ML-Miner
A machine learning tool used for identification of novel Biosynthetic Gene Clusters

Paul A. Wambo

A thesis submitted in partial fulfillment of
the requirements for the
M.Sc. degree in Biology – Specialization in Bioