mostly contains precursor proteins) and manually annotated as AMP. This AMP labelling method was developed to reduce the chance of possible false positives, where AMPs may have been annotated as an AMP by the UniProt keyword “Antimicrobial [KW-0929]”, but where there was no experimental evidence for this protein to contain antimicrobial activity.

4.3.4 Machine learning model construction and BLAST searches

The aim was to compare the AMP classification performance of machine learning models versus homology. These two methods were implemented using custom training and test sets for each selected organism, designed to simulate a situation where a new organism was sequenced, for which the AMP complement was unknown. To construct an AMP machine learning model, a training dataset was constructed for each selected organism which contained the AMPs in the AMP database as a positive dataset, and the Swiss-Prot protein sequences not annotated as "Antimicrobial" as a negative dataset. To avoid bias, the given organism was excluded for both the positive and negative datasets, prior to construction. As the objective was to discover AMPs based on their full-length sequence (rather than the mature peptide), the same approach as the R package ampir (Fingerhut *et al.*, 2020) was used to create its 'precursor' model. This approach is designed to remove mature peptide entries from the training database as was implemented as outlined in chapter 2.3.1.2. This resulted in a training pool of 1,635 AMPs (positive set) and 250,897 non-AMPs (negative set). The final positive and negative dataset was then constructed for each organism by first removing sequences of the target organism from both sets. Finally, the negative dataset was randomly subsampled so that it contained 10 times the number of sequences present in the positive dataset. A support vector machine with radial kernel (rSVM) was used as a training algorithm to create each machine learning model with the R package caret, version 6.0.88 (Kuhn, 2019) in R version 3.6.1. Data were centred and scaled as a preprocessing method prior to training. The training details were as follows: class weights were used to prevent the major class being overemphasised, model hyperparameters (C, sigma) were tuned using a grid search of values and best values were determined by 10-fold cross validation, class probabilities were used to measure performance, optimised with the Kappa metric. The resulting final model, for each given organism, was used to predict AMPs in the given organism's respective proteome with ampir, version 1.1.0.

To find AMPs based on sequence similarity, blastp, version 2.11.0, from BLAST+ was used (Camacho *et al.*, 2009). Similar to the machine learning method, a different query dataset was created for each organism which contained all AMPs of the AMP database, excluding the given organism. Thus the query set for this blastp analyses was identical to the positive training set used in the machine learning analyses. This AMP query dataset was used to search a local BLAST database, constructed from the complete proteome of the given organism. Thus the search database for blastp analyses was equivalent to the set of proteins used as input to the machine learning analyses. The expectation (E) value threshold was left to the default value of 10 so that only very poor matches were removed. This resulted in raw outputs that could later be filtered based on bit-score to analyse the performance of the model at different thresholds (see below). Output was saved in a tabular format, retaining the top five matches per sequence.

To make both the machine learning and BLAST methods directly comparable, the results for these methods were matched back to the proteomes of each organism to create a table with the following information for each entry in the proteome: an AMP probability prediction score, a bit-score value, and antimicrobial status (AMP or non-AMP). The bit-score was used instead of the E-value as a statistical measure of sequence similarity because it is not dependent on database size (Xiong, 2006). For the antimicrobial status column, an entry was considered to be a true AMP if it was also present in the AMP database (see the previous data selection section for details). These labels and AMP scores obtained from the machine learning and BLAST methods were used to construct precision-recall (PR) curves as a method to compare the two AMP finding methods. The area under the PR curves (AUPRC) values were calculated as a summary performance metric using the precrec, version 0.12.7, R package (Saito and Rehmsmeier, 2017).

To test the effectiveness of both AMP finding methods as a function of taxonomic distance, a taxonomic representation score (see section 4.3.5) was calculated for each organism and this was plotted against the AUPRC values for each method. A Spearman’s rank order correlation test was used to test for a significant relationship between taxonomic representation score and AUPRC.

4.3.5 Taxonomic representation score

Part of the aim of this study was to observe how the performance of AMP finding methods changes with taxonomic distance. To achieve this aim, a score was devised that represents the evolutionary distance between AMPs in the AMP database and AMPs in a given target organism. The initial step was to calculate the taxonomic distance between all pairs of organisms present in the AMP database. This was achieved by extracting the names of all 788 organisms that made up the 3,304 AMPs in the AMP database which were then uploaded to the TimeTree server (accessed August 2021), http://www.timetree.org/, (Kumar *et al.*, 2017b), to obtain molecular time estimates of divergence. Viruses and species that did not contain a binomial name were excluded as these were not recognised by TimeTree. In addition, not all species were represented in the TimeTree database and some species were known under a different name, resulting in 221 unresolved names. The resulting timetree was exported as a Newick file and read back into R using the ape, version 5.6.1, R package (Paradis and Schliep, 2019). Out of the 221 unresolved names, 81 were replaced by TimeTree to a different name. To match the names back to the AMP database, these organisms were identified and renamed back to their original name. The corrected names were exported back to tree format using the treeio, version 1.16.2, R package (Wang *et al.*, 2020b). The remaining 140 organisms were not found by TimeTree and were likely not present in their database. These organisms primarily included anurans, arachnids and hymenopterans. The cophenetic pairwise distances, which in this context are equal to double the time to the most recent common ancestor, between organisms were then calculated with the ape package and matched back to the AMPs in the AMP database via the known relationship between each AMP and its organism of origin.

The second step was to use the distances to construct a representation score for each target organism, to determine how closely related the AMPs in the AMP database are to that organism. To do this, the distances needed to be converted into a score that decreases as a function of distance. Initially the inverse distance was used (1/distance), however, this has the property that it goes to infinity at a small distance value, which resulted in distorted high scores for well represented taxa. It was expected that there should be some taxonomic distance threshold within which two homologous AMP sequences will be highly likely to retain the same or similar function (i.e. remain an AMP). The goal was to create a score that reflects the number of AMPs in the database within this distance for a target organism. A sigmoid curve is a good choice for this because it is relatively flat with a value close to 1 out to a point, before smoothly transitioning to 0. Under this system the dominant contribution to the score for most organisms will be AMPs that are identified from organisms that are taxonomically close to it. One difficulty however, is the choice of taxonomic distance at which to set the threshold, which appears as the term s in Equation 4.1.

$$\Theta_t = \sum_{i \neq t} \theta_i n_i$$

$$\theta_i = \frac{s}{s + e^{\frac{d_i}{s}}}$$

Equation 4.1: The taxonomic representation score of AMPs (Θ_t) based on a sigmoid function (θ_i) where s is a parameter controlling the shape of a sigmoid curve that determines the relative weighting of AMPs given their taxonomic distance from the target organism, t, n_i is the number of known AMPs of species i, which have the same taxonomic distance, d_i, to the target organism.

The units of s are therefore the same as taxonomic distance which essentially makes it a cut-off divergence time point where AMPs might become more findable by homology and potentially more functionally similar. There is no completely objective way to choose s as it depends on the target organisms and organisms present within the AMP database. Therefore, all the results shown should be interpreted under the clear understanding that these are valid for a particular value of s.

To choose an appropriate value for s, the relationship between taxonomic distance and AMP database composition was visualised in several ways. First the distribution of taxonomic distances of the selected organisms were examined to determine how they might contribute to each other's scores for various potential values of s. Next, the effect of different values of s on the taxonomic representation scores of the selected organisms was examined to ensure differentiation between organisms could be observed. This detection of differentiation between organisms was important as the goal was to highlight the differences between machine learning and homology-based AMP finding methods. The chosen value of s was then used to calculate the taxonomic representation score of the selected organisms.

To examine how both the chosen value of s, and the composition of the AMP database affect the taxonomic representation score, the AMP contributions of different species to the taxonomic representation score were investigated. First the top two contributing species to all selected organisms were examined for a broad view. Second, for a more detailed view, the top 10 contributing mammals to the taxonomic representation score of the mammals were selected. The phylogenetic relationships among these selected organisms were then obtained from TimeTree, exported as a Newick file and visualised and annotated with the ggtree, version 3.0.4, (Yu *et al.*, 2016) R package.

4.3.5.1 Explainer: calculation of the taxonomic representation score

The taxonomic score reflects how well a training dataset of AMPs in different organisms represent the target organism of interest. Organisms that have a high taxonomic representation score, will be closely related (taxonomically) to organisms that contribute large numbers of AMPs to the database. In contrast, organisms that have a low taxonomic representation score, are distantly related to organisms that contribute the bulk of AMPs to the AMP database.

To obtain the taxonomic representation score for a target organism (see Figure 4.2), first a phylogenetic tree (Figure 4.2B) is used to calculate pairwise distances between all organisms, for every AMP present in an AMP database, excluding AMPs belonging to the target organism. These distances are then transformed to scores, using a mathematical function of the distance values (Figure 4.2A). These scores are therefore a function of the pairwise distance and are subsequently dependent on the pairwise distance values. This dependency is illustrated in Figure 4.2C, which shows the distance values d_i plotted against a function of distance, θ_i, termed the score, using a sigmoid curve. This function, θ_i, (see Equation 4.1), which has a shape determined by the parameter s, was selected so high scores could be obtained for closely related organisms, i.e. when the pairwise distance is low, and low scores could be obtained for distantly related organisms, i.e. when the pairwise distance is high. This is important because when organisms are closely related, the possibility of having shared AMPs due to homology is very high. However, it is likely that this possibility of shared AMPs decreases with taxonomic distance, resulting in potentially more divergent AMPs in taxonomically distant organisms. The function θ_i therefore, is an attempt to represent this. Finally, all the scores are added together into a sum value which is the taxonomic representation score for the target organism, Θ_t.

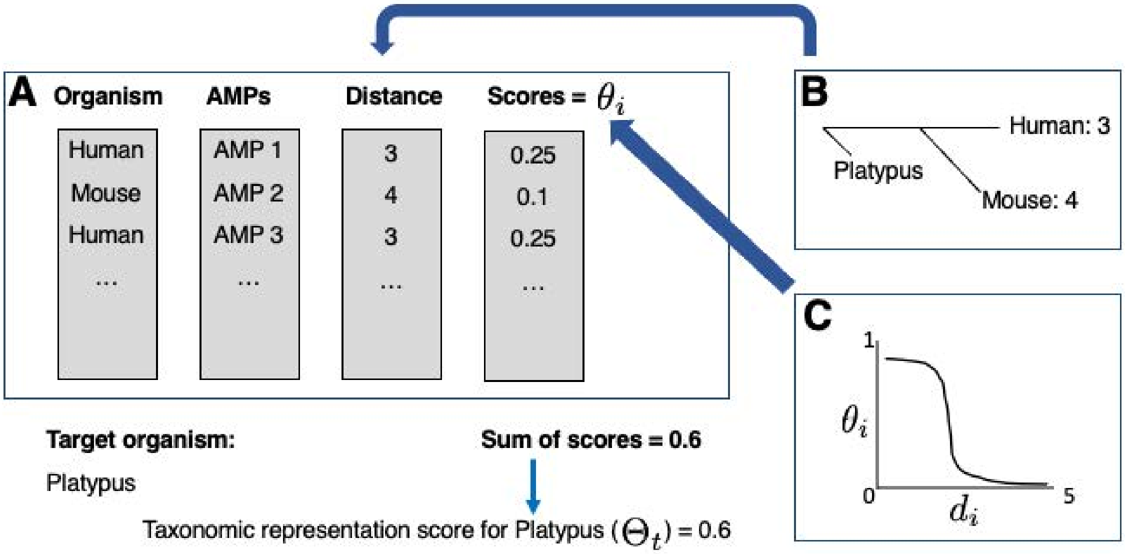


Figure 4.2: Simplified diagram illustrating how the taxonomic representation score was calculated using example data. The target organism is the organism for which the taxonomic representation score is being calculated. In this diagram, platypus is used as an example target organism. A shows an example dataset containing four columns. The Organism and AMPs columns contain all of the organisms and AMPs in the training data, except for the target organism and any AMP sequences associated with the target organism. The Distance column contains the pairwise, or taxonomic, distance between that organism and the target organism. These distance values are calculated from a phylogenetic tree, as illustrated in B. The Scores column contains scores which are calculated as a function of the d_i values, illustrated in C. In this mathematical function, θ_i (see Equation 4.1), s controls the shape of the sigmoid curve that determines the relative weighting of AMPs given their taxonomic distance, d_i, from the target organism.

Finally, the calculated scores will be summarised into a single value which will be the taxonomic representation score for the target organism, Θ_t.

4.4 Results

4.4.1 Organism selection

Within the AMP database, the eukaryotic phyla that contained the most AMPs were Chordata, Arthropoda and Streptophyta with 1,747, 685 and 556 AMPs, respectively (see Figure 4.3). The bacterium with the most AMPs was *Escherichia coli* with 29 AMPs. Within Chordata, represented with 1,747 AMPs, the class Amphibia appears to be the best studied for AMPs as this class contains the most known AMPs (825 AMPs), followed by the class Mammalia with 638 AMPs. Of these 825 amphibian AMPs, 824 correspond to a single taxonomic order, Anura. The anuran with the most AMPs was *Bombina maxima*, which has 50 known AMPs (see Figure 4.3). However, *B. maxima* did not have a reference proteome listed in UniProt and could therefore not be included. As anurans are so well represented with AMPs, an alternative organism to *B. maxima* was selected. The organism that was chosen contained the most AMPs in the anuran order and also had an available proteome, which was *Lithobates catesbeianus* (previously known as *Rana catesbeiana*) with 13 AMPs. The remaining organisms which had the most AMPs in their respective orders were the stolidobranchian *Styela clava* (tunicate), the squamate *Crotalus durissus terrificus* (snake) in Chordata and the aranea *Lachesana tarabaevi* (spider), the xiphosura *Tachypleus tridentatus* (horseshoe crab) and the scorpion *Chaerilus tricostatus* in Arthropoda. However, none of these animals had a reference proteome in UniProt. Unfortunately there were no alternatives for these organisms, i.e. there were no other organisms in their respective orders that had more than 10 AMPs and a proteome, and these taxa could therefore not be included.

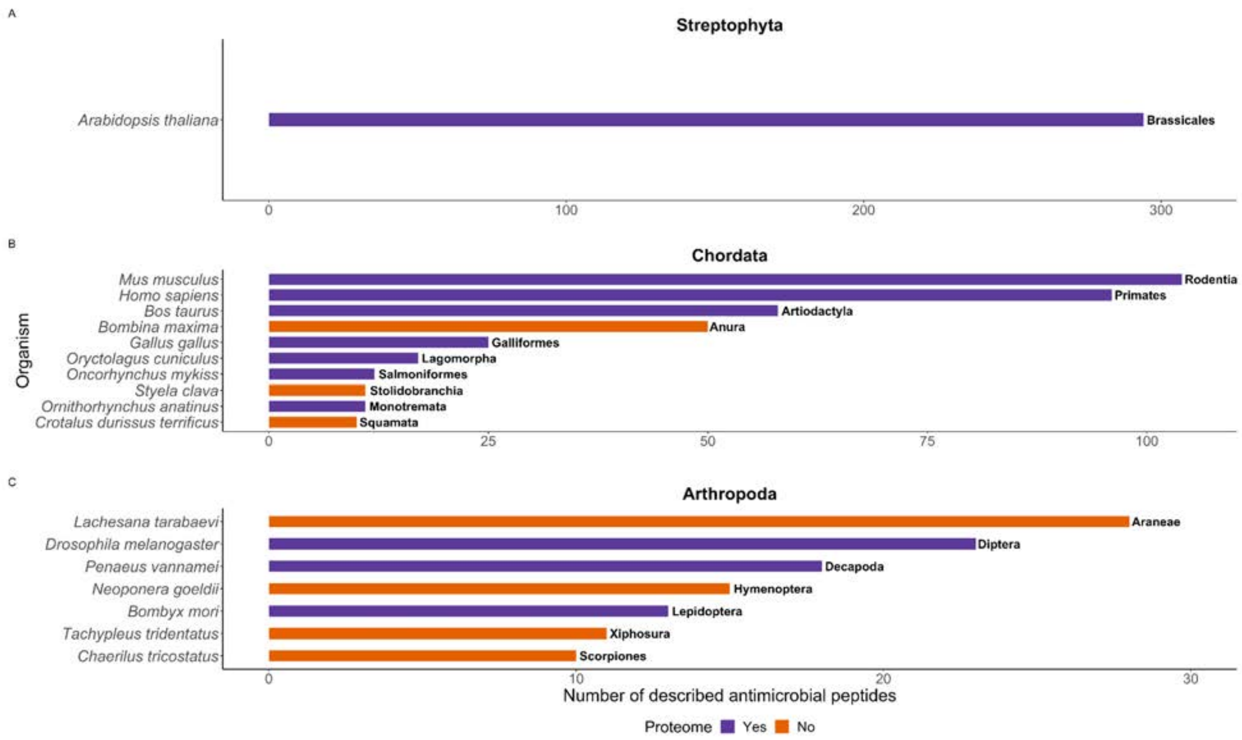


Figure 4.3: The number of described antimicrobial peptides (AMPs) in organisms that had the most number of AMPs in their respective taxonomic orders (marked in bold) from the eukaryotic phyla A) Streptophyta, B) Chordata and C) Arthropoda within the AMP database. Only organisms that had more than 10 described AMPs are shown. The organisms that were selected for further analysis included all the organisms that had a proteome (marked in purple).

The final organism selection for further analysis included six mammals: *Mus musculus* (mouse), *Homo sapiens* (human), *Bos taurus* (cow), *Oryctolagus cuniculus* (rabbit), *Ornithorhynchus anatinus* (platypus), the bird *Gallus gallus*, the frog *Lithobates catesbeianus*, the fish *Oncorhynchus mykiss*, the insects *Drosophila melanogaster* (fruit fly), *Bombyx mori* (moth), the shrimp *Penaeus vannamei*, the plant *Arabidopsis thaliana* and the bacteria *Escherichia coli* (see Table 4.1). These 13 organisms had the greatest number of AMPs within the AMP database per their respective taxonomic order, and also had a proteome. Generally there was high overlap of AMPs between the AMP database and the AMPs present in the proteomes, found via the UniProt “Antimicrobial” keyword. However, most organisms, i.e., *M. musculus*, *H. sapiens*, *B. taurus*, *O. cuniculus*, *O. anatinus*, *G. gallus*, *O. mykiss*, *D. melanogaster* and *B. mori*, contained more AMPs in their proteomes than in the AMP database. This likely reflects inclusion of some unverified AMPs in the proteomes that are not present in Swiss-Prot and therefore not in the AMP database. In contrast, the organisms *P. vannamei*, *L. catesbeianus* and *E. coli* all include more AMPs in the AMP database than in their respective proteomes. In addition, in some organisms the overlap of AMPs between the AMP database and their respective proteomes was extremely low or lacking (e.g. for *O. mykiss* and *L. catesbeianus*). Initially it was assumed that this could reflect that the proteomes lack the proper functional annotation for these AMPs, i.e. AMPs are erroneously annotated as non-AMPs. However, this was the case for only a few AMPs (see Table S4.1). The remaining AMPs, which were present in the AMP database but which were not found within the proteomes of each respective organism, could be absent because these AMPs potentially correspond to genes not yet annotated, i.e., those for which a sequence is known but the location in the genome is not. Furthermore, the *E. coli* proteome used in this chapter corresponded to the K-12 strain, which may lack AMPs that are present in the AMP database which could reflect other *E. coli* strains.

The proteomes of the organisms *O. mykiss*, *P. vannamei*, *L. catesbeianus* and *E. coli* K-12 contained fewer than 10 verified AMPs, the AMPs that overlap with the AMP database. To ensure that sufficient verified AMPs were available for calculating performance metrics, these organisms were excluded from subsequent analysis.

Table 4.1: Organisms which contain a reference proteome and have the most AMPs according to the AMP database. The number of AMPs for each organism are shown for: the AMP database, the organisms' respective proteome, the AMPs in their respective proteome which overlaps with the AMP database.

Organism Name	**Reference proteome ID**	**AMPs in AMP database**	**AMPs in proteome**	**AMPs overlap**	**Gene count**
Mus musculus	UP000000589	104	131	99	22,001
Homo sapiens	UP000005640	96	115	95	20,600

Bos taurus	UP000009136	58	116	54	23,847
Oryctolagus cuniculus	UP000001811	17	83	17	21,193
Ornithorhynchus anatinus	UP000002279	11	27	11	17,390
Gallus gallus	UP000000539	25	29	25	18,113
Oncorhynchus mykiss	UP000193380	12	15	0	46,405
Drosophila melanogaster	UP000000803	23	30	23	13,821
Penaeus vannamei	UP000283509	18	3	0	25,399
Bombyx mori	UP000005204	15	25	13	14,773
Arabidopsis thaliana	UP000006548	294	294	294	27,468
Lithobates catesbeianus	UP000228934	13	11	0	28,218
Escherichia coli K-12	UP000000625	29	4	4	4,392

4.4.2 Taxonomic representation score

The distribution of taxonomic distances of the selected organisms and their relative contribution to AMP databases is shown in Figure 4.4. The closest distances of around 100 million years are found between the majority of mammalian species and together these also comprise a large fraction of AMPs. Within the mammals, the platypus (*O.*

anatinus) is furthest away at a distance of approximately 300 million years. The next furthest are the birds (represented by chicken, *G. gallus*) at around 600 million years, followed by the insects *D. melanogaster* and *B. mori* at around 1600 million years, with the plant *A. thaliana* being furthest away at 3000 million years. Note that since these times represent the total branch length between a pair of organisms, they are equal to double the estimated divergence times.

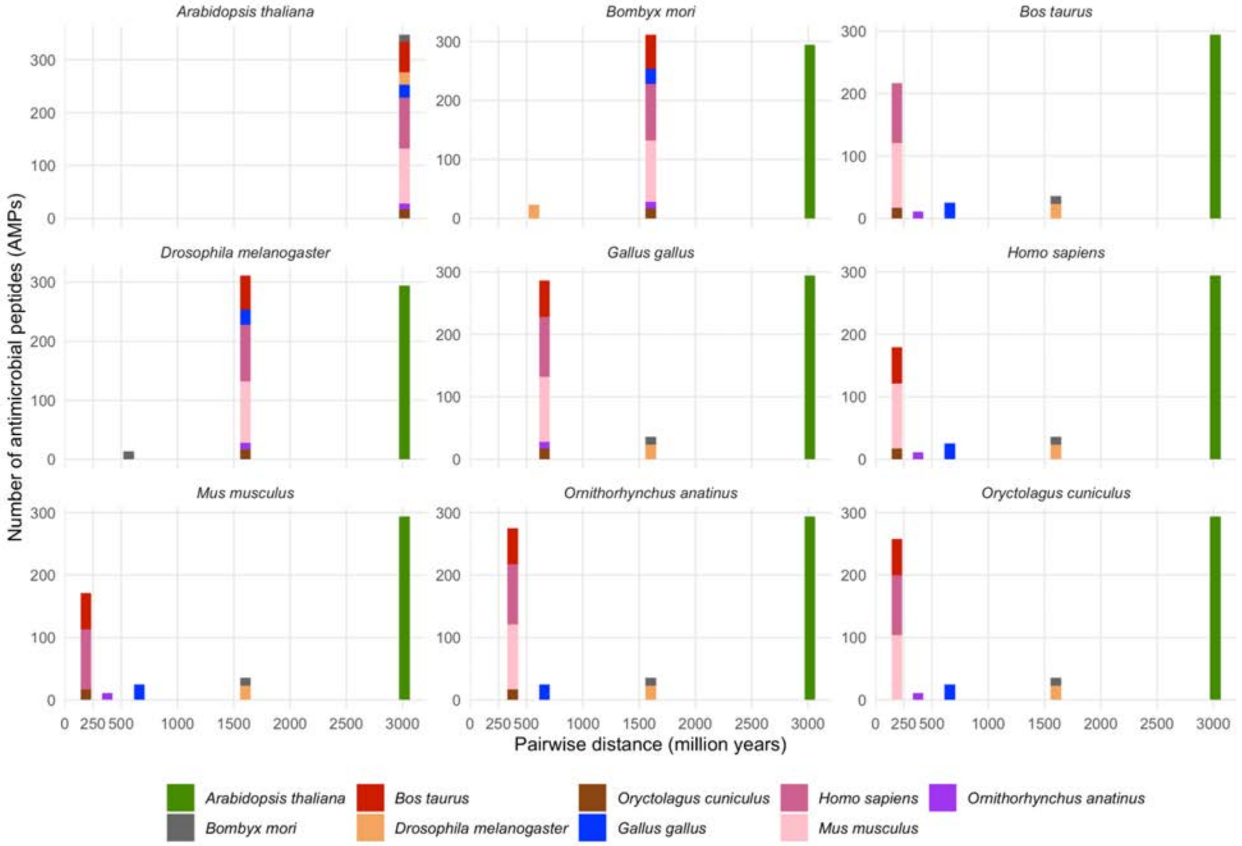


Figure 4.4: Histogram of pairwise distance between each faceted organism and other selected organisms present in the AMP dataset.

To determine an appropriate value of the parameter s in equation 4.1 different values were examined to establish the effect on distribution of taxonomic representation scores. The shape of the sigmoid curve (which determines the distance scores for individual AMPs) across a range of theoretical pairwise distances is shown for varying values of s (5, 20, 30, 50, 100 and 1000) in Figure 4.5A. This shows that as s increases the transition point between high and low scores occurs for larger divergence times.

Figure 4.5B shows the effect the various values of s have on the distribution of taxonomic representation scores across the selected organisms in this study. It is apparent that both very low (5) and high (100, 1000) values of s are not suitable for this selection of organisms. At low s the taxonomic representation score for most of the organisms is close to 0. This happens because there are too few AMPs in the database from species with taxonomic distances within the range (~30 million years) at which the score drops to 0 for this value of s. In contrast, the high s values cause the taxonomic representation score of the organisms to become too similar, as the taxonomic distance range of the organisms is likely narrower than the distance ranges covered by the higher s values. The intermediate s value of 30 appears to be an appropriate option as it appears to show the most differentiation between organisms without setting their taxonomic representation score too low or too high. Therefore, the value 30 for s was chosen as the final parameter to calculate the taxonomic representation scores for the organism selection used in the remainder of this study.

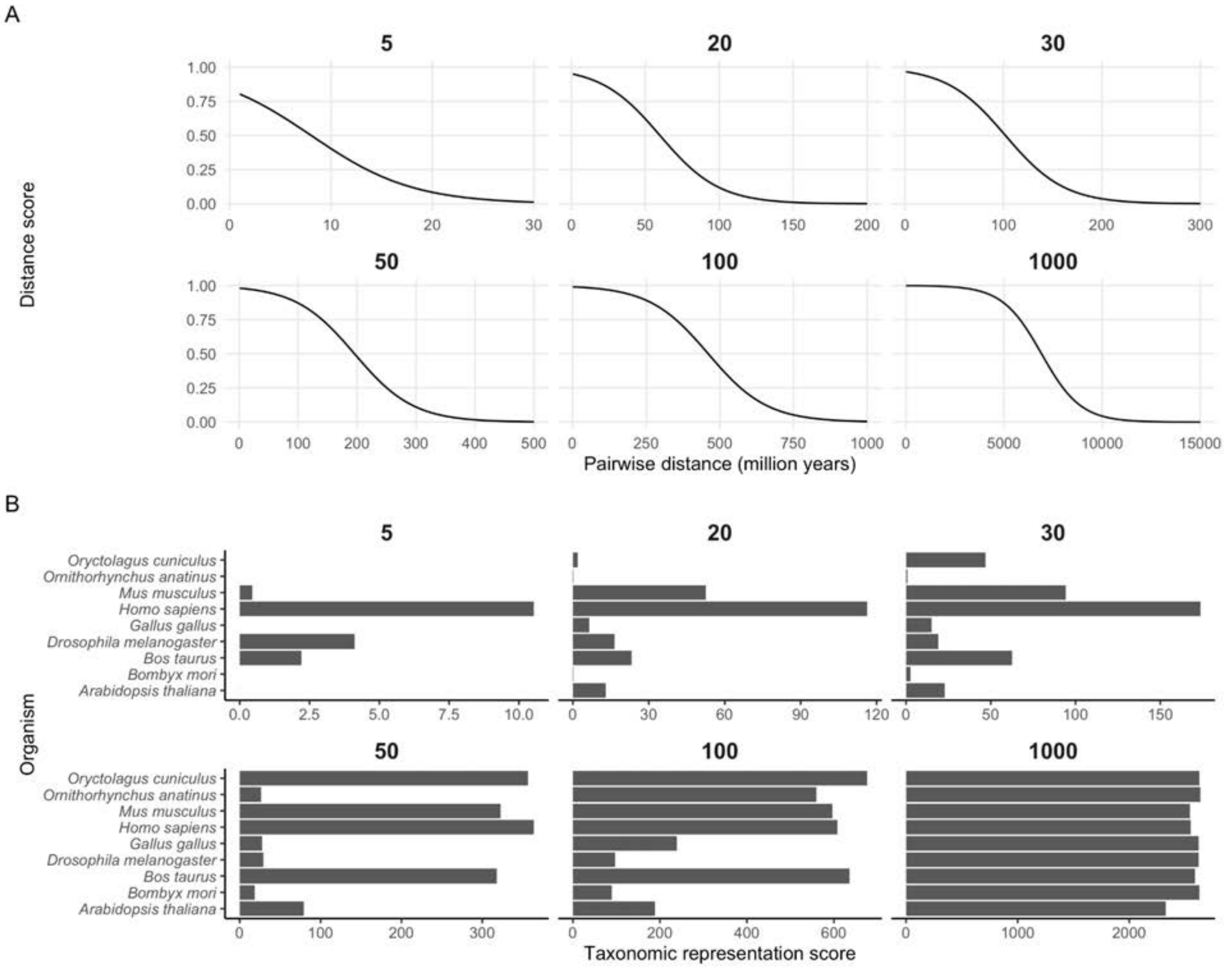


Figure 4.5: A) Theoretical sigmoid curves for various values of s B) The effect of different values of s on the distribution of taxonomic representation scores across the selected organisms.

The taxonomic representation score reflects how well the AMP database represents this organism. To understand how the composition of the AMP database affects the taxonomic representation score, the contributions of AMPs and organisms were investigated (see Figure 4.6). One of the goals was to determine the effective diversity of organisms contributing to the score. For example, how much of the score comes from a handful of closely related organisms that might possibly have high weighting and many AMPs, versus how much of the score comes from many small contributions (few AMPs or distant relationships). The bars in Figure 4.6 corresponding to each selected organism are broken down to show the contribution of AMPs of other organisms. A single coloured bar generally reflects the contribution of a single organism, with the exception to the black “Other” contributions bar, which consist of AMP contributions made by numerous other organisms. The length of the bars indicates how much that organism contributes to the taxonomic representation score. *H. sapiens* and *M. musculus* have the highest score indicating these are best represented in the AMP database. In contrast, *O. anatinus* and *B. mori* have the lowest scores and representation in the AMP database. Overall, the largest contributions to each organism appear to come from species that are closely related to that organism. The selected organisms (shown on the y axis) do not appear to contribute much to each other, with the exception of *M. musculus*, which can be observed to contribute AMPs to *O. cuniculus* and *O. anatinus*.

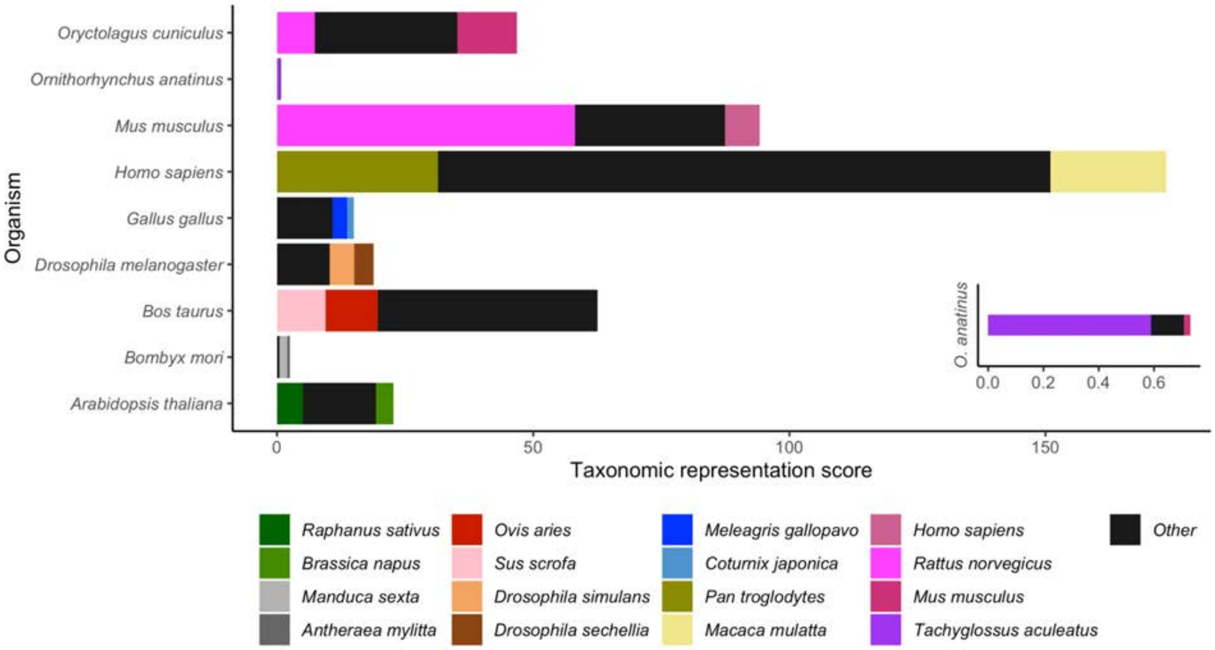


Figure 4.6: The AMP contributions of different species to the taxonomic representation score of the selected organisms. Only the two species with the largest contributions are shown. The contributions of the remaining species were amalgamated into the "Other" category. *Ornithorhynchus anatinus* is displayed as a zoomed in inset plot near the bottom right of the main plot.

The black, “Other” sections in the bars in Figure 4.6 typically comprise a large proportion of the taxonomic representation score. This is likely due to the fact that only the top two species with the largest contributions were shown, and the contributions of the remaining species were combined into the “Other” category. The mammals in particular appear to contain large “Other” sections. To unpack the contributions made by organisms within the “Other” contributions, a plot was constructed to show the top 10 contributions to the taxonomic representation scores of the mammals (see Figure 4.7).

Overall Figure 4.7 indicates that the taxonomic representation score for an organism is derived largely from closely related organisms, and to a much lesser extent, to more distantly related organisms. The makeup of the score depends on the position of the organism in the phylogenetic tree. For example, *O. anatinus* (a monotreme) has the lowest taxonomic representation score likely because it has the fewest number of known AMPs but also because it is most taxonomically distant to the other mammals. This distance becomes apparent when examining the contributions of other mammals to *O. anatinus*’ taxonomic distance score. Most mammals contribute very little, because their AMPs are weighted very low due to the high taxonomic distance. In contrast, the largest contribution, over 60%, comes from the short-beaked echidna, *Tachyglossus aculeatus*, as this animal is a monotreme and is therefore most closely related to *O. anatinus*. Even though *T. aculeatus* has far fewer known AMPs compared to the other contributing mammals, its AMPs are heavily weighted due to the close taxonomic relationship to *O. anatinus* and subsequently greatly influences its taxonomic representation score. Similarly, *R. norvegicus* contributes the most, almost 60%, to *M. musculus*’ score due to their close taxonomic relationship. In contrast, *R. norvegicus* contributes much less to organisms that are taxonomically further away, e.g. *R. norvegicus* contributes only about 3% to *B. taurus*. Organism contributions therefore vary depending on the taxonomic relationship and the total number of AMPs contributed.

In contrast to *O. anatinus* and *M. musculus*, which both have a skewed contribution distribution driven by the contribution of a single organism, *H. sapiens* and *B. taurus* contain a more even distribution of organisms that contribute to their taxonomic representation scores. This is likely because, as can be observed in Figure 4.7A, their respective taxonomic orders are better represented, i.e. their orders contain more contributing species, compared to the monotreme and rodent orders. This indicates that additional species and AMPs could improve the taxonomic representation score, especially for organisms that are more taxonomically isolated.

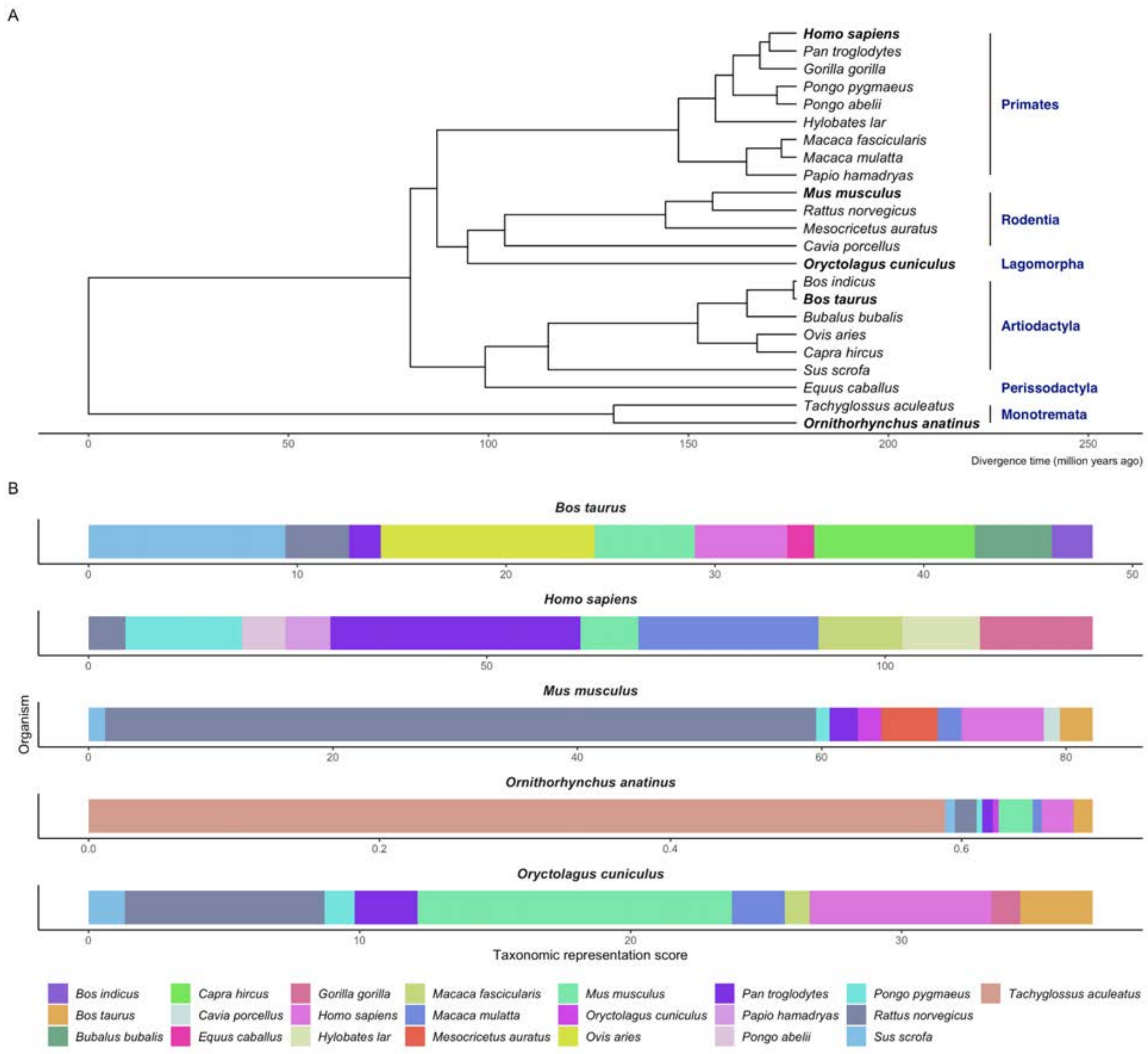


Figure 4.7: A) A phylogenetic tree from the five mammals present in the selected organisms (*Homo sapiens*, *Mus musculus*, *Oryctolagus cuniculus*, *Bos taurus* and *Ornithorhynchus anatinus*, marked in bold), and the mammal species that contribute most to the taxonomic representation score of these mammals. The respective

taxonomic orders of the mammals are marked in dark blue bold. B) The contributions of the different mammal species to the taxonomic representation score of the mammals in the selected organisms. Only the 10 species with the largest contributions are shown.

4.4.3 Performance of AMP finding methods on selected organisms

BLAST performance was significantly positively correlated with the taxonomic representation score (see Figure 4.8) according to a Spearman’s rank correlation test (p-value = 0.02, Spearman’s rank correlation coefficient, ρ = 0.78). In contrast, no such correlation was found for the machine learning predictor (ampir) (p-value = 0.74, ρ = 0.13). Machine learning distinctly outperformed BLAST for three organisms with a low taxonomic representation score, i.e. *O. anatinus*, *G. gallus* and *A. thaliana*. Although BLAST performed better than machine learning for two different organisms with low taxonomic representation scores, *B. mori* and *D. melanogaster*, the differences between methods were relatively small in these cases. The two species with the largest taxonomic representation score are *M. musculus* and *H. sapiens*. The analyses were repeated excluding *M. musculus* and *H. sapiens* which resulted in BLAST: p = 0.24, ρ = 0.54, and machine learning: 0.56 and ρ = -0.29. Similar to the initial results, BLAST still contains a higher correlation coefficient with taxonomic distance compared to machine learning, however, without statistical significance.

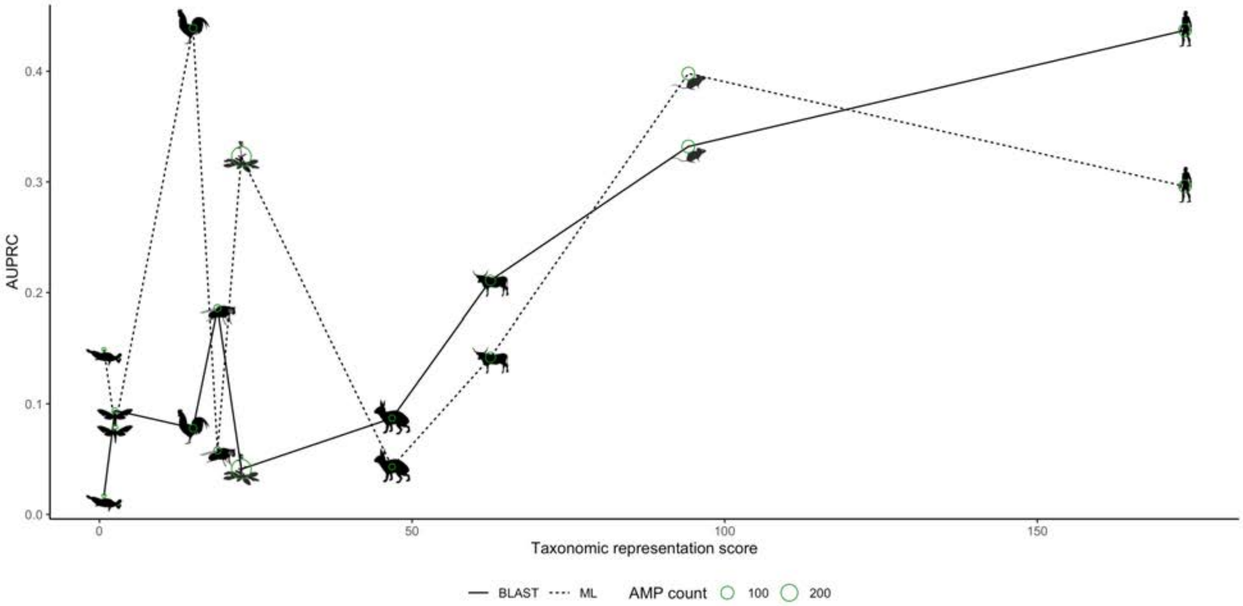


Figure 4.8: The performance of the BLAST (black solid line) and machine learning (ML) (black dashed line) methods to find AMPs in the proteomes of nine organisms. Performance is measured in the Area under the Precision-Recall curve (AUPRC). AMPs were labelled as AMPs if annotated with the UniProt "Antimicrobial" keyword and if

these AMPs overlapped with the AMP database generated from Swiss-Prot and the APD, DRAMP or dbAMP databases. The nine organisms from left to right are: *Ornithorhynchus anatinus*, *Bombyx mori*, *Gallus gallus*, *Drosophila melanogaster*, *Arabidopsis thaliana*, *Oryctolagus cuniculus*, *Bos taurus*, *Mus musculus* and *Homo sapiens*.

To test whether this overall result (BLAST performance depends on taxonomic distance while machine learning does not) is robust to the choice of s, alternative results across a range of s are shown in Table 4.2. This showed that machine learning performance was not significantly correlated to taxonomic representation for any value of s and always had lower correlation coefficients compared to BLAST. This illustrates the robustness of the findings to the variations in s, to the extent that it was possible to test this within the constraints of the dataset (same set of organisms for all tests).

Table 4.2: The effect of different sigmoid-values (s) on the statistical performance of AMP finding methods BLAST and machine learning (ML) using a Spearman's rank order correlation test. ρ: Spearman's rank correlation coefficient.

s-value	**Method**	**p-value**	**ρ**	**Test statistic**
5	ML	0.55	0.23	92
	BLAST	0.06	0.67	40
20	ML	0.39	0.33	80
	BLAST	0.02	0.77	28
30	ML	0.74	0.13	104
	BLAST	0.02	0.78	26
50	ML	1.00	0.00	120
	BLAST	0.10	0.60	48

100	ML	0.84	-0.08	130
	BLAST	0.44	0.30	84
1000	ML	0.84	-0.08	130
	BLAST	0.44	0.30	84

4.4 Discussion

There is an implicit assumption that machine learning models offer an advantage in AMP prediction over conventional BLAST searches as evidenced by the numerous AMP machine learning models being developed. However, in the literature surveyed, this presumed advantage has not been empirically tested. Given that BLAST remains one of the most frequently used tools to discover novel AMPs it is important to test its relative efficacy compared with the alternative machine-learning-based approach.

This study compared the performance of machine learning models and BLAST searches to find AMPs in the proteomes of organisms that cover a wide taxonomic range (see Figure 4.8). The general findings were that BLAST was significantly better at finding AMPs in close taxonomic distances compared to AMPs that were further away. However, no such statistical difference was found in the performance of machine learning models. This indicates that AMP machine learning models are less dependent on taxonomic distance than homology and may therefore be more effective at finding novel AMPs in taxonomically distant organisms. Although this was an expected result based on the premise that machine learning models are less reliant on homology, it was not guaranteed since machine learning models do depend on the composition of their training data, and this is heavily taxonomically biased. It is the first empirical demonstration of the intuition that machine learning models might degrade less quickly at larger taxonomic distances because they are supposed to be looking for physicochemical properties that are common across the tree of life. In contrast, since

BLAST works by identifying similar sequences it is expected that its performance for AMP classification would be heavily dependent on taxonomic distance, and specifically the availability of AMPs from closely related species within its database.

This conforms largely to how BLAST is used to discover novel AMPs; primarily by using known AMPs to find similar novel AMPs in closely related species (Panteleev et al., 2020; Xiao et al., 2020; Chen et al., 2021; Peel et al., 2021; Zhuang et al., 2021). These studies predominantly use species from taxa for which multiple AMPs are known (e.g. amphibians, mammals and fish). In contrast, studies which use species from less represented taxa (e.g. insects and mollusks) appear to favour searching AMP databases for homologous AMPs (Duwadi *et al.*, 2018; Hayashida and da Silva Junior, 2021; Wang *et al.*, 2021). Studies which implemented a machine learning approach to discover novel AMPs similarly used organisms from less represented taxa e.g. insects (Lee *et al.*, 2020a; Lee *et al.*, 2020b; Lee *et al.*, 2021b) and a crustacea (Yang *et al.*, 2018). These studies overall found more AMPs compared to the studies which used BLAST, which subsequently led to a higher number of synthesised and verified AMPs. Interestingly, the aforementioned studies which used machine learning AMP predictors, all used fairly old predictors, the most common being CAMP (Thomas *et al.*, 2010). It is possible that the prediction results obtained in these studies could have been improved upon via the use of more modern and better optimised machine learning AMP predictors. Especially considering it is becoming more common to predict AMPs from a genome or transcriptome, which require genome-wide optimised machine learning models (see chapter 2 and 3).

The results confirm the intuition that machine learning should work best at large taxonomic distances, presumably due to its focus on physicochemical properties rather than homology, whereas BLAST may be most effective for identifying AMPs in very closely related species (Lata, Sharma and Raghava, 2007; Lata, Mishra and Raghava, 2010; Khosravian *et al.*, 2013; Ohtsuka and Inagaki, 2020; Santos-Júnior *et al.*, 2020). Nevertheless, the limited testing data and highly variable coverage of known AMPs across species mean that these results should be interpreted with caution. Specific issues of concern are (1) the fact that many AMPs remain undiscovered which is likely to have inflated false negative results for some species (especially those with less complete AMP coverage) in this study, (2) the limited number of species that could be included resulting in low statistical power for the analysis presented in Figure 4.8, and (3) the possibility that some AMPs in our truth set may themselves have been annotated via some form of homology search leading to a degree of circularity in the findings. Furthermore, the removal of *M. musculus* and *H. sapiens* removed the statistical significance result for BLAST. This is likely due to the fact that these two organisms contain the highest taxonomic representation scores. However, it must also be noted that there is a large gap between the taxonomic representation scores of *M. musculus* and *H. sapiens* ($\Theta > 94$) and the remaining organisms ($\Theta < 62$). The current analysis represents the best effort that can be made given the current state of AMP databases, however, it is hoped that as these databases are improved the results can be verified and improved upon.

One area where greater precision would be of value is in determining the threshold taxonomic distance at which the performance of BLAST degrades below that of machine learning models to find AMPs. The results of this study were insufficiently precise for this purpose, and therefore only a general statement can be made over the extremes of life's taxonomic distance: in the extreme of short taxonomic distance, BLAST will work very well, and potentially better than machine learning, and in the extreme of long taxonomic distance, BLAST is unlikely to work well at all.

The measured performance of AMP finding methods was found to be highly variable, sometimes between organisms within relatively close taxonomic distances. This could be due to peculiarities in the AMP repertoires of specific organisms. For example, BLAST performed particularly poorly on the proteome of *O. anatinus*, compared to its performance on the other mammals (Figure 4.8). In this species, one AMP family, the cathelicidin family, has diversified compared to eutherian mammals (Warren *et al.*, 2008), and this may have contributed to the poor performance of BLAST to identify AMPs in *O. anatinus*. This indicates that organisms contain unique AMPs that even in closely related organisms, may be difficult to predict, particularly with homology-based methods.

These results suggest that machine learning is a promising method to discover novel AMPs, especially in organisms that are taxonomically isolated and are not well represented in the AMP databases. Furthermore, machine learning methods could offer advantages to discover novel AMP less similar to known AMPs in closely related taxonomic groups. Although both homology-based methods and machine learning methods are reliant on the known AMPs present in the AMP databases, machine learning appears to be more robust as it is not solely dependent on sequence similarity. Nevertheless, homology-based methods are adept at finding closely related AMPs. Therefore, a combination of homology and machine learning methods would likely result in the highest degree of AMP prediction coverage. It must be noted that the machine learning results were obtained by using specified methods designed for the machine learning predictor ampir, to optimise AMP prediction on a genome-wide scale. This method was used to most accurately predict AMPs in proteomes. Different AMP predictors, which do not prioritise genome-wide AMP discovery and which may use different test sets (i.e. not proteomes), may obtain varying results.

One of the challenges of this study was choosing a suitable value for the parameter, s, used in the calculation of the taxonomic representation score. In choosing this value it was important to keep in mind the taxonomic distance scale, as this is directly related to the value of s, i.e. larger sigmoid-values cover larger taxonomic distances (see Figure 4.5A). While it was clear that extreme sigmoid-values, i.e. either very low or high, were inappropriate, it was challenging to choose from among the intermediate s values. However, the final chosen value of s (30) appeared most appropriate as this value caused the taxonomic representation score to show clear differentiation between organisms. This value simultaneously minimised extreme values of the taxonomic representation score (i.e. where taxonomically distant organisms resulted in a taxonomic representation score that was close to 0, or where the scores of closely related organisms became too similar). The robustness of this finding in the variations of

s was demonstrated in Table 4.2, where the correlation pattern became clearer with mid-range s values. Finally, the differentiation between organisms with the chosen value of s 30 ultimately highlighted the differences between machine learning and homology, which was the goal of this study.

4.5 Conclusion

This study examined the performance of machine learning models and homology-based searches to discover AMPs in the proteomes of organisms across a wide taxonomic scale. A novel taxonomic metric, the taxonomic representation score, was employed to provide a score on how well the AMPs in an organism are represented in a database consisting of currently known AMPs. It was found that only the homology-based searches method was correlated with the taxonomic representation score, suggesting that homology-based methods to find AMPs are only suitable for organisms for which many closely related AMPs are known. No such restriction was found for the machine learning method, indicating that machine learning models are a useful tool for novel AMP discovery. This study provides important implications for the AMP discovery field as it highlights limitations and appropriateness of AMP prediction methods available for AMP discovery.

Chapter 5: General Discussion

5.1 Major outcomes and significance

Antimicrobial peptides (AMPs) are natural antibiotics that are produced by the innate immune system in all life forms. The major roles of AMPs include defending the host against pathogens and regulating the hosts' microbiome (Zhang and Gallo, 2016). AMPs are therefore critical in maintaining the health of the host organism. Due to the effectiveness of AMP activity against microbes, and the rise of antimicrobial resistance, AMPs are of great interest as candidates for therapeutic drug design (Moretta *et al.*, 2021). To facilitate AMP discovery, many machine learning AMP predictors have been developed (Xu *et al.*, 2021). In addition, advances in genomic sequencing are revealing the complete sequences of genomes and gene products for an ever increasing number of organisms. Bioinformatic tools, designed to analyse these large datasets, are now required across a wide range of applications including AMP prediction (Yin *et al.*, 2017). The overall aim of this thesis was to fill this gap, by developing improved machine learning methods, specifically designed for identifying antimicrobial peptides in genome-scale data. The initial aim was simply to adapt current methods, designed for use with very small datasets, to genome-wide scanning workflows. At the time I commenced this thesis, almost all available AMP predictors, e.g. (Thomas *et al.*, 2010; Xiao *et al.*, 2013; Meher *et al.*, 2017; Veltri, Kamath and Shehu, 2018) were only accessible via a web interface. These AMP prediction web interfaces often limited the number of sequences that could be provided as input (Xiao *et al.*, 2013; Meher *et al.*, 2017; Yan *et al.*, 2020) making them impractical to use for entire proteomes. Additionally, web servers are difficult to include in bioinformatic pipelines, frequently used for large scale analyses, such as comparative genomics. Therefore, a new bioinformatic tool, called ampir, which incorporated an AMP prediction model and which was optimised for large data, was developed (chapter 2).

During the construction of ampir, several inherent problems were identified with the approaches used for selection of training data (chapter 2) and testing data (chapter 3) in existing AMP prediction models. These problems arise partly as a result of efforts to compensate for inherent limitations in available data, but they also reflect issues that only become apparent when scanning large datasets (such as genomes) for AMP candidates. These issues are summarised in the conceptual figure (Figure 5.1)

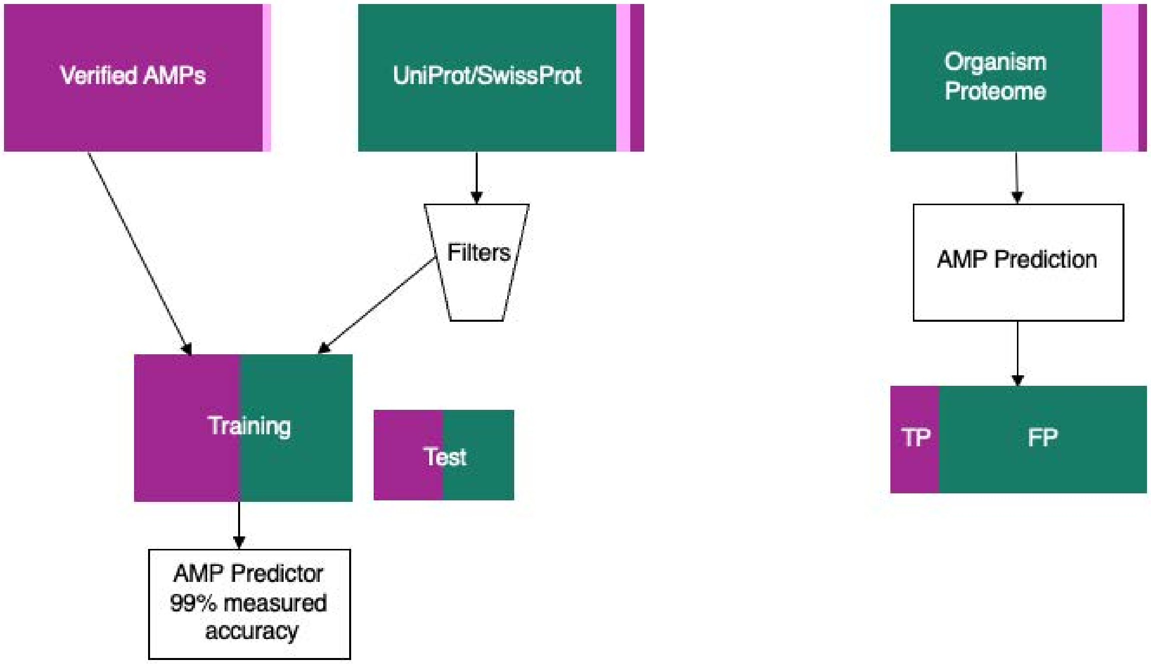


Figure 5.1: Summary of the AMP predictor development process (left) and how this impacts AMP prediction on real world datasets (right). Green indicates non-AMPs and other colours indicate AMPs. Data that are complex in composition are simplified through filtration, often removing the most challenging cases (light pink) to produce balanced training and test sets with uniformly biased composition. This results in inflated measures of AMP predictor performance. Real world datasets such as an organism proteome are complex and include challenging cases that reduce real performance. In addition, they are highly unbalanced leading to a large number of false positives. TP: true positives, FP: false positives.

The first issue concerning the AMP predictor development process is that no database currently exists that contains verified non-AMP sequences that can easily be used for the negative dataset (Liu *et al.*, 2017). To compensate for this, the UniProt protein database (UniProt Consortium, 2021) is almost always used as a foundation for the negative dataset, usually by selecting proteins not annotated with keywords such as “antimicrobial”, “antibiotic”, “antibacterial”. However, in addition to these keywords, proteins that are secreted are also commonly excluded (Thomas *et al.*, 2010; Bhadra *et al.*, 2018; Veltri, Kamath and Shehu, 2018; Jhong *et al.*, 2019; Santos-Júnior *et al.*, 2020; Xu *et al.*, 2021). This is likely done because AMPs are secreted proteins and removal of secreted proteins in the negative dataset would therefore further contrast the positive and negative dataset. This contrast enhances the apparent performance of the AMP predictor when tested against a similarly biased dataset, but compromises real-world performance by removing the most challenging cases. This is especially problematic for secreted proteins, as these constitute a substantial part of the genome (~13% for human) and, in addition to the innate immune system, contribute to a variety of biological pathways, including those with functions related to e.g. metabolism and insulin regulation (Pinheiro-Machado *et al.*, 2020). Subsequently, by specifically removing secreted proteins from the negative dataset, the AMP predictor may incorrectly link secretion to be a unique AMP characteristic. Consequently, the AMP predictor is likely to have a high misclassification rate when it is used on a dataset which contains a large proportion of secreted proteins (e.g. a genome). The second issue this thesis discovered with the training data used for AMP prediction was that it often contains a mix of precursor proteins and mature peptides (chapter 2 and 3), even though mature peptide sequences are very rarely known for novel putative AMPs on a genome scale. Chapter 3 revealed that AMP predictor models which include mature peptides in their training data, were less effective at classifying AMPs in ‘omics datasets, which purely contain precursor proteins.

This thesis was the first to survey the training data of AMP predictors with the goal of recognising and quantifying these issues. Whether or not a training dataset is appropriate depends entirely on the way that a given AMP predictor is used in practice. This thesis identified two real world use cases for AMP predictors. One is where the input dataset is low-throughput and consists of a few mature peptides, e.g. those that are synthesised, which users may want to test for antimicrobial activity. The other is for genome-wide scanning using ‘omics datasets, where thousands of precursor proteins are classified to find AMP candidates. Therefore, two models were implemented into ampir, one for use on mature datasets (ampir_mature, trained exclusively with mature peptides) and one for use on ‘omics datasets (ampir_precursor, trained exclusively with precursor proteins). Ampir was the first AMP predictor to adopt this approach. This thesis focused primarily on the development and use of the ampir_precursor model in a genome-scanning context. In this context, it was identified that the composition of the negative training dataset for AMP models should be as close to a typical set of non-AMP proteins in a genome as possible. This aids in preparing the model to be used on real biological datasets. Therefore, the negative training data for ampir were selected using minimal filtering. The main filters imposed were sequence length filters, to separate the likely mature peptides from precursor proteins. By using minimal filtering of the negative dataset, and by solely including precursor proteins in the training dataset in the ampir_precursor model, it was shown that classification performance on ‘omics datasets can be greatly improved (chapter 3). This is important because the genome-scanning use case is now commonly used as part of an overall workflow for identifying novel AMPs (Lee *et al.*, 2020a; Lee *et al.*, 2020b; Yakovlev *et al.*, 2020; Hassan, Qutb and Dong, 2021; Lee *et al.*, 2021a; Lee *et al.*, 2021b; Liscano *et al.*, 2021; Onime *et al.*, 2021; Ma *et al.*, 2022)

Removing secreted proteins and including mature peptides in AMP predictor training data are just two aspects of a broader issue, that is, a mismatch between the composition of training and test data with real data used for prediction. Common statistical practice is to evaluate the performance of predictors on hold-back test sets, which are compositionally similar to the training datasets (chapter 3). Since it is extremely challenging, and indeed impractical, to ensure that training data for AMP predictors reflects the composition of real proteomes, I proposed that well characterised proteomes such as those of *Homo sapiens* and *Arabidopsis thaliana* be used as an additional performance test. However, this is highly dependent on the availability of well characterised proteomes and may not suit all AMP predictor use cases (e.g. those optimised to discover specific AMP families).

Another crucially important issue that has so far received little attention in the AMP predictor literature (except see Santos-Júnior *et al.* (2020)) is that of the test dataset balance. However, it must be noted that this is a separate issue from using an imbalanced training dataset which has been explored in several AMP predictors and has been suggested to increase model performance (Xiao *et al.*, 2013; Lin and Xu, 2016; Bhadra *et al.*, 2018; Santos-Júnior *et al.*, 2020). As recognised in a recent review (Whalen *et al.*, 2022), imbalance is the default state for genomic data and this is certainly true in the context of 'omics workflows that attempt to identify AMPs. In this thesis I surveyed AMPs in well characterised proteomes and found that they only comprise approximately 1% of cases, yet the majority of AMP predictors use test sets where AMPs and non-AMPs are balanced. Subsequently, as shown in chapter 3, the reported performance of these AMP predictors do not reflect the performance obtained from highly imbalanced datasets. These findings have also been highlighted in recently published literature as a common problem in the application of machine learning models in genomics (Whalen *et al.*, 2022). This highlights the well-timed relevance of this thesis, as the practical use of machine learning models in genomic prediction is being revised to reflect more realistic use. Some recommendations made in this thesis are similar to those made by Whalen *et al.* (2022), that the imbalance used in the test set should be similar to the imbalance found in real genomic data. However, in this thesis I also explored the idea that predictors could be benchmarked according to their performance in the low false positive regime. This is important in many applications where the overall false negative rate is less important than identifying a (potentially small) number of true positives with minimal false positives. In 'omics-scanning the need to reduce false positives is very high as this increases costs associated with wet-lab validation. Some AMP predictors e.g. Veltri, Kamath and Shehu (2018) performed particularly poorly in this regard, possibly because they were not designed to operate at any decision threshold other than p=0.5. Some performance metrics commonly used to evaluate the performance of AMP predictors on test sets are highly sensitive to the balance of classes in datasets and should therefore be reported only for a standardised dataset balance (usually chosen as 1:1) (Chicco, 2017; Sofaer, Hoeting and Jarnevich, 2018; Chicco and Jurman, 2020; Chicco, Tötsch and Jurman, 2021; Whalen *et al.*, 2022).

A final issue explored in this thesis was the taxonomic composition of currently characterised AMPs and how this affects AMP finding methods. This is especially important for studies which focus on identifying novel AMPs in organisms from taxa that have low representation in the AMP database. These types of studies are increasingly common as modern genome sequencing methods are revealing genomic resources for a wide range of non-model taxa. A recent study assessed the effectiveness of various machine learning based AMP prediction tools on different invertebrate taxa and found considerable variation in performance between taxa and predictors (Rádai, Kiss and Nagy, 2021). In that paper the goal was to provide an indication of which predictors perform best in different taxonomic groups. While such a review may provide a useful guide for users in the short-term it is not clear to what extent the results generalise to taxa or predictors that were not included. In this thesis (chapter 4) I attempted a more general approach to addressing the issue of taxonomic bias in AMP prediction by 1) training custom organism specific models where the training data for that model excluded all proteins belonging to the target organism and 2) developing a novel taxonomic distance metric that measured how well an organism is represented in the training data. Overall, no significant difference was discovered in the performance of the custom machine learning models relative to taxonomic distance, however considerable variation in performance was observed between organisms. This might indicate that taxonomic distance alone is not a good predictor of model performance, and therefore that exhaustive approaches such as that of (Rádai, Kiss and Nagy, 2021) are necessary. Alternatively it might reflect some of the many difficulties I encountered in generating high quality, independent, taxon-specific test sets. As AMP databases improve it may be worthwhile repeating the work in chapter 4 to re-evaluate the relationship between performance and taxonomic distance with improved data. Nevertheless, one clear finding was obtained in chapter 4, namely that machine learning AMP predictors that use physicochemical properties as features are better equipped to predict AMPs in taxonomically distant organisms compared with conventional sequence similarity methods such as BLAST, which showed a significant decline in performance with taxonomic distance. This thesis was the first to empirically test the relative effectiveness of machine learning based AMP prediction and homology-based methods to discover AMPs in organisms that span a wide taxonomic range.

A survey of methods used to identify novel AMPs provided as part of chapter 4 highlighted the fact that homology-based methods remain more widely used than machine learning AMP predictors as tools to produce candidate lists for experimental validation. It may be that homology-based searches are preferred by some authors as their focus is on finding similar AMPs in taxonomically close organisms, e.g. (Panteleev *et al.*, 2020; Xiao *et al.*, 2020; Chen *et al.*, 2021; Peel *et al.*, 2021; Zhuang *et al.*, 2021). Nevertheless, I found that homology-based searches are also used to discover AMPs on a genome-wide scale, including on organisms that are not well represented in the AMP database (Duwadi *et al.*, 2018; Wang *et al.*, 2019; Yakovlev *et al.*, 2020; Hayashida and da Silva Junior, 2021; Zhang *et al.*, 2021a). Based on my findings in chapter 4 I would expect that these authors would benefit from adopting a machine-learning AMP prediction approach, and that this recommendation be applied for any organism that lacks strong representation of AMPs from close relatives in public sequence databases.

5.2 Limitations and Future directions

This thesis discovered some issues that currently limit the ability of AMP predictors to reach optimal performance. These limitations are overall related to available data, which also cover limitations to the field in general.

First, AMPs are diverse (Schmitt, Rosa and Destoumieux-Garzón, 2016) and in addition to evolving from common ancestry, can also arise via convergent evolution (Unckless and Lazzaro, 2016; Lazzaro, Zasloff and Rolff, 2020). Despite over 3,000 AMPs currently known and available to be used in the training data in AMP predictors, a large proportion of these comprise close homologs from a select few taxa. Therefore, it is likely many AMPs and potential AMP families remain undiscovered. Despite the fact that machine learning based AMP predictors are trained on physicochemical properties, which can reflect antimicrobial activity (Torrent *et al.*, 2011; Meher *et al.*, 2017), physicochemical properties can vary between different AMP families (Khamis *et al.*, 2015; Wang *et al.*, 2020c; Rádai, Kiss and Nagy, 2021). Furthermore, physicochemical properties of AMPs may not be consistent across taxa, even for AMPs that correspond to the same AMP families (Rádai, Kiss and Nagy, 2021). In addition, there are also AMP families that are distinct to certain taxonomic groups, e.g thionins in plants (Höng *et al.*, 2021) which could have distinct physicochemical properties. AMP predictor models can therefore potentially be optimised by constructing taxon specific models. However, this is heavily dependent on the availability of taxon specific AMPs, which are currently lacking for many taxa. Therefore, it is recommended that the AMP discovery field focus their attention on finding, and experimentally verifying activity for, AMPs in taxa currently not well represented. As this requires a lot of resources, efforts should initially be focussed on a single representative organism for under-represented taxa of particular interest. A genome-wide optimised AMP predictor can then be used on the proteome of that organism to find AMP candidates. The most likely candidates (i.e. those with high probability scores and suitable structures) can then be synthesised and characterised. Once verified, these proteins can be appropriately annotated in the proteome of the organism and added to the training data of the model. With the added AMPs, the model can then be retrained to increase its learning ability. Repeating this cycle, and complementing the search with homology-based prediction should eventually reveal a high proportion of the AMP repertoire for the organism.

A potential method to select organisms that represent taxa currently understudied for AMPs is by examining the number of AMPs currently characterised in various taxa whilst similarly considering the number of species present in these taxa. For instance, investigation of the number of AMPs present in various taxa in the Swiss-Prot database revealed that the majority of currently characterised AMPs correspond to the Chordata phylum (see Figure 5.2). The Arthropoda phylum is represented by approximately half of the number of AMPs compared to Chordata, despite the Arthropoda phylum containing far more species. This apparent preference for characterising AMPs in select taxonomic groups can also be observed on a finer level, e.g. within classes in the Chordata and Arthropoda phyla (Figure 5.2B) which clearly shows the uneven distribution of characterised AMPs in different taxa.

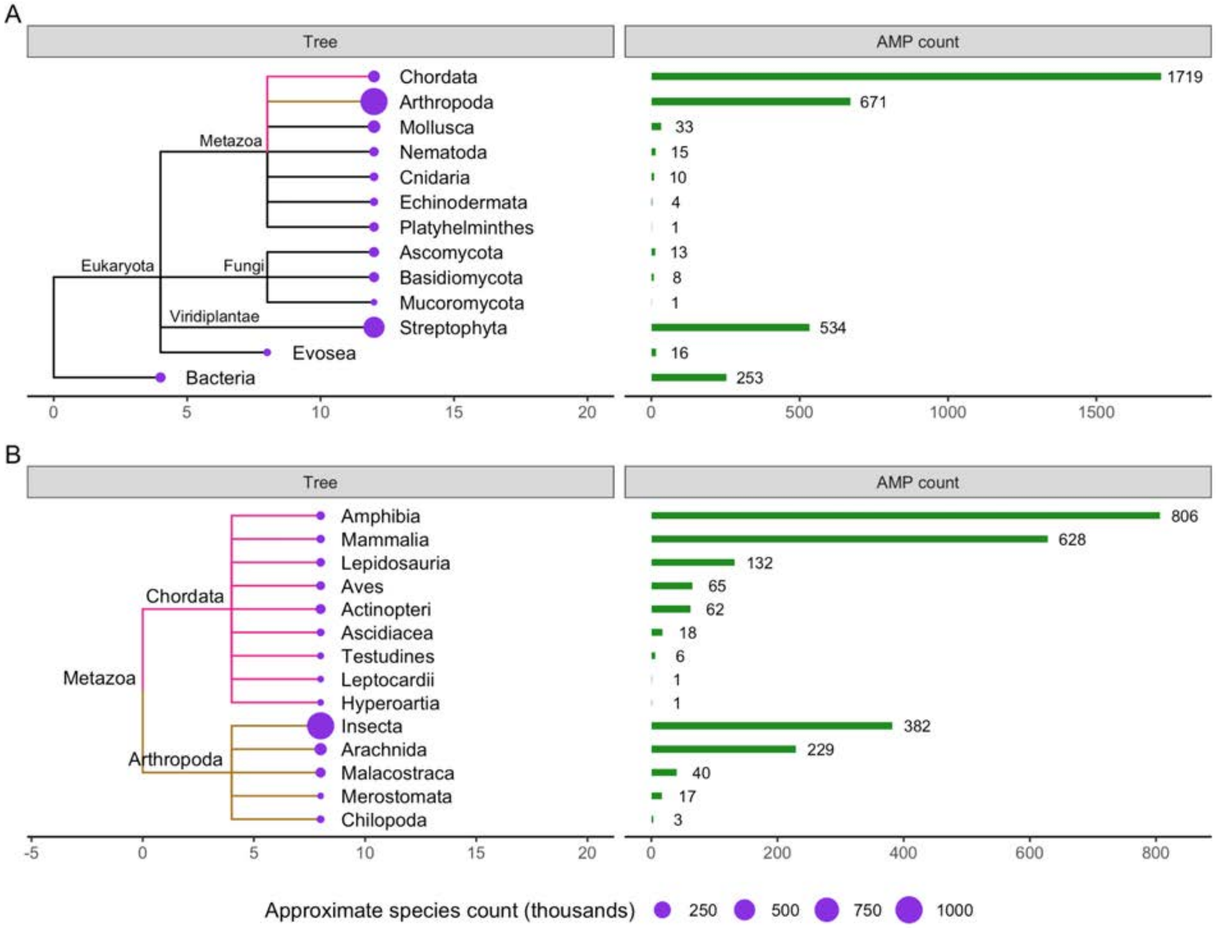


Figure 5.2: Phylogenetic trees constructed using the NCBI Taxonomy Browser https://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi and the ggtree R package (Yu *et al.*, 2016) displaying the number of antimicrobial peptides in A) bacteria and phyla within Eukaryota and B) classes within the metazoan phyla Chordata (pink branches) and Arthropoda (brown branches) in the Swiss-Prot database (accessed April 2021).

Invertebrates are likely a good source of novel AMPs due to their extensive species diversity and potential for taxon specific AMPs. In addition, invertebrates lack an adaptive immune system which may mean they contain a higher diversity of AMPs to act as their defence mechanism (Tincu and Taylor, 2004). Furthermore, there are multiple invertebrate taxonomic groups in which relatively few AMPs have been characterised, despite their species diversity. This is especially apparent in the metazoan phyla Mollusca, Nematoda, Cnidaria, Echinodermata and Platyhelminthes (see Figure 5.2A). These phyla largely reflect marine invertebrates which have been frequently highlighted as a promising source for novel AMP discovery (Tincu and Taylor, 2004; Rosenstiel *et al.*, 2009; Destoumieux-Garzón *et al.*, 2016; Schmitt, Rosa and Destoumieux-Garzón, 2016; Panteleev *et al.*, 2020; Wu *et al.*, 2021). Therefore, selecting organisms within these unrepresented taxa as a focal point for AMP discovery can greatly improve the training data for AMP predictors, especially to discover taxon specific AMPs.

Another way in which organisms can be under-represented in AMP databases is if they lack high quality genomic resources. Amphibians are a case in point as they are highly studied for AMPs (Wang, Li and Wang, 2016) but since there are few whole genome sequences for amphibians their AMP repertoires have not been compiled into reference proteomes. This was a limitation in chapter 4 where reference proteomes were used for benchmarking in order to ensure that a complete set of AMP and non-AMP sequences were used. Therefore, targeted genome sequencing to increase diversity of available organisms, in combination with effort to discover AMPs in these unrepresented organisms, can greatly improve the utility of benchmarking methods suggested in chapter 3.

Another potential limitation of benchmarking on whole proteomes (suggested in chapter 3) is the likelihood that AMP classification in these resources is incomplete. Even in organisms with the best characterised AMP repertoires it is highly likely that additional AMPs remain to be discovered. This means that benchmarking based on these proteomes is likely to result in inflated estimates of the false positive rate.

To optimise the usability for AMP prediction models in a genome-scanning context, a future predictor that can distinguish between the components of an AMP precursor sequence could potentially achieve higher performance. This is because the N-terminal signal peptide, mature AMP region and C-terminal region all likely have different physicochemical properties. If an input sequence could be subdivided into these components it would be possible to calculate their physicochemical properties separately and provide them as input to a composite predictor. Partitioning the input sequence precisely is currently an unsolved problem, however, very accurate methods already exist to identify the signal peptide (Teufel *et al.*, 2022). Pro-peptide cleavage can also be predicted using ProP (Duckert, Brunak and Blom, 2004) or MatureP (Orfanoudaki *et al.*, 2017), and the AMP mature peptide is normally between 20 and 30 amino acids in length. Thus an approximate partitioning would be possible and might still provide improved performance over existing models which treat the entire sequence as a single entity. A very simple option to generate a composite predictor would be to aggregate features from all three regions. Alternatively, separate models could be trained on each segment based on Swiss-Prot data where this is known (see Figure 3.1) and their results aggregated to form an ensemble prediction.

5.3 Conclusion

In summary, through the generation of the novel AMP predictor, ampir, this thesis demonstrated a need for developers to integrate greater knowledge of AMP protein structure and typical use-cases in the development and evaluation of AMP prediction models. Specifically, chapter 2 and 3 encompass a need to achieve a better match between data used for developing and testing AMP predictors, and the data that are present in real datasets such as proteomes. The two key aspects of this are composition (the types of molecules present) and balance (the ratio of AMPs and non-AMPs). Neither is trivial to achieve, however, this thesis proposed advances on current practices. These recommendations include the exclusive inclusion of precursor protein sequences in the training and testing data for AMP predictors. Furthermore, benchmarking should include tests on highly imbalanced data and report statistics sensitive to balance to reflect the fact that AMPs typically comprise a very small fraction of a typical genome. Finally, AMP predictors should use benchmark datasets that are compositionally similar to realistic input data (e.g. proteomes) as an additional benchmarking procedure.

Chapter 4 found that machine learning based AMP predictors are more effective at discovering AMPs on a genome-wide scale in taxonomically distant organisms compared to homology-based methods. However, AMP predictors must be optimised for genome-wide scanning (see chapters 2 and 3) for this to hold true. Nevertheless, a machine learning based AMP prediction approach is a valuable method for AMP discovery, especially for organisms belonging to taxa that are not well represented in AMPs.

Finally, ampir is available as an open source and well documented package in R, and also as a convenient web server. It can therefore be used by the wider community to aid AMP discovery, as well as serve as a reference point for future AMP predictors that wish to specialise in genome-wide scanning.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990) 'Basic local alignment search tool', *Journal of Molecular Biology,* 215(3), pp. 403-410, DOI: 10.1016/S0022-2836(05)80360-2.

Andrä, J., Herbst, R. and Leippe, M. (2003) 'Amoebapores, archaic effector peptides of protozoan origin, are discharged into phagosomes and kill bacteria by permeabilizing their membranes', *Developmental & Comparative Immunology,* 27(4), pp. 291-304, DOI: 10.1016/S0145-305X(02)00106-4.

Anselme, C., Pérez-Brocal, V., Vallier, A., Vincent-Monegat, C., Charif, D., Latorre, A., Moya, A. and Heddi, A. (2008) 'Identification of the weevil immune genes and their expression in the bacteriome tissue', *BMC Biology,* 6, pp. 43, DOI: 10.1186/1741-7007-6-43.

Aronica, P. G. A., Reid, L. M., Desai, N., Li, J., Fox, S. J., Yadahalli, S., Essex, J. W. and Verma, C. S. (2021) 'Computational methods and tools in antimicrobial peptide research', *Journal of Chemical Information and Modeling,* 61(7), pp. 3172-3196, DOI: 10.1021/acs.jcim.1c00175.

Bairoch, A. and Apweiler, R. (2000) 'The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000', *Nucleic Acids Research,* 28(1), pp. 45-48, DOI: 10.1093/nar/28.1.45.

Baltzer, S. A. and Brown, M. H. (2011) 'Antimicrobial peptides: promising alternatives to conventional antibiotics', *Journal of Molecular Microbiology and Biotechnology,* 20(4), pp. 228-235, DOI: 10.1159/000331009.

Bergel, A., Bañados, F., Robbes, R. and Röthlisberger, D. (2012) 'Spy: A flexible code profiling framework', *Computer Languages, Systems & Structures,* 38(1), pp. 16-28, DOI: 10.1016/j.cl.2011.10.002.

Besse, A., Peduzzi, J., Rebuffat, S. and Carré-Mlouka, A. (2015) 'Antimicrobial peptides and proteins in the face of extremes: Lessons from archaeocins', *Biochimie,* 118, pp. 344-355, DOI: 10.1016/j.biochi.2015.06.004.

Bhadra, P., Yan, J., Li, J., Fong, S. and Siu, S. W. I. (2018) 'AmPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest', *Scientific Reports,* 8(1), pp. 1697, DOI: 10.1038/s41598-018-19752-w.

Boman, H. G. (1995) 'Peptide antibiotics and their role in innate immunity', *Annual Review of Immunology,* 13, pp. 61-92, DOI: 10.1146/annurev.iy.13.040195.000425.

Bosch, T. C. G. (2013) 'Cnidarian-microbe interactions and the origin of innate immunity in metazoans', *Annual Review of Microbiology,* 67, pp. 499-518, DOI: 10.1146/annurev-micro-092412-155626.

Bosch, T. C. G. (2014) 'Rethinking the role of immunity: lessons from *Hydra*', *Trends in Immunology,* 35(10), pp. 495-502, DOI: 10.1016/j.it.2014.07.008.

Breiman, L. (2001) 'Random Forests', *Springer Science and Business Media LLC*, DOI: 10.1023/a:1010933404324.

Burdukiewicz, M., Sidorczuk, K., Rafacz, D., Pietluch, F., Chilimoniuk, J., Rödiger, S. and Gagat, P. (2020) 'Proteomic screening for prediction and design of antimicrobial peptides with AmpGram', *International Journal of Molecular Sciences,* 21(12), DOI: 10.3390/ijms21124310.

Cai, B. and Jiang, X. (2016) 'Computational methods for ubiquitination site prediction using physicochemical properties of protein sequences', *BMC Bioinformatics,* 17, pp. 116, DOI: 10.1186/s12859-016-0959-z.

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K. and Madden, T. L. (2009) 'BLAST+: architecture and applications', *BMC Bioinformatics,* 10, pp. 421, DOI: 10.1186/1471-2105-10-421.

Chai, J., Chen, X., Ye, T., Zeng, B., Zeng, Q., Wu, J., Kascakova, B., Martins, L. A., Prudnikova, T., Smatanova, I. K., Kotsyfakis, M. and Xu, X. (2021) 'Characterization and functional analysis of cathelicidin-MH, a novel frog-derived peptide with anti-septicemic properties', *eLife,* 10, DOI: 10.7554/eLife.64411.

Chan, D. (2020) 'Sunsetting Mercurial support in Bitbucket', *Bitbucket blog*, DOI. Available at: https://bitbucket.org/blog/sunsetting-mercurial-support-in-bitbucket

Chang, W., Cheng, J., Allaire, J. J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A. and Borges, B. (2021) *shiny: Web Application Framework for R*: R. Available at: https://CRAN.R-project.org/package=shiny.

Charlesworth, J. C. and Burns, B. P. (2015) 'Untapped resources: biotechnological potential of peptides and secondary metabolites in archaea', *Archaea,* 2015, pp. 282035, DOI: 10.1155/2015/282035.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002) 'SMOTE: Synthetic Minority Over-sampling Technique', *The Journal of artificial intelligence research,* 16, pp. 321-357, DOI: 10.1613/jair.953.

Chen, C. H. and Lu, T. K. (2020) 'Development and challenges of antimicrobial peptides for therapeutic applications', *Antibiotics (Basel, Switzerland),* 9(1), DOI: 10.3390/antibiotics9010024.

Chen, J., Lin, Y.-F., Chen, J.-H., Chen, X. and Lin, Z.-H. (2021) 'Molecular characterization of cathelicidin in tiger frog (*Hoplobatrachus rugulosus*): Antimicrobial activity and immunomodulatory activity', *Comparative Biochemistry and Physiology. Toxicology & Pharmacology,* 247, pp. 109072, DOI: 10.1016/j.cbpc.2021.109072.

Chen, R.-C., Dewi, C., Huang, S.-W. and Caraka, R. E. (2020) 'Selecting critical features for data classification based on machine learning methods', *Journal of big data,* 7(1), pp. 52, DOI: 10.1186/s40537-020-00327-4.

Chicco, D. (2017) 'Ten quick tips for machine learning in computational biology', *BioData mining,* 10, pp. 35, DOI: 10.1186/s13040-017-0155-3.

Chicco, D. and Jurman, G. (2020) 'The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation', *BMC Genomics,* 21(1), pp. 6, DOI: 10.1186/s12864-019-6413-7.

Chicco, D., Tötsch, N. and Jurman, G. (2021) 'The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation', *BioData mining,* 14(1), pp. 13, DOI: 10.1186/s13040-021-00244-z.

Chou, K.-C. (2001) 'Prediction of protein cellular attributes using pseudo-amino acid composition', *Proteins,* 43(3), pp. 246-255, DOI: 10.1002/prot.1035.

Chou, K.-C. (2009) 'Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology', *Int. Journal of Clinical Pharmacology and Therapeutics,* 6(4), pp. 262-274, DOI: 10.2174/157016409789973707.

Chou, K.-C. (2011) 'Some remarks on protein attribute prediction and pseudo amino acid composition', *Journal of Theoretical Biology,* 273(1), pp. 236-247, DOI: 10.1016/j.jtbi.2010.12.024.

Cogen, A. L., Yamasaki, K., Muto, J., Sanchez, K. M., Crotty Alexander, L., Tanios, J., Lai, Y., Kim, J. E., Nizet, V. and Gallo, R. L. (2010) '*Staphylococcus epidermidis* antimicrobial delta-toxin (phenol-soluble modulin-gamma) cooperates with host antimicrobial peptides to kill group A *Streptococcus*', *Plos One,* 5(1), pp. e8557, DOI: 10.1371/journal.pone.0008557.

Davis, J. and Goadrich, M. 'The relationship between Precision-Recall and ROC curves'. *the 23rd international conference*, 2006/06/25/. New York, New York, USA: ACM Press, 233-240.

Destoumieux-Garzón, D., Rosa, R. D., Schmitt, P., Barreto, C., Vidal-Dupiol, J., Mitta, G., Gueguen, Y. and Bachère, E. (2016) 'Antimicrobial peptides in marine invertebrate health and disease', *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences,* 371(1695), DOI: 10.1098/rstb.2015.0300.

Dodd, L. E. and Pepe, M. S. (2003) 'Partial AUC estimation and regression', *Biometrics,* 59(3), pp. 614-623, DOI: https://doi.org/10.1111/1541-0420.00071.

Dong, M., Kwok, S. H., Humble, J. L., Liang, Y., Tang, S. W., Tang, K. H., Tse, M. K., Lei, J. H., Ramalingam, R., Koohi-Moghadam, M., Au, D. W. T., Sun, H. and Lam, Y. W. (2021) 'BING, a novel antimicrobial peptide isolated from Japanese medaka plasma, targets bacterial envelope stress response by suppressing cpxR expression', *Scientific Reports,* 11(1), pp. 12219, DOI: 10.1038/s41598-021-91765-4.

Du, Z.-Q., Wang, Y., Ma, H.-Y., Shen, X.-L., Wang, K., Du, J., Yu, X.-D., Fang, W.-H. and Li, X.-C. (2019) 'A new crustin homologue (SpCrus6) involved in the antimicrobial and antiviral innate immunity in mud crab, *Scylla paramamosain*', *Fish & Shellfish Immunology,* 84, pp. 733-743, DOI: 10.1016/j.fsi.2018.10.072.

Duckert, P., Brunak, S. and Blom, N. (2004) 'Prediction of proprotein convertase cleavage sites', *Protein Engineering, Design & Selection,* 17(1), pp. 107-112, DOI: 10.1093/protein/gzh013.

Duwadi, D., Shrestha, A., Yilma, B., Kozlovski, I., Sa-Eed, M., Dahal, N. and Jukosky, J. (2018) 'Identification and screening of potent antimicrobial peptides in arthropod genomes', *Peptides,* 103, pp. 26-30, DOI: 10.1016/j.peptides.2018.01.017.

Eberl, G. (2010) 'A new vision of immunity: homeostasis of the superorganism', *Mucosal Immunology,* 3(5), pp. 450-460, DOI: 10.1038/mi.2010.20.

Eisenberg, D., Weiss, R. M. and Terwilliger, T. C. (1982) 'The helical hydrophobic moment: a measure of the amphiphilicity of a helix', *Nature,* 299(5881), pp. 371-374, DOI: 10.1038/299371a0.

Essig, A., Hofmann, D., Münch, D., Gayathri, S., Künzler, M., Kallio, P. T., Sahl, H.-G., Wider, G., Schneider, T. and Aebi, M. (2014) 'Copsin, a novel peptide-based fungal antibiotic interfering with the peptidoglycan synthesis', *The Journal of Biological Chemistry,* 289(50), pp. 34953-34964, DOI: 10.1074/jbc.M114.599878.

Fan, L., Sun, J., Zhou, M., Zhou, J., Lao, X., Zheng, H. and Xu, H. (2016) 'DRAMP: a comprehensive data repository of antimicrobial peptides', *Scientific Reports,* 6, pp. 24482, DOI: 10.1038/srep24482.

Fawcett, T. (2006) 'An introduction to ROC analysis', *Pattern recognition letters,* 27(8), pp. 861-874, DOI: 10.1016/j.patrec.2005.10.010.

Fernandes, F. C., Rigden, D. J. and Franco, O. L. (2012) 'Prediction of antimicrobial peptides based on the adaptive neuro-fuzzy inference system application', *Biopolymers,* 98(4), pp. 280-287, DOI: 10.1002/bip.22066.

Fingerhut, L. C. H. W., Miller, D. J., Strugnell, J. M., Daly, N. L. and Cooke, I. R. (2020) 'ampir: an R package for fast genome-wide prediction of antimicrobial peptides', *Bioinformatics*, DOI: 10.1093/bioinformatics/btaa653.

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), pp. 1148-1155, DOI: 10.1093/bioinformatics/btm068.

Franzenburg, S., Walter, J., Künzel, S., Wang, J., Baines, J. F., Bosch, T. C. G. and Fraune, S. (2013) 'Distinct antimicrobial peptide expression determines host species-specific bacterial associations', *Proceedings of the National Academy of Sciences of the United States of America,* 110(39), pp. E3730-8, DOI: 10.1073/pnas.1304960110.

Fraune, S., Anton-Erxleben, F., Augustin, R., Franzenburg, S., Knop, M., Schröder, K., Willoweit-Ohl, D. and Bosch, T. C. G. (2015) 'Bacteria-bacteria interactions within the microbiota of the ancestral metazoan *Hydra* contribute to fungal resistance', *The ISME Journal,* 9(7), pp. 1543-1556, DOI: 10.1038/ismej.2014.239.

Fraune, S., Augustin, R., Anton-Erxleben, F., Wittlieb, J., Gelhaus, C., Klimovich, V. B., Samoilovich, M. P. and Bosch, T. C. G. (2010) 'In an early branching metazoan, bacterial colonization of the embryo is controlled by maternal antimicrobial peptides', *Proceedings of the National Academy of Sciences of the United States of America,* 107(42), pp. 18067-18072, DOI: 10.1073/pnas.1008573107.

Fu, H., Cao, Z., Li, M. and Wang, S. (2020) 'ACEP: improving antimicrobial peptides recognition through automatic feature fusion and amino acid embedding', *BMC Genomics,* 21(1), pp. 597, DOI: 10.1186/s12864-020-06978-0.

Gabere, M. N. and Noble, W. S. (2017) 'Empirical comparison of web-based antimicrobial peptide prediction tools', *Bioinformatics,* 33(13), pp. 1921-1929, DOI: 10.1093/bioinformatics/btx081.

Gallo, R. L. and Huttner, K. M. (1998) 'Antimicrobial peptides: an emerging concept in cutaneous biology', *The Journal of Investigative Dermatology,* 111(5), pp. 739-743, DOI: 10.1046/j.1523-1747.1998.00361.x.

Gill, S. R., Pop, M., Deboy, R. T., Eckburg, P. B., Turnbaugh, P. J., Samuel, B. S., Gordon, J. I., Relman, D. A., Fraser-Liggett, C. M. and Nelson, K. E. (2006) 'Metagenomic analysis of the human distal gut microbiome', *Science,* 312(5778), pp. 1355-1359, DOI: 10.1126/science.1124234.

Gong, Z., Pei, X., Ren, S., Chen, X., Wang, L., Ma, C., Xi, X., Chen, T., Shaw, C. and Zhou, M. (2020) 'Identification and rational design of a novel antibacterial peptide Dermaseptin-AC from the skin secretion of the red-eyed tree frog *Agalychnis callidryas*', *Antibiotics (Basel, Switzerland),* 9(5), DOI: 10.3390/antibiotics9050243.

González-García, M., Rodríguez, A., Alba, A., Vázquez, A. A., Morales-Vicente, F. E., Pérez-Erviti, J., Spellerberg, B., Stenger, S., Grieshober, M., Conzelmann, C., Münch, J., Raber, H., Kubiczek, D., Rosenau, F., Wiese, S., Ständker, L. and Otero-González, A. (2020) 'New antibacterial peptides from the freshwater mollusk *Pomacea poeyana* (Pilsbry, 1927)', *Biomolecules,* 10(11), DOI: 10.3390/biom10111473.

Hall, T. J., McQuillan, C., Finlay, E. K., O'Farrelly, C., Fair, S. and Meade, K. G. (2017) 'Comparative genomic identification and validation of β-defensin genes in the *Ovis aries* genome', *BMC Genomics,* 18(1), pp. 278, DOI: 10.1186/s12864-017-3666-x.

Hancock, R. E. (1997) 'Peptide antibiotics', *The Lancet,* 349(9049), pp. 418-422, DOI: 10.1016/S0140-6736(97)80051-7.

Hancock, R. E. and Chapple, D. S. (1999) 'Peptide antibiotics', *Antimicrobial Agents and Chemotherapy,* 43(6), pp. 1317-1323, DOI: 10.1128/AAC.43.6.1317.

Hanson, M. A., Lemaitre, B. and Unckless, R. L. (2019) 'Dynamic evolution of antimicrobial peptides underscores trade-offs between immunity and ecological Fitness', *Frontiers in immunology,* 10, pp. 2620, DOI: 10.3389/fimmu.2019.02620.

Hassan, M. F., Qutb, A. M. and Dong, W. (2021) 'Prediction and activity of a cationic α-helix antimicrobial peptide ZM-804 from maize', *International Journal of Molecular Sciences,* 22(5), DOI: 10.3390/ijms22052643.

Hawkins, D. M. (2004) 'The problem of overfitting', *Journal of Chemical Information and Computer Sciences,* 44(1), pp. 1-12, DOI: 10.1021/ci0342472.

Hayashida, P. Y. and da Silva Junior, P. I. (2021) 'Insights into antimicrobial peptides from *Limacus flavus* mucus', *Current Microbiology,* 78(8), pp. 2970-2979, DOI: 10.1007/s00284-021-02552-3.

He, D., Cao, Z., Zhang, R. and Li, W. (2021) 'Molecular cloning and functional identification of the antimicrobial peptide gene Ctri9594 from the venom of the scorpion *Chaerilus tricostatus*', *Antibiotics (Basel, Switzerland),* 10(8), DOI: 10.3390/antibiotics10080896.

He, H. and Garcia, E. A. (2009) 'Learning from imbalanced data', *IEEE transactions on knowledge and data engineering,* 21(9), pp. 1263-1284, DOI: 10.1109/TKDE.2008.239.

Hibbing, M. E., Fuqua, C., Parsek, M. R. and Peterson, S. B. (2010) 'Bacterial competition: surviving and thriving in the microbial jungle', *Nature Reviews. Microbiology,* 8(1), pp. 15-25, DOI: 10.1038/nrmicro2259.

Hilton, M., Nelson, N., Tunnell, T., Marinov, D. and Dig, D. 'Trade-offs in continuous integration: assurance, security, and flexibility'. *the 2017 11th Joint Meeting*, 2017/09/04/. New York, New York, USA: ACM Press, 197-207.

Höng, K., Austerlitz, T., Bohlmann, T. and Bohlmann, H. (2021) 'The thionin family of antimicrobial peptides', *Plos One,* 16(7), pp. e0254549, DOI: 10.1371/journal.pone.0254549.

Hooper, L. V., Littman, D. R. and Macpherson, A. J. (2012) 'Interactions between the microbiota and the immune system', *Science,* 336(6086), pp. 1268-1273, DOI: 10.1126/science.1223490.

Huang, T., Gu, W., Wang, B., Zhang, Y., Cui, L., Yao, Z., Zhao, C. and Xu, G. (2019) 'Identification and expression of the hepcidin gene from brown trout (*Salmo trutta*) and functional analysis of its synthetic peptide', *Fish & Shellfish Immunology,* 87, pp. 243-253, DOI: 10.1016/j.fsi.2019.01.020.

Huttner, K. M., Lambeth, M. R., Burkin, H. R., Burkin, D. J. and Broad, T. E. (1998) 'Localization and genomic organization of sheep antimicrobial peptide genes', *Gene,* 206(1), pp. 85-91, DOI: 10.1016/S0378-1119(97)00569-6.

Ince, D. C., Hatton, L. and Graham-Cumming, J. (2012) 'The case for open computer programs', *Nature,* 482(7386), pp. 485-488, DOI: 10.1038/nature10836.

Innan, H. and Kondrashov, F. (2010) 'The evolution of gene duplications: classifying and distinguishing between models', *Nature Reviews. Genetics,* 11(2), pp. 97-108, DOI: 10.1038/nrg2689.

Javan, R. R., van Tonder, A. J., King, J. P., Harrold, C. L. and Brueggemann, A. B. (2018) 'Genome sequencing reveals a large and diverse repertoire of antimicrobial peptides', *Frontiers in microbiology,* 9, pp. 2012, DOI: 10.3389/fmicb.2018.02012.

Jhong, J.-H., Chi, Y.-H., Li, W.-C., Lin, T.-H., Huang, K.-Y. and Lee, T.-Y. (2019) 'dbAMP: an integrated resource for exploring antimicrobial peptides with functional activities and physicochemical properties on transcriptome and proteome data', *Nucleic Acids Research,* 47(D1), pp. D285-D297, DOI: 10.1093/nar/gky1030.

Jiang, Y., Wu, Y., Wang, T., Chen, X., Zhou, M., Ma, C., Xi, X., Zhang, Y., Chen, T., Shaw, C. and Wang, L. (2020) 'Brevinin-1GHd: a novel *Hylarana guentheri* skin secretion-derived Brevinin-1 type peptide with antimicrobial and anticancer therapeutic potential', *Bioscience reports,* 40(5), DOI: 10.1042/BSR20200019.

Joseph, S., Karnik, S., Nilawe, P., Jayaraman, V. K. and Idicula-Thomas, S. (2012) 'ClassAMP: a prediction tool for classification of antimicrobial peptides', *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 9(5), pp. 1535-1538, DOI: 10.1109/TCBB.2012.89.

Kang, X., Dong, F., Shi, C., Liu, S., Sun, J., Chen, J., Li, H., Xu, H., Lao, X. and Zheng, H. (2019) 'DRAMP 2.0, an updated data repository of antimicrobial peptides', *Scientific data,* 6(1), pp. 148, DOI: 10.1038/s41597-019-0154-y.

Kavousi, K., Bagheri, M., Behrouzi, S., Vafadar, S., Atanaki, F. F., Lotfabadi, B. T., Ariaeenejad, S., Shockravi, A. and Moosavi-Movahedi, A. A. (2020) 'IAMPE: NMR-assisted computational prediction of antimicrobial peptides', *Journal of Chemical Information and Modeling,* 60(10), pp. 4691-4701, DOI: 10.1021/acs.jcim.0c00841.

Khamis, A. M., Essack, M., Gao, X. and Bajic, V. B. (2015) 'Distinct profiling of antimicrobial peptide families', *Bioinformatics,* 31(6), pp. 849-856, DOI: 10.1093/bioinformatics/btu738.

Khosravian, M., Faramarzi, F. K., Beigi, M. M., Behbahani, M. and Mohabatkar, H. (2013) 'Predicting antibacterial peptides by the concept of Chou's pseudo-amino acid composition and machine learning methods', *Protein and Peptide Letters,* 20(2), pp. 180-186, DOI: 10.2174/092986613804725307.

Kim, D., Soundrarajan, N., Lee, J., Cho, H.-S., Choi, M., Cha, S.-Y., Ahn, B., Jeon, H., Le, M. T., Song, H., Kim, J.-H. and Park, C. (2017) 'Genomewide analysis of the antimicrobial peptides in *Python bivittatus* and characterization of cathelicidins with potent antimicrobial activity and low cytotoxicity', *Antimicrobial Agents and Chemotherapy,* 61(9), DOI: 10.1128/AAC.00530-17.

Kohavi, R. (1995) 'A study of cross-validation and bootstrap for accuracy estimation and model selection', *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2 IJCAI'95*. Montreal, Canada: Morgan Kaufmann Publishers Inc., pp. 1137–1143.

Krassowski, M. (2021) *krassowski/complex-upset*: Zenodo. Available at: http://doi.org/10.5281/zenodo.3700590.

Kuhn, M. (2008) 'Building predictive models in R using the caret package', *Journal of Statistical Software,* 28(5), DOI: 10.18637/jss.v028.i05.

Kuhn, M. (2019) 'The caret Package', *The caret Package github.io bookdown*, DOI. Available at: https://topepo.github.io/caret/

Kumar, R., Jangir, P. K., Das, J., Taneja, B. and Sharma, R. (2017a) 'Genome analysis of *Staphylococcus capitis* TE8 reveals repertoire of antimicrobial peptides and adaptation strategies for growth on human skin', *Scientific Reports,* 7(1), pp. 10447, DOI: 10.1038/s41598-017-11020-7.

Kumar, S. and Dudley, J. (2007) 'Bioinformatics software for biologists in the genomics era', *Bioinformatics,* 23(14), pp. 1713-1717, DOI: 10.1093/bioinformatics/btm239.

Kumar, S., Stecher, G., Suleski, M. and Hedges, S. B. (2017b) 'Timetree: A resource for timelines, timetrees, and divergence times', *Molecular Biology and Evolution,* 34(7), pp. 1812-1819, DOI: 10.1093/molbev/msx116.

Lata, S., Mishra, N. K. and Raghava, G. P. S. (2010) 'AntiBP2: improved version of antibacterial peptide prediction', *BMC Bioinformatics,* 11 Suppl 1, pp. S19, DOI: 10.1186/1471-2105-11-S1-S19.

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

Lawrence, T. J., Carper, D. L., Spangler, M. K., Carrell, A. A., Rush, T. A., Minter, S. J., Weston, D. J. and Labbé, J. L. (2020) 'amPEPpy 1.0: A portable and accurate antimicrobial peptide prediction tool', *Bioinformatics*, DOI: 10.1093/bioinformatics/btaa917.

Lazzaro, B. P., Zasloff, M. and Rolff, J. (2020) 'Antimicrobial peptides: Application informed by evolution', *Science,* 368(6490), DOI: 10.1126/science.aau5480.

Lee, J. H., Chung, H., Shin, Y. P., Kim, I.-W., Natarajan, S., Veerappan, K., Seo, M., Park, J. and Hwang, J. S. (2020a) 'Transcriptome analysis of *Psacothea hilaris*: De Novo Assembly and antimicrobial peptide prediction', *Insects,* 11(10), DOI: 10.3390/insects11100676.

Lee, J. H., Chung, H., Shin, Y. P., Kim, M.-A., Natarajan, S., Veerappan, K., Kim, S. H., Park, J. and Hwang, J. S. (2020b) 'Deciphering novel antimicrobial peptides from the transcriptome of *Papilio xuthus*', *Insects,* 11(11), DOI: 10.3390/insects11110776.

Lee, J. H., Chung, H., Shin, Y. P., Kim, M.-A., Natarajan, S., Veerappan, K., Kim, S. H., Park, J. and Hwang, J. S. (2021a) 'Uncovering antimicrobial peptide from *Zophobas atratus* using transcriptome Analysis', *International journal of peptide research and therapeutics,* 27(3), pp. 1827-1835, DOI: 10.1007/s10989-021-10213-z.

Lee, K.-W., Kim, J.-G., Veerappan, K., Chung, H., Natarajan, S., Kim, K.-Y. and Park, J. (2021b) 'Utilizing red spotted apollo butterfly transcriptome to identify antimicrobial peptide candidates against *Porphyromonas gingivalis*', *Insects,* 12(5), DOI: 10.3390/insects12050466.

Leptihn, S., Har, J. Y., Wohland, T. and Ding, J. L. (2010) 'Correlation of charge, hydrophobicity, and structure with antimicrobial activity of S1 and MIRIAM peptides', *Biochemistry,* 49(43), pp. 9161-9170, DOI: 10.1021/bi1011578.

Lex, A., Gehlenborg, N., Strobelt, H., Vuillemot, R. and Pfister, H. (2014) 'Upset: visualization of intersecting sets', *IEEE transactions on visualization and computer graphics,* 20(12), pp. 1983-1992, DOI: 10.1109/TVCG.2014.2346248.

Ley, R. E., Bäckhed, F., Turnbaugh, P., Lozupone, C. A., Knight, R. D. and Gordon, J. I. (2005) 'Obesity alters gut microbial ecology', *Proceedings of the National Academy of Sciences of the United States of America,* 102(31), pp. 11070-11075, DOI: 10.1073/pnas.0504978102.

Li, B., Lyu, P., Xie, S., Qin, H., Pu, W., Xu, H., Chen, T., Shaw, C., Ge, L. and Kwok, H. F. (2019) 'LFB: A novel antimicrobial Brevinin-like peptide from the skin secretion of the Fujian large headed frog, *Limnonectes fujianensi*', *Biomolecules,* 9(6), DOI: 10.3390/biom9060242.

Li, C., Sutherland, D., Hammond, S. A., Yang, C., Taho, F., Bergman, L., Houston, S., Warren, R. L., Wong, T., Hoang, L. M. N., Cameron, C. E., Helbing, C. C. and Birol, I. (2020) 'AMPlify: attentive deep learning model for discovery of novel antimicrobial peptides effective against WHO priority pathogens', *BioRxiv*, DOI: 10.1101/2020.06.16.155705.

Li, F., Gao, Z., Wang, K., Zhao, Y., Wang, H., Zhao, M., Zhao, Y., Bai, L., Yu, Z. and Yang, X. (2021a) 'A novel defensin-like peptide contributing to antimicrobial and antioxidant capacity of the tick *Dermacentor silvarum* (Acari: Ixodidae)', *Experimental & Applied Acarology,* 83(2), pp. 271-283, DOI: 10.1007/s10493-020-00584-1.

Li, L.-L., Liu, T.-L., Wu, P., Du, N.-Y., Tian, L.-H. and Hou, Z.-J. (2021b) 'Molecular identification and antibacterial activity analysis of blue fox (*Vulpes lagopus*) β-defensins 108 and 122', *Animals : an open access journal from MDPI,* 11(7), DOI: 10.3390/ani11071857.

Li, W., Jaroszewski, L. and Godzik, A. (2001) 'Clustering of highly homologous sequences to reduce the size of large protein databases', *Bioinformatics,* 17(3), pp. 282-283, DOI: 10.1093/bioinformatics/17.3.282.

Lin, W. and Xu, D. (2016) 'Imbalanced multi-label learning for identifying antimicrobial peptides and their functional types', *Bioinformatics,* 32(24), pp. 3745-3752, DOI: 10.1093/bioinformatics/btw560.

Liscano, Y., Medina, L., Oñate-Garzón, J., Gúzman, F., Pickholz, M. and Delgado, J. P. (2021) 'In silico selection and evaluation of pugnins with antibacterial and anticancer activity using skin transcriptome of treefrog (*Boana pugnax*)', *Pharmaceutics,* 13(4), DOI: 10.3390/pharmaceutics13040578.

Liu, S., Bao, J., Lao, X. and Zheng, H. (2018) 'Novel 3D structure based model for activity prediction and design of antimicrobial peptides', *Scientific Reports,* 8(1), pp. 11189, DOI: 10.1038/s41598-018-29566-5.

Liu, S., Fan, L., Sun, J., Lao, X. and Zheng, H. (2017) 'Computational resources and tools for antimicrobial peptides', *Journal of Peptide Science,* 23(1), pp. 4-12, DOI: 10.1002/psc.2947.

Login, F. H., Balmand, S., Vallier, A., Vincent-Monégat, C., Vigneron, A., Weiss-Gayet, M., Rochat, D. and Heddi, A. (2011) 'Antimicrobial peptides keep insect endosymbionts under control', *Science,* 334(6054), pp. 362-365, DOI: 10.1126/science.1209728.

Loo, M. P. J. v. d. (2014) 'The stringdist package for approximate string matching', *The R journal,* 6(1), pp. 111, DOI: 10.32614/RJ-2014-011.

Lüders, T., Birkemo, G. A., Fimland, G., Nissen-Meyer, J. and Nes, I. F. (2003) 'Strong synergy between a eukaryotic antimicrobial peptide and bacteriocins from lactic acid bacteria', *Applied and Environmental Microbiology,* 69(3), pp. 1797-1799, DOI: 10.1128/aem.69.3.1797-1799.2003.

Ma, Y., Guo, Z., Xia, B., Zhang, Y., Liu, X., Yu, Y., Tang, N., Tong, X., Wang, M., Ye, X., Feng, J., Chen, Y. and Wang, J. (2022) 'Identification of antimicrobial peptides from the human gut microbiome using deep learning', *Nature Biotechnology*, DOI: 10.1038/s41587-022-01226-0.

Manners, J. M. (2007) 'Hidden weapons of microbial destruction in plant genomes', *Genome Biology,* 8(9), pp. 225, DOI: 10.1186/gb-2007-8-9-225.

Margulis, L. (1993) *Symbiosis in cell evolution: microbial communities in the Archean and Proterozoic eons.* 2nd edn. New York: Freeman.

McFall-Ngai, M., Hadfield, M. G., Bosch, T. C. G., Carey, H. V., Domazet-Lošo, T., Douglas, A. E., Dubilier, N., Eberl, G., Fukami, T., Gilbert, S. F., Hentschel, U., King, N., Kjelleberg, S., Knoll, A. H., Kremer, N., Mazmanian, S. K., Metcalf, J. L., Nealson, K., Pierce, N. E., Rawls, J. F., Reid, A., Ruby, E. G., Rumpho, M., Sanders, J. G., Tautz, D. and Wernegreen, J. J. (2013) 'Animals in a bacterial world, a new imperative for the life sciences', *Proceedings of the National Academy of Sciences of the United States of America,* 110(9), pp. 3229-3236, DOI: 10.1073/pnas.1218525110.

Meher, P. K., Sahu, T. K., Saini, V. and Rao, A. R. (2017) 'Predicting antimicrobial peptides with improved accuracy by incorporating the compositional, physico-chemical and structural features into Chou's general PseAAC', *Scientific Reports,* 7, pp. 42362, DOI: 10.1038/srep42362.

Mergaert, P. (2018) 'Role of antimicrobial peptides in controlling symbiotic bacterial populations', *Natural Product Reports,* 35(4), pp. 336-356, DOI: 10.1039/c7np00056a.

Meyer, M. (2014) 'Continuous Integration and Its Tools', *IEEE Software,* 31(3), pp. 14-16, DOI: 10.1109/MS.2014.58.

Mitchell, A. L., Attwood, T. K., Babbitt, P. C., Blum, M., Bork, P., Bridge, A., Brown, S. D., Chang, H.-Y., El-Gebali, S., Fraser, M. I., Gough, J., Haft, D. R., Huang, H., Letunic, I., Lopez, R., Luciani, A., Madeira, F., Marchler-Bauer, A., Mi, H., Natale, D. A., Necci, M., Nuka, G., Orengo, C., Pandurangan, A. P., Paysan-Lafosse, T., Pesseat, S., Potter, S. C., Qureshi, M. A., Rawlings, N. D., Redaschi, N., Richardson, L. J., Rivoire, C., Salazar, G. A., Sangrador-Vegas, A., Sigrist, C. J. A., Sillitoe, I., Sutton, G. G., Thanki, N., Thomas, P. D., Tosatto, S. C. E., Yong, S.-Y. and Finn, R. D. (2019) 'InterPro in 2019: improving coverage, classification and access to protein sequence annotations', *Nucleic Acids Research,* 47(D1), pp. D351-D360, DOI: 10.1093/nar/gky1100.

Moravej, H., Moravej, Z., Yazdanparast, M., Heiat, M., Mirhosseini, A., Moosazadeh Moghaddam, M. and Mirnejad, R. (2018) 'Antimicrobial peptides: features, action, and their resistance mechanisms in bacteria', *Microbial Drug Resistance,* 24(6), pp. 747-767, DOI: 10.1089/mdr.2017.0392.

Moretta, A., Scieuzo, C., Petrone, A. M., Salvia, R., Manniello, M. D., Franco, A., Lucchetti, D., Vassallo, A., Vogel, H., Sgambato, A. and Falabella, P. (2021) 'Antimicrobial peptides: A new hope in biomedical and pharmaceutical fields', *Frontiers in cellular and infection microbiology,* 11, pp. 668632, DOI: 10.3389/fcimb.2021.668632.

Nakatsuji, T., Chen, T. H., Narala, S., Chun, K. A., Two, A. M., Yun, T., Shafiq, F., Kotol, P. F., Bouslimani, A., Melnik, A. V., Latif, H., Kim, J.-N., Lockhart, A., Artis, K., David, G., Taylor, P., Streib, J., Dorrestein, P. C., Grier, A., Gill, S. R., Zengler, K., Hata, T. R., Leung, D. Y. M. and Gallo, R. L. (2017) 'Antimicrobials from human skin commensal bacteria protect against *Staphylococcus aureus* and are deficient in atopic dermatitis', *Science Translational Medicine,* 9(378), DOI: 10.1126/scitranslmed.aah4680.

NCBI Resource Coordinators (2018) 'Database resources of the National Center for Biotechnology Information', *Nucleic Acids Research,* 46(D1), pp. D8-D13, DOI: 10.1093/nar/gkx1095.

Ng, X. Y., Rosdi, B. A. and Shahrudin, S. (2015) 'Prediction of antimicrobial peptides based on sequence alignment and support vector machine-pairwise algorithm utilizing LZ-complexity', *BioMed research international,* 2015, pp. 212715, DOI: 10.1155/2015/212715.

Nicolas, P., Vanhoye, D. and Amiche, M. (2003) 'Molecular strategies in biological evolution of antimicrobial peptides', *Peptides,* 24(11), pp. 1669-1680, DOI: 10.1016/j.peptides.2003.08.017.

Noble, W. S. (2006) 'What is a support vector machine?', *Nature Biotechnology,* 24(12), pp. 1565-1567, DOI: 10.1038/nbt1206-1565.

Ohtsuka, Y. and Inagaki, H. (2020) 'In silico identification and functional validation of linear cationic α-helical antimicrobial peptides in the ascidian *Ciona intestinalis*', *Scientific Reports,* 10(1), pp. 12619, DOI: 10.1038/s41598-020-69485-y.

Onime, L. A., Oyama, L. B., Thomas, B. J., Gani, J., Alexander, P., Waddams, K. E., Cookson, A., Fernandez-Fuentes, N., Creevey, C. J. and Huws, S. A. (2021) 'The rumen eukaryotome is a source of novel antimicrobial peptides with therapeutic potential', *BMC Microbiology,* 21(1), pp. 105, DOI: 10.1186/s12866-021-02172-8.

Orfanoudaki, G., Markaki, M., Chatzi, K., Tsamardinos, I. and Economou, A. (2017) 'MatureP: prediction of secreted proteins with exclusive information from their mature regions', *Scientific Reports,* 7(1), pp. 3263, DOI: 10.1038/s41598-017-03557-4.

Osorio, D., Rondon-Villarreal, P. and Torres, R. (2015) 'Peptides: A Package for Data Mining of Antimicrobial Peptides', *R Journal,* 7(1), pp. 4-14, DOI:

Pane, K., Durante, L., Crescenzi, O., Cafaro, V., Pizzo, E., Varcamonti, M., Zanfardino, A., Izzo, V., Di Donato, A. and Notomista, E. (2017) 'Antimicrobial potency of cationic antimicrobial peptides can be predicted from their amino acid composition: Application to the detection of "cryptic" antimicrobial peptides', *Journal of Theoretical Biology,* 419, pp. 254-265, DOI: 10.1016/j.jtbi.2017.02.012.

Panteleev, P. V., Tsarev, A. V., Safronova, V. N., Reznikova, O. V., Bolosov, I. A., Sychev, S. V., Shenkarev, Z. O. and Ovchinnikova, T. V. (2020) 'Structure elucidation and functional studies of a novel β-hairpin antimicrobial peptide from the marine polychaeta *Capitella teleta*', *Marine Drugs,* 18(12), DOI: 10.3390/md18120620.

Paradis, E. and Schliep, K. (2019) 'ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R', *Bioinformatics,* 35(3), pp. 526-528, DOI: 10.1093/bioinformatics/bty633.

Pasupuleti, M., Schmidtchen, A. and Malmsten, M. (2012) 'Antimicrobial peptides: key components of the innate immune system', *Critical reviews in biotechnology,* 32(2), pp. 143-171, DOI: 10.3109/07388551.2011.594423.

Pearson, W. R. (2013) 'An introduction to sequence similarity ("homology") searching', *Current Protocols in Bioinformatics,* Chapter 3, pp. Unit3.1, DOI: 10.1002/0471250953.bi0301s42.

Peel, E., Cheng, Y., Djordjevic, J. T., O'Meally, D., Thomas, M., Kuhn, M., Sorrell, T. C., Huston, W. M. and Belov, K. (2021) 'Koala cathelicidin PhciCath5 has antimicrobial activity, including against *Chlamydia pecorum*', *Plos One,* 16(4), pp. e0249658, DOI: 10.1371/journal.pone.0249658.

Pérez de la Lastra, J. M., Asensio-Calavia, P., González-Acosta, S., Baca-González, V. and Morales-delaNuez, A. (2021) 'Bioinformatic analysis of genome-predicted bat cathelicidins', *Molecules (Basel, Switzerland),* 26(6), pp. 1811, DOI: 10.3390/molecules26061811.

Pinheiro-Machado, E., Milkewitz Sandberg, T. O., Pihl, C., Hägglund, P. M. and Marzec, M. T. (2020) 'In silico approach to predict pancreatic β-cells classically secreted proteins', *Bioscience reports,* 40(2), DOI: 10.1042/BSR20193708.

Porto, W. F., Pires, Á. S. and Franco, O. L. (2012) 'CS-AMPPred: an updated SVM model for antimicrobial activity prediction in cysteine-stabilized peptides', *Plos One,* 7(12), pp. e51444, DOI: 10.1371/journal.pone.0051444.

Qureshi, A., Tandon, H. and Kumar, M. (2015) 'AVP-IC50 Pred: Multiple machine learning techniques-based prediction of peptide antiviral activity in terms of half maximal inhibitory concentration (IC50)', *Biopolymers,* 104(6), pp. 753-763, DOI: 10.1002/bip.22703.

R Core Team (2021) *R: A Language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Available at: http://www.R-project.org/.

Rádai, Z., Kiss, J. and Nagy, N. A. (2021) 'Taxonomic bias in AMP prediction of invertebrate peptides', *Scientific Reports,* 11(1), pp. 17924, DOI: 10.1038/s41598-021-97415-z.

Ramazi, S., Mohammadi, N., Allahverdi, A., Khalili, E. and Abdolmaleki, P. (2022) 'A review on antimicrobial peptides databases and the computational tools', *Database: the Journal of Biological Databases and Curation,* 2022, DOI: 10.1093/database/baac011.

Real, E., Rain, J.-C., Battaglia, V., Jallet, C., Perrin, P., Tordo, N., Chrisment, P., D'Alayer, J., Legrain, P. and Jacob, Y. (2004) 'Antiviral drug discovery strategy using combinatorial libraries of structurally constrained peptides', *Journal of Virology,* 78(14), pp. 7410-7417, DOI: 10.1128/JVI.78.14.7410-7417.2004.

Riley, M. A. and Wertz, J. E. (2002) 'Bacteriocins: evolution, ecology, and application', *Annual Review of Microbiology,* 56, pp. 117-137, DOI: 10.1146/annurev.micro.56.012302.161024.

Rončević, T., Gerdol, M., Spazzali, F., Florian, F., Mekinić, S., Tossi, A. and Pallavicini, A. (2018) 'Parallel identification of novel antimicrobial peptide sequences from multiple anuran species by targeted DNA sequencing', *BMC Genomics,* 19(1), pp. 827, DOI: 10.1186/s12864-018-5225-5.

Rončević, T., Puizina, J. and Tossi, A. (2019) 'Antimicrobial peptides as anti-infective agents in pre-post-antibiotic era?', *International Journal of Molecular Sciences,* 20(22), DOI: 10.3390/ijms20225713.

Rosenstiel, P., Philipp, E. E. R., Schreiber, S. and Bosch, T. C. G. (2009) 'Evolution and function of innate immune receptors--insights from marine invertebrates', *Journal of Innate Immunity,* 1(4), pp. 291-300, DOI: 10.1159/000211193.

RStudio Team (2021) *RStudio: Integrated Development Environment for R*. Boston, MA: RStudio, PBC. Available at: http://www.rstudio.com.

Saito, T. and Rehmsmeier, M. (2015) 'The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets', *Plos One,* 10(3), pp. e0118432, DOI: 10.1371/journal.pone.0118432.

Saito, T. and Rehmsmeier, M. (2017) 'Precrec: fast and accurate precision-recall and ROC curve calculations in R', *Bioinformatics,* 33(1), pp. 145-147, DOI: 10.1093/bioinformatics/btw570.

Salzman, N. H., Hung, K., Haribhai, D., Chu, H., Karlsson-Sjöberg, J., Amir, E., Teggatz, P., Barman, M., Hayward, M., Eastwood, D., Stoel, M., Zhou, Y., Sodergren, E., Weinstock, G. M., Bevins, C. L., Williams, C. B. and Bos, N. A. (2010) 'Enteric defensins are essential regulators of intestinal microbial ecology', *Nature Immunology,* 11(1), pp. 76-83, DOI: 10.1038/ni.1825.

Sangar, V., Blankenberg, D. J., Altman, N. and Lesk, A. M. (2007) 'Quantitative sequence-function relationships in proteins based on gene ontology', *BMC Bioinformatics,* 8, pp. 294, DOI: 10.1186/1471-2105-8-294.

Santos-Júnior, C. D., Pan, S., Zhao, X.-M. and Coelho, L. P. (2020) 'Macrel: antimicrobial peptide screening in genomes and metagenomes', *PeerJ,* 8, pp. e10555, DOI: 10.7717/peerj.10555.

Schmitt, P., Rosa, R. D. and Destoumieux-Garzón, D. (2016) 'An intimate link between antimicrobial peptide sequence diversity and binding to essential components of bacterial membranes', *Biochimica et Biophysica Acta,* 1858(5), pp. 958-970, DOI: 10.1016/j.bbamem.2015.10.011.

Sharma, R., Shrivastava, S., Kumar Singh, S., Kumar, A., Saxena, S. and Kumar Singh, R. (2021) 'AniAMPpred: artificial intelligence guided discovery of novel antimicrobial peptides in animal kingdom', *Briefings in Bioinformatics*, DOI: 10.1093/bib/bbab242.

Shen, B., Wei, K., Yang, J., Jing, F. and Zhang, J. (2021) 'Molecular characterization and functional analyses of a hepcidin gene from *Bostrychus sinensis*', *Aquaculture,* 544, pp. 737114, DOI: 10.1016/j.aquaculture.2021.737114.

Shinzato, C., Khalturin, K., Inoue, J., Zayasu, Y., Kanda, M., Kawamitsu, M., Yoshioka, Y., Yamashita, H., Suzuki, G. and Satoh, N. (2021) 'Eighteen coral genomes reveal the evolutionary origin of acropora strategies to accommodate environmental changes', *Molecular Biology and Evolution,* 38(1), pp. 16-30, DOI: 10.1093/molbev/msaa216.

Silverstein, K. A. T., Moskal, W. A., Wu, H. C., Underwood, B. A., Graham, M. A., Town, C. D. and VandenBosch, K. A. (2007) 'Small cysteine-rich peptides resembling antimicrobial peptides have been under-predicted in plants', *The Plant Journal: for Cell and Molecular Biology,* 51(2), pp. 262-280, DOI: 10.1111/j.1365-313X.2007.03136.x.

Sofaer, H. R., Hoeting, J. A. and Jarnevich, C. S. (2018) 'The area under the precision-recall curve as a performance metric for rare binary events', *Methods in Ecology and Evolution*, DOI: 10.1111/2041-210X.13140.

Tam, J. P., Wang, S., Wong, K. H. and Tan, W. L. (2015) 'Antimicrobial Peptides from Plants', *Pharmaceuticals (Basel, Switzerland),* 8(4), pp. 711-757, DOI: 10.3390/ph8040711.

Teufel, F., Almagro Armenteros, J. J., Johansen, A. R., Gíslason, M. H., Pihl, S. I., Tsirigos, K. D., Winther, O., Brunak, S., von Heijne, G. and Nielsen, H. (2022) 'SignalP 6.0 predicts all five types of signal peptides using protein language models', *Nature Biotechnology*, DOI: 10.1038/s41587-021-01156-3.

Thaiss, C. A., Zmora, N., Levy, M. and Elinav, E. (2016) 'The microbiome and innate immunity', *Nature,* 535(7610), pp. 65-74, DOI: 10.1038/nature18847.

Thakur, N., Qureshi, A. and Kumar, M. (2012) 'AVPpred: collection and prediction of highly effective antiviral peptides', *Nucleic Acids Research,* 40(Web Server issue), pp. W199-204, DOI: 10.1093/nar/gks450.

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 Research,* 38(Database issue), pp. D774-80, DOI: 10.1093/nar/gkp1021.

Tincu, J. A. and Taylor, S. W. (2004) 'Antimicrobial peptides from marine invertebrates', *Antimicrobial Agents and Chemotherapy,* 48(10), pp. 3645-3654, DOI: 10.1128/AAC.48.10.3645-3654.2004.

Torrent, M., Andreu, D., Nogués, V. M. and Boix, E. (2011) 'Connecting peptide physicochemical and antimicrobial properties by a rational prediction model', *Plos One,* 6(2), pp. e16968, DOI: 10.1371/journal.pone.0016968.

Unckless, R. L. and Lazzaro, B. P. (2016) 'The potential for adaptive maintenance of diversity in insect antimicrobial peptides', *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences,* 371(1695), DOI: 10.1098/rstb.2015.0291.

UniProt Consortium (2019) 'UniProt: a worldwide hub of protein knowledge', *Nucleic Acids Research,* 47(D1), pp. D506-D515, DOI: 10.1093/nar/gky1049.

UniProt Consortium (2021) 'UniProt: the universal protein knowledgebase in 2021', *Nucleic Acids Research,* 49(D1), pp. D480-D489, DOI: 10.1093/nar/gkaa1100.

Van de Velde, W., Zehirov, G., Szatmari, A., Debreczeny, M., Ishihara, H., Kevei, Z., Farkas, A., Mikulass, K., Nagy, A., Tiricz, H., Satiat-Jeunemaître, B., Alunni, B., Bourge, M., Kucho, K.-i., Abe, M., Kereszt, A., Maroti, G., Uchiumi, T., Kondorosi, E. and Mergaert, P. (2010) 'Plant peptides govern terminal differentiation of bacteria in symbiosis', *Science,* 327(5969), pp. 1122-1126, DOI: 10.1126/science.1184057.

Veltri, D., Kamath, U. and Shehu, A. (2017) 'Improving recognition of antimicrobial peptides and target selectivity through machine learning and genetic programming', *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 14(2), pp. 300-313, DOI: 10.1109/TCBB.2015.2462364.

Veltri, D., Kamath, U. and Shehu, A. (2018) 'Deep learning improves antimicrobial peptide recognition', *Bioinformatics,* 34(16), pp. 2740-2747, DOI: 10.1093/bioinformatics/bty179.

Vishnepolsky, B., Gabrielian, A., Rosenthal, A., Hurt, D. E., Tartakovsky, M., Managadze, G., Grigolava, M., Makhatadze, G. I. and Pirtskhalava, M. (2018) 'Predictive model of linear antimicrobial peptides active against gram-negative bacteria', *Journal of Chemical Information and Modeling,* 58(5), pp. 1141-1151, DOI: 10.1021/acs.jcim.8b00118.

Vishnepolsky, B. and Pirtskhalava, M. (2014) 'Prediction of linear cationic antimicrobial peptides based on characteristics responsible for their interaction with the membranes', *Journal of Chemical Information and Modeling,* 54(5), pp. 1512-1523, DOI: 10.1021/ci4007003.

Waghu, F. H., Barai, R. S., Gurung, P. and Idicula-Thomas, S. (2016) 'CAMPR3: a database on sequences, structures and signatures of antimicrobial peptides', *Nucleic Acids Research,* 44(D1), pp. D1094-7, DOI: 10.1093/nar/gkv1051.

Wang, D., Chen, X., Zhang, X., Li, J., Yi, Y., Bian, C., Shi, Q., Lin, H., Li, S., Zhang, Y. and You, X. (2019) 'Whole genome sequencing of the giant grouper (*Epinephelus* *lanceolatus*) and high-throughput screening of putative antimicrobial peptide genes', *Marine Drugs,* 17(9), DOI: 10.3390/md17090503.

Wang, G., Li, X. and Wang, Z. (2016) 'APD3: the antimicrobial peptide database as a tool for research and education', *Nucleic Acids Research,* 44(D1), pp. D1087-93, DOI: 10.1093/nar/gkv1278.

Wang, H., He, H., Chen, X., Zhou, M., Wei, M., Xi, X., Ma, C., Du, Q., Chen, T., Shaw, C. and Wang, L. (2020a) 'A novel antimicrobial peptide (Kassinatuerin-3) isolated from the skin secretion of the African frog, *Kassina senegalensis*', *Biology,* 9(7), DOI: 10.3390/biology9070148.

Wang, L.-G., Lam, T. T.-Y., Xu, S., Dai, Z., Zhou, L., Feng, T., Guo, P., Dunn, C. W., Jones, B. R., Bradley, T., Zhu, H., Guan, Y., Jiang, Y. and Yu, G. (2020b) 'Treeio: An R package for phylogenetic tree input and output with richly annotated and associated data', *Molecular Biology and Evolution,* 37(2), pp. 599-603, DOI: 10.1093/molbev/msz240.

Wang, M., Zhou, Z., Li, S., Zhu, W. and Hu, X. (2021) 'Identification and characterization of antimicrobial peptides from butterflies: an integrated bioinformatics and experimental study', *Frontiers in microbiology,* 12, pp. 720381, DOI: 10.3389/fmicb.2021.720381.

Wang, P., Ge, R., Liu, L., Xiao, X., Li, Y. and Cai, Y. (2017) 'Multi-label learning for predicting the activities of antimicrobial peptides', *Scientific Reports,* 7(1), pp. 2202, DOI: 10.1038/s41598-017-01986-9.

Wang, P., Hu, L., Liu, G., Jiang, N., Chen, X., Xu, J., Zheng, W., Li, L., Tan, M., Chen, Z., Song, H., Cai, Y.-D. and Chou, K.-C. (2011) 'Prediction of antimicrobial peptides based on sequence alignment and feature selection methods', *Plos One,* 6(4), pp. e18476, DOI: 10.1371/journal.pone.0018476.

Wang, Q., Xia, R., Ji, J. J., Zhu, Q., Li, X. P., Ma, Y. and Xu, Y. C. (2020c) 'Diversity of antimicrobial peptides in three partially sympatric frog species in northeast asia and implications for evolution', *Genes,* 11(2), DOI: 10.3390/genes11020158.

Wang, Z. and Wang, G. (2004) 'APD: the antimicrobial peptide database', *Nucleic Acids Research,* 32(Database issue), pp. D590-2, DOI: 10.1093/nar/gkh025.

Warren, W. C. and Hillier, L. W. and Marshall Graves, J. A. and Birney, E. and Ponting, C. P. and Grützner, F. and Belov, K. and Miller, W. and Clarke, L. and Chinwalla, A. T. and Yang, S.-P. and Heger, A. and Locke, D. P. and Miethke, P. and Waters, P. D. and Veyrunes, F. and Fulton, L. and Fulton, B. and Graves, T. and Wallis, J. and Puente, X. S. and López-Otín, C. and Ordóñez, G. R. and Eichler, E. E. and Chen, L. and Cheng, Z. and Deakin, J. E. and Alsop, A. and Thompson, K. and Kirby, P. and Papenfuss, A. T. and Wakefield, M. J. and Olender, T. and Lancet, D. and Huttley, G. A. and Smit, A. F. A. and Pask, A. and Temple-Smith, P. and Batzer, M. A. and Walker, J. A. and Konkel, M. K. and Harris, R. S. and Whittington, C. M. and Wong, E. S. W. and Gemmell, N. J. and Buschiazzo, E. and Vargas Jentzsch, I. M. and Merkel, A. and

Schmitz, J. and Zemann, A. and Churakov, G. and Ole Kriegs, J. and Brosius, J. and Murchison, E. P. and Sachidanandam, R. and Smith, C. and Hannon, G. J. and Tsend-Ayush, E. and McMillan, D. and Attenborough, R. and Rens, W. and Ferguson-Smith, M. and Lefèvre, C. M. and Sharp, J. A. and Nicholas, K. R. and Ray, D. A. and Kube, M. and Reinhardt, R. and Pringle, T. H. and Taylor, J. and Jones, R. C. and Nixon, B. and Dacheux, J.-L. and Niwa, H. and Sekita, Y. and Huang, X. and Stark, A. and Kheradpour, P. and Kellis, M. and Flicek, P. and Chen, Y. and Webber, C. and Hardison, R. and Nelson, J. and Hallsworth-Pepin, K. and Delehaunty, K. and Markovic, C. and Minx, P. and Feng, Y. and Kremitzki, C. and Mitreva, M. and Glasscock, J. and Wylie, T. and Wohldmann, P. and Thiru, P. and Nhan, M. N. and Pohl, C. S. and Smith, S. M. and Hou, S. and Renfree, M. B. and Mardis, E. R. and Wilson, R. K. and authors, A. l. o. and their affiliations appears at the end of the, p. (2008) 'Genome analysis of the platypus reveals unique signatures of evolution', *Nature,* 453(7192), pp. 175-183, DOI: 10.1038/nature06936.

Wehkamp, J., Salzman, N. H., Porter, E., Nuding, S., Weichenthal, M., Petras, R. E., Shen, B., Schaeffeler, E., Schwab, M., Linzmeier, R., Feathers, R. W., Chu, H., Lima, H., Fellermann, K., Ganz, T., Stange, E. F. and Bevins, C. L. (2005) 'Reduced Paneth cell alpha-defensins in ileal Crohn's disease', *Proceedings of the National Academy of Sciences of the United States of America,* 102(50), pp. 18129-18134, DOI: 10.1073/pnas.0505256102.

Weiss, G. M. (2004) 'Mining with rarity', *ACM SIGKDD Explorations Newsletter,* 6(1), pp. 7, DOI: 10.1145/1007730.1007734.

Whalen, S., Schreiber, J., Noble, W. S. and Pollard, K. S. (2022) 'Navigating the pitfalls of applying machine learning in genomics', *Nature Reviews. Genetics,* 23(3), pp. 169-181, DOI: 10.1038/s41576-021-00434-9.

Wickham, H. (2015) *R Packages: Organize, Test, Document, And Share Your Code.* 1 edn. Sebastopol, CA: O'reilly Media.

Wickham, H. (2019) *Advanced R, Second Edition (chapman & Hall/crc The R Series).* 2 edn.: Chapman And Hall/crc.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T., Miller, E., Bache, S., Müller, K., Ooms, J., Robinson, D., Seidel, D., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K. and Yutani, H. (2019) 'Welcome to the tidyverse', *The Journal of Open Source Software,* 4(43), pp. 1686, DOI: 10.21105/joss.01686.

Wu, D., Gao, Y., Tan, Y., Liu, Y., Wang, L., Zhou, M., Xi, X., Ma, C., Bininda-Emonds, O. R. P., Chen, T. and Shaw, C. (2018) 'Discovery of Distinctin-Like-Peptide-PH (DLP-PH) from the skin secretion of *Phyllomedusa hypochondrialis*, a prototype of a novel family of antimicrobial peptide', *Frontiers in microbiology,* 9, pp. 541, DOI: 10.3389/fmicb.2018.00541.

Wu, R., Patocka, J., Nepovimova, E., Oleksak, P., Valis, M., Wu, W. and Kuca, K. (2021) 'Marine invertebrate peptides: antimicrobial peptides', *Frontiers in microbiology,* 12, pp. 785085, DOI: 10.3389/fmicb.2021.785085.

Xiao, N., Cao, D.-S., Zhu, M.-F. and Xu, Q.-S. (2015) 'protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences', *Bioinformatics,* 31(11), pp. 1857-1859, DOI: 10.1093/bioinformatics/btv042.

Xiao, X., Wang, P., Lin, W.-Z., Jia, J.-H. and Chou, K.-C. (2013) 'iAMP-2L: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types', *Analytical Biochemistry,* 436(2), pp. 168-177, DOI: 10.1016/j.ab.2013.01.019.

Xiao, Y., Lyu, W., Yang, H., Xu, X., Zhou, C., Lu, L. and Zhang, L. (2020) 'Molecular characterization, mRNA gene expression, and antimicrobial activity of 2 new cathelicidin genes in goose', *Poultry Science,* 99(6), pp. 2983-2991, DOI: 10.1016/j.psj.2020.03.021.

Xiong, J. (2006) *Essential Bioinformatics.* Cambridge : Cambridge University Press, 2006.

Xu, J., Li, F., Leier, A., Xiang, D., Shen, H.-H., Marquez Lago, T. T., Li, J., Yu, D.-J. and Song, J. (2021) 'Comprehensive assessment of machine learning-based methods for predicting antimicrobial peptides', *Briefings in Bioinformatics*, DOI: 10.1093/bib/bbab083.

Yakovlev, I. A., Lysøe, E., Heldal, I., Steen, H., Hagen, S. B. and Clarke, J. L. (2020) 'Transcriptome profiling and in silico detection of the antimicrobial peptides of red king crab *Paralithodes camtschaticus*', *Scientific Reports,* 10(1), pp. 12679, DOI: 10.1038/s41598-020-69126-4.

Yan, J., Bhadra, P., Li, A., Sethiya, P., Qin, L., Tai, H. K., Wong, K. H. and Siu, S. W. I. (2020) 'Deep-AmPEP30: Improve short antimicrobial peptides prediction with deep learning', *Molecular therapy. Nucleic acids,* 20, pp. 882-894, DOI: 10.1016/j.omtn.2020.05.006.

Yang, S., Huang, H., Wang, F., Aweya, J. J., Zheng, Z. and Zhang, Y. (2018) 'Prediction and characterization of a novel hemocyanin-derived antimicrobial peptide from shrimp *Litopenaeus vannamei*', *Amino Acids,* 50(8), pp. 995-1005, DOI: 10.1007/s00726-018-2575-x.

Yeaman, M. R. and Yount, N. Y. (2003) 'Mechanisms of antimicrobial peptide action and resistance', *Pharmacological Reviews,* 55(1), pp. 27-55, DOI: 10.1124/pr.55.1.2.

Yin, Z., Lan, H., Tan, G., Lu, M., Vasilakos, A. V. and Liu, W. (2017) 'Computing platforms for big biological data analytics: perspectives and challenges', *Computational and structural biotechnology journal,* 15, pp. 403-411, DOI: 10.1016/j.csbj.2017.07.004.

Yoshida, M., Hinkley, T., Tsuda, S., Abul-Haija, Y. M., McBurney, R. T., Kulikov, V., Mathieson, J. S., Galiñanes Reyes, S., Castro, M. D. and Cronin, L. (2018) 'Using evolutionary algorithms and machine learning to explore sequence space for the discovery of antimicrobial peptides', *Chem,* 4(3), pp. 533-543, DOI: 10.1016/j.chempr.2018.01.005.

Yu, G., Smith, D. K., Zhu, H., Guan, Y. and Lam, T. T.-Y. (2016) 'ggtree: An R package for visualization and annotation of phylogenetic trees with their covariates and other associated data', *Methods in Ecology and Evolution*, DOI: 10.1111/2041-210X.12628.

Zare, M., Mohabatkar, H., Faramarzi, F. K., Beigi, M. M. and Behbahani, M. (2015) 'Using chou’s pseudo amino acid composition and machine learning method to predict the antiviral peptides', *The open bioinformatics journal,* 9(1), pp. 13-19, DOI: 10.2174/1875036201509010013.

Zasloff, M. (2002) 'Antimicrobial peptides of multicellular organisms', *Nature,* 415(6870), pp. 389-395, DOI: 10.1038/415389a.

Zhang, L., Chen, D., Yu, L., Wei, Y., Li, J. and Zhou, C. (2019) 'Genome-wide analysis of the ovodefensin gene family: Monophyletic origin, independent gene duplication and presence of different selection patterns', *Infection, Genetics and Evolution,* 68, pp. 265-272, DOI: 10.1016/j.meegid.2019.01.001.

Zhang, L.-J. and Gallo, R. L. (2016) 'Antimicrobial peptides', *Current Biology,* 26(1), pp. R14-9, DOI: 10.1016/j.cub.2015.11.017.

Zhang, M., Cao, M., Xiu, Y., Fu, Q., Yang, N., Su, B. and Li, C. (2021a) 'Identification of antimicrobial peptide genes in black rockfish *Sebastes schlegelii* and their responsive mechanisms to *Edwardsiella tarda* infection', *Biology,* 10(10), DOI: 10.3390/biology10101015.

Zhang, Y., Deng, P., Dai, C., Wu, M., Liu, X., Li, L., Pan, X. and Yuan, J. (2022) 'Investigation of putative antimicrobial peptides in *Carassius gibel*, revealing a practical approach to screening antimicrobials', *Fish & Shellfish Immunology*, DOI: 10.1016/j.fsi.2021.12.050.

Zhang, Y., Lin, J., Zhao, L., Zeng, X. and Liu, X. (2021b) 'A novel antibacterial peptide recognition algorithm based on BERT', *Briefings in Bioinformatics*, DOI: 10.1093/bib/bbab200.

Zhao, X., Wu, H., Lu, H., Li, G. and Huang, Q. (2013) 'LAMP: A database linking antimicrobial peptides', *Plos One,* 8(6), pp. e66557, DOI: 10.1371/journal.pone.0066557.

Zhuang, C., Huo, H., Yang, N., Fu, Q., Xue, T., Zhu, Q., Wang, B., Liu, X. and Li, C. (2021) 'Characterization of antibacterial activities and the related mechanisms of a β-defensin in turbot (*Scophthalmus maximus*)', *Aquaculture,* 541, pp. 736839, DOI: 10.1016/j.aquaculture.2021.736839.

Zolkifli, N. N., Ngah, A. and Deraman, A. (2018) 'Version control system: A review', *Procedia Computer Science,* 135, pp. 408-415, DOI: 10.1016/j.procs.2018.08.191.

Appendices

In the interest of reproducibility and transparency, methodology details, code and data necessary to reproduce the content in this thesis can be found online:

Chapter 2: ampir: an R package for fast genome-wide prediction of antimicrobial peptides

- ❖ Ampir software sourcecode and documentation:
  - ▪ https://github.com/Legana/ampir
- ❖ Ampir Shiny App sourcecode:
  - ▪ https://github.com/Legana/ampir_shiny
- ❖ Ampir methodology:
  - ▪ https://github.com/Legana/AMP_pub

Chapter 3: Benchmarking antimicrobial peptide (AMP) machine learning models in a genome-scanning context

- ▪ https://github.com/Legana/AMP_prediction_in_genomes

Chapter 4: When are machine learning AMP predictors better than homology for AMP detection in genomes?

- ▪ https://github.com/Legana/ML_vs_homology

TABLE OF CONTENTS

Supplementary information for Chapter 1: General introduction

Table S1.1: Novel AMPs found between 2018-2021 which were experimentally verified.

AMP name	Sequence data	Method	Candidates found	Additional steps	Number tested for AMP activity	Final number with AMP activity	AMP type	AMP activity	Organism	Reference and DOI
SpCrus6	Transcriptome from hemocyte, gill and hepatopancreas	BLAST to other invertebrate crustins	-	Alignment to other crustins. SignalP. Physicochemical properties	1	1	Crustin	Gram +, antiviral	Mud crab: *Scylla paramamosain*	Du et al. 2019 10.1016/j.fsi.2018.10.072
LFB	cDNA from RNA skin secretion sample	PCR amplification based on primer targeting the signal peptide region in this AMP family.	-	BLAST for structure. Physicochemical properties. 2D structures.	1	1	Brevinin-like	Gram+, gram-, antifungal	Frog: *Limnonectes fujianensi*	Li et al. 2019 10.3390/biom9060242
DRP-AC4	Translation of cDNA, cloned from cDNA library from	PCR amplification from highly conserved AMP	-	Alignment to other dermaseptins. Physicochemical properties.	1	1	Dermaseptin, α-helical	gram +, gram -	Frog: *Agalychnis callidryas*	Gong et al. 2020 10.3390/antibiotics9050243

	skin secretion	precursor primer from closely related *Phyllomedusa* species.		Secondary structures.						
Brevinin-1GHd	cDNA from RNA skin secretion sample	PCR amplification from highly conserved AMP precursor domain primer from closely related *Rana* species.	-	BLAST to all sequences in GenBank. Primary and secondary structures. Physicochemical properties.	1	1	Brevinin,α-helical	Gram +, gram -, antifungal	Frog: *Hylarana guentheri*	Jiang et al. 2020 10.1042/BSR20200019
Kassinatuerin-3	cDNA from RNA skin secretion sample.	PCR amplification designed on 5′-untranslated region of *Kassina* species.	-	Secondary structure. Physicochemical properties.	1	1	Kassinatuerin	Gram +, antifungal	Frog: *Kassina senegalensis*	Wang et al. 2020 10.3390/biology9070148
KH.C1.640, KH.C7.94, KH.S1531.4, KH.S908.1, and	Genome	Rules based screening method based on physicochemical properties and	22	SignalP. Structure. Membrane spanning regions to eliminate transmembrane proteins. Physicochemic	5	3	Linear cationic α-helical peptide (LCAMP)	Gram +, gram -, fungal (excluding KH.S921.1)	Sea squirt: *Ciona intestinalis*	Ohtsaka and Inagaki 2020 10.1038/s41598-020-69485-y

KH.S921.1		subcellular localisation		al properties range.						
Pom-1 and Pom-2	37 Peptides found via MS/MS on whole organism matched to SwissProt Mollusca proteins and APD3 AMP database with MaxQuant	Machine learning. CAMP3, AMPscanner and iAMPPpred. Results merged and prob_AMP values averaged for every sequence.	37	Two highest scoring peptides were chosen. Physicochemical properties. Structure.	2	2	α-helical	Gram + (Pom1 only), gram -	Snail: *Pomacea poeyana*	Garcia et al. 2020 10.3390/biom10111473
Cathelicidin-MH	cDNA from RNA skin secretion sample.	PCR amplification based on primer from cathelicidin domain	-	Physicochemical properties. Alignment to cathelicidins from other species. Stucture	1	1	cathelicidin	Gram -, gram +, antifungal	Frog: *Microhyla heymonsivogt*	Chai et al. 2021 10.7554/eLife.64411

Pugnin A and B	Transcriptome from skin	BLAST to AMP, SignalP and amphibian skin databases. Machine learning, CAMP SVM predictor.	375	Similarity and prob_AMP >90% Physicochemical properties. Secondary structure. Alignment and protein function	2	2	α-helical	Gram -, gram +	Frog: *Boana pugnax*	Liscano et al. 2021 10.3390/pharmaceutics13040578
14 peptides	Transcriptome	BLAST to CAMP database Machine learning (CAMP)	177	Structure. Physicochemical properties. BLAST similarity score.	22	14	α-helical	Gram -, gram +, antifungal	Mealworm: *Zophobas atratus*	Lee et al. 2021 10.1007/s10989-021-10213-z
Bthepc	Hepcidin cDNA of liver tissue	PCR amplification of Hepcidin primer from closely related species.	-	Alignment. Phylogenetic tree. Physicochemical properties. Structure	1	1	Hepcidin	gram +, gram -, antifungal	Fish: *Salmo trutta*	Huang et al. 2019 10.1016/j.fsi.2019.01.020

capitellacin	Genome	Homology to AMP gene preproalvinellacin encoding the AMP alvinellacin in related species	-	SignalP. Protein domain. Physicochemical properties. Structure.	1	1	**β**-hairpin, BRICHOS-domain	gram +, gram -	Polychaeta: *Capitella teleta*	Pantaleev et al. 2020 10.3390/md18120620
7 peptides	Transcriptome	Machine learning (CAMP) BLAST to CAMP database	248	Structure. Physicochemical properties. BLAST similarity score.	14	7	α-helical	Gram +, gram -, antifungal	Butterfly: *Papilio xuthus*	Lee et al. 2020 10.3390/insects11110776
CATH 2, CATH 3	Genome	Closely related cathelicins BLAST against goose genome	-	High similarity. Alignment. Tree.	2	2	cathelicidin	gram +, gram -	Bird: *Anser cygnoides*	Xiao et al. 2020 10.1016/j.psj.2020.03.021
Lubelisin	Metatranscriptome from cow rumen	Machine learning (APD, AMPA, BACTIBASE, CAMP)	208. 13 after spot screen	Structures. Physicochemical properties. Spot screen (fluorescence)	1	1	α-helical	Gram +, gram -	rumen eukaryotome	Onime at al. 2021 10.1186/s12866-021-02172-8

ZM-804	Transcriptome	Machine learning (CAMP) Top predicted peptide picked and checked with dbAMP, ClassAMP, iAMPpred and AntiBP.	14	Top prediction score (prob_AMP). Physicochemical properties. Structure	1	1	Cationic α-helical	Gram +, gram -	Maize: *Zea mays*	Hassan et al. 2021 10.3390/ijms22052643
HR-CATH	Skin transcriptome	BLAST skin transcriptome to get cDNA of tiger frog cathelicidin gene	-	SignalP. Physicochemical properties. Alignment to other frog cathelicidins.	1	1	cathelicidin	Gram +, gram -	Frog: *Hoplobatrachus rugulosus*	Chen et al. 2021 10.1016/j.cbpc.2021.109072
L1	Hemocyanin of *Litopenaeus vannamei*	Machine learning (AntiBP, CAMP, APD)	20	Structure. Focus on predicted peptides with α-helical structures	5	5	α-helical β-turn antimicrobial peptide	Gram +, gram -	Shrimp: *Litopenaeus vannamei*	Yang et al. 2018 10.1007/s00726-018-2575-x
DLP-PH	Shotgun cloning of skin secretion derived cDNA library	PCR amplification of AMP 5′-untranslated region primer from closely related species.	-	Physicochemical properties. Structure	1	1	α-helical. Similar to distinctin AMP	Gram +, gram -, antifungal	Frog: *Phyllomedusa hypochondrialis*	Wu et al. 2018 10.3389/fmicb.2018.00541

SmBD	Transcriptome	BLAST against β-defensins in 2 other fish	-	SignalP. Physicochemical properties. Alignment. Structure.	1	1	β-defensin	Gram +, gram -	Fish: *Scophthalmus maximus*	Zhuang et al. 2021 10.1016/j.aquaculture.2021.736839
PhciCath5	Genome	BLAST using mammalian cathelicidins as query sequences	10	SignalP. Physicochemical properties. Alignment. Full length coding sequence.	5	1	cathelicidin	Gram +, gram -, antifungal	Koala: *Phascolarctos cinereus*	Peel et al. 2021 10.1371/journal.pone.0249658
PS-029, TPS-032, TPS-035	Transcriptome	Machine learning predictors (CAMP, ADAM) BLAST against AMP databases	-	Rules based screening based on physicochemical properties and structure	15	3		Gram +, antifungal	Butterfly: *Porphyromonas gingivalis*	Lee et al. 2021 10.3390/insects12050466
13 peptides	Transcriptome	Machine learning predictors (CAMP, ADAM) BLAST against AMP databases	193	Rules based screening based on physicochemical properties. a-helix regions selected	13	13	α-helical	Gram +, gram -, antifungal	Beetle: *Psacothea hilaris*	Lee et al. 2020 10.3390/insects11100676
LFMP-001,	Mucus mass spec,	Alignment against	-	High alignment score.	2	2		Gram +	Slug: *Limac*	Hayashida & da Silva

LFMP-002	searched against Transcriptome and SwissProt	APD database		Physicochemical properties.					*us flavus*	Junior 2021. 10.1007/s00284-021-02552-3
BING	Plasma Mass spec , searched against NCBI db	Machine learning (CAMP) BLAST against CAMP.	430	Rules based using physicochemical properties.	-	1	β-sheet	Gram -, gram +	Fish: *Oryzias latipes*	Dong et al. 2021 10.1038/s41598-021-91765-4
vBD108 , vBD122	cDNA sequences from testis and epidymis RNA samples	PCR amplification using primer of canine β-defensin	-	Alignment. Physicochemical properties. SignaP.	2	2	β-defensin	Gram +, gram -	Fox: *Vulpes lagopus*	Li et al. 2021 10.3390/ani11071857
Ds-defensin	cDNA library from total RNA	PCR amplification from highly conserved defensin signal peptide region.	-	Alignment, SignalP. Physicochemical properties. Tertiary structure.	1	1	defensin	Gram +, gram -	Tick: *Dermacentor silvarum*	Li et al. 2021 10.1007/s10493-020-00584-1

BsHep	cDNA sequence from spleen RNA	PCR amplification of Hepcidin primer from closely related species.	-	Alignment. Physicochemical properties. Structure.	1	1	hepcidin	Gram +, gram -	Fish: *Bostrychus sinensis*	Shen et al. 2021 10.1016/j.aquaculture.2021.737114
Ctri9594	cDNA library from venom gland	PCR amplification using primer. Details of primer not provided.	-	SignalP. Alignment. Structure	1	1	Amphiphilic cation α-helical	gram +	Scorpion: *Chaerilus tricostatus*	He et al. 2021 10.3390/antibiotics10080896

Supplementary information for Chapter 2: ampir: an R package for fast genome-wide prediction of antimicrobial peptides

Table S2.1: Tests written for the functions within the ampir R package.

ampir functions with test files	**Tests**
df_to_faa	It writes a file It writes a file with the correct FASTA output
calculate_features	It results in a 45 column data.frame It works with multiple rows as input It returns an error when the input sequences are shorter than the min_length parameter. It returns the correct values for the physicochemical calculations.
calc_pseudo_comp	It gives correct results with default lambda parameter and with lambda_max It works with mixed length sequences Gives an error when sequence length is less than or equal to lambda_min
chunk_rows	It works when the parameter ncores is set to 1, 2 or 3.
predict_amps	It returns a data.frame with 3 columns It works when input contains: invalid or short aa sequences, sequences equal to min_len only invalid sequences, sequences contain a stop codon at the end It works with multiple cores It works when explicitly set model parameter to “precursor” or “mature” It gives an error when: sequences are not in character format, features are different to those included in ampir, the model parameter is set to NULL, the model parameter does not exist
read_faa	It results in a two column data.frame
remove_nonstandardaa	It returns a data.frame and that it removes the entire row of the sequence that contains nonstandard amino acids

Supplementary information for Chapter 3: Benchmarking antimicrobial peptide (AMP) classification models in a genome-scanning context

To explore the effect of data imbalance on performance metrics, a precision-recall curve was used. It encapsulates the trade-off between the true positive rate (recall or sensitivity) and the positive predictive value (precision) for a predictive model using various probability thresholds. The traditional precision-recall curve contains recall on the x-axis and precision on the y-axis which can be used as a guide to maximise one metric over the other. However, as the output of AMP predictors are generally probability values, i.e. how likely is it that a protein sequence is an AMP or not, the interest here was to know which probability threshold includes the most number of true AMPs. Therefore, the recall and precision of a test dataset containing both AMPs and non-AMPs over a probability threshold were calculated using a custom function written in R. The probability predictions of the ampir v0.1 model on the balanced test set of ampir v.0.1, which contained 996 AMPs and non-AMPs, were used. To match the estimated 0.01 realistic proportion of AMPs in a genome, 100 replications containing random selections of 10 AMPs and combined with all 996 non-AMPs present in the test set of ampir v.0.1 were selected. The average recall and precision metrics were calculated for these datasets over a probability threshold of 0.01 to 0.99 (see Figure S3.1A). The precision and recall metrics can also be extrapolated from balanced test sets to indicate performance on imbalance data using Equation 3.2 for an α value of 0.01, which refers to an AMP proportion of 0.01 in a genome. The precision and recall metrics for α = 0.01 (see Figure S3.1B) appear extremely similar to the average curves of 0.01 AMPs in the ampir v.0.1 test set across a range of probability values (see Figure S3.1A). This was as expected and shows the α variable is a valid depiction as the proportion of AMPs in a test set.

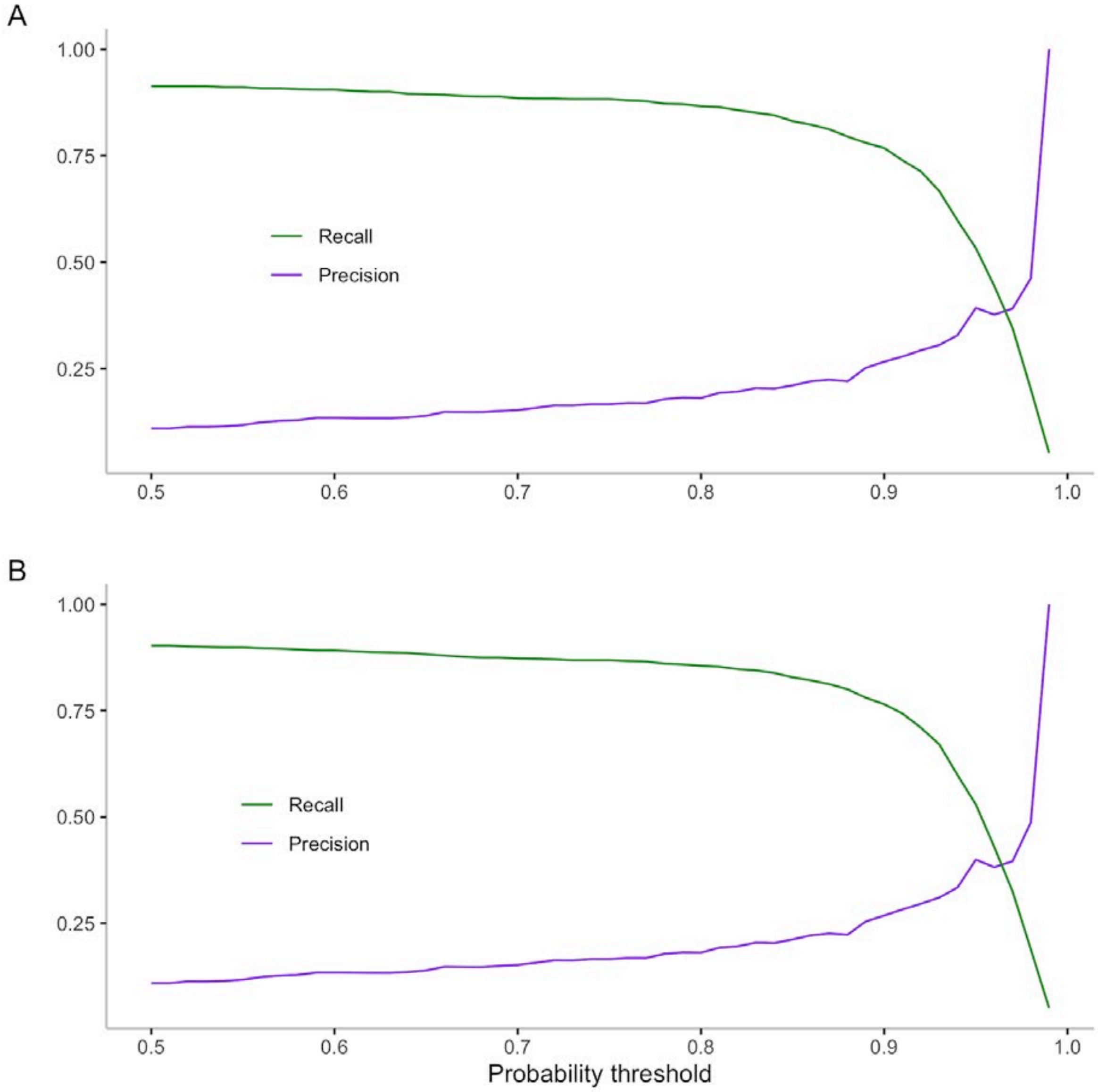


Figure S3.1: Calculated precision and recall metrics for ampir v.0.1 based on alternative methods of rescaling test results to an α value of 0.01, where α represents the proportion of AMPs in a genome. A: shows an average over 100 random subsamples of the ampir v0.1 test set in which the proportion of true cases was reduced to 0.01. B: shows results calculated using the entire ampir v0.1 test set with values of the confusion matrix rescaled to match expectations for $\alpha = 0.01$ based on equations 3.2 and 3.4.

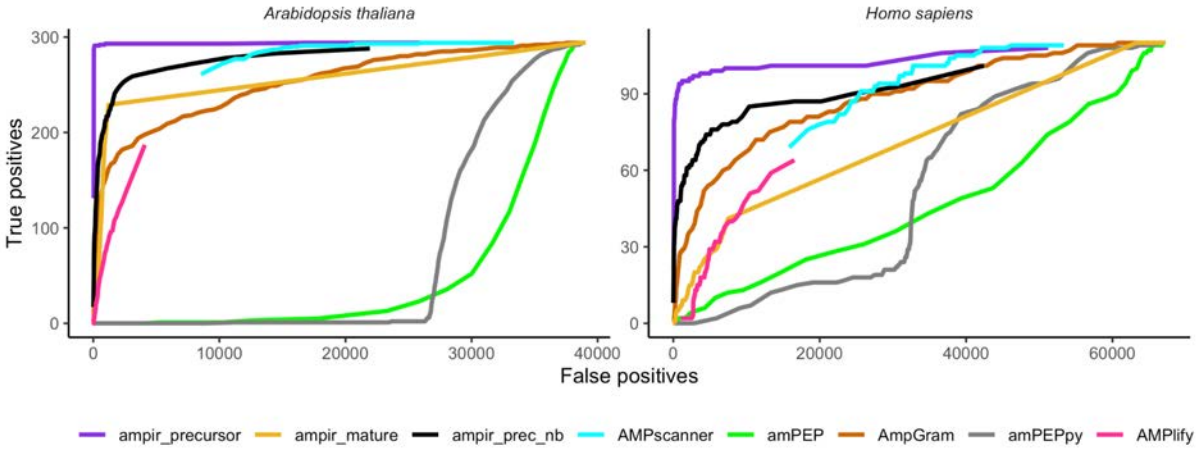


Figure S3.2: The ability of various models to predict AMPs in the proteomes of *Arabidopsis thaliana* and *Homo sapiens* using the full x-axis range. The y-axis is scaled to show the full complement of known AMPs in each genome (294 for *A. thaliana*, 112 for *H. sapiens*).

Table S3.1: Performance metrics of various AMP predictors on the proteomes of *Homo sapiens* and *Arabidopsis thaliana*

For *Homo sapiens*								
Acc	**Sp**	**Rec**	**Pr**	**F1**	**MCC**	**AUROC**	**AUPRC**	**Model**
0.97	0.97	0.86	0.05	0.09	0.20	0.94	0.30	ampir_precursor
0.05	0.05	1.00	0.00	0.00	0.01	0.74	0.00	ampir_mature
0.97	0.97	0.56	0.03	0.06	0.13	0.85	0.11	ampir_prec_nb
0.50	0.50	0.92	0.00	0.01	0.03	0.79	0.01	AMP Scanner
0.49	0.49	0.39	0.00	0.00	-0.01	0.42	0.00	amPEP
0.62	0.62	0.80	0.00	0.01	0.04	0.81	0.01	AmpGram

0.52	0.52	0.27	0.00	0.00	-0.02	0.49	0.00	amPEPpy
0.90	0.90	0.16	0.00	0.01	0.01	0.67	0.00	AMPlify
For *Arabidopsis thaliana*								
Acc	**Sp**	**Rec**	**Pr**	**F1**	**MCC**	**AUROC**	**AUPRC**	**Model**
0.99	0.99	0.99	0.38	0.54	0.60	1.00	0.83	ampir_precursor
0.01	0.01	1.00	0.01	0.02	0.01	0.97	0.15	ampir_mature
0.99	0.99	0.59	0.28	0.38	0.40	0.95	0.34	ampir_prec_nb
0.47	0.47	1.00	0.01	0.03	0.08	0.92	0.09	AMP Scanner
0.48	0.48	0.02	0.00	0.00	-0.09	0.16	0.00	amPEP
0.59	0.59	0.86	0.02	0.03	0.08	0.86	0.14	AmpGram
0.31	0.32	0.03	0.00	0.00	-0.12	0.24	0.00	amPEPpy
0.96	0.99	0.02	0.05	0.03	0.01	0.62	0.05	AMPlify

Acc: accuracy, Sp: specificity, Rec: recall, Pr: precision, F1: F1 score, MCC: Matthew’s correlation coefficient, AUROC: area under the ROC curve, AUPRC: area under the precision recall curve.

Supplementary information for Chapter 4: When are machine learning antimicrobial peptide (AMP) predictors better than homology for AMP detection in genomes?

Table S4.1: AMPs present in the AMP database which were either not annotated as an AMP in the proteomes or which were entirely absent in the proteomes, despite belonging to the same respective organisms.

Organism Name	Number of AMPs in proteome not annotated as AMP	AMPs absent in proteome
Mus musculus	1	4
Homo sapiens	1	0
Bos taurus	1	3
Oryctolagus cuniculus	0	0
Ornithorhynchus anatinus	0	0
Gallus gallus	1	0
Oncorhynchus mykiss	2	10
Drosophila melanogaster	3	0
Penaeus vannamei	0	18
Bombyx mori	0	2
Arabidopsis thaliana	2	1
Lithobates catesbeianus	1	13
Escherichia coli K-12	0	25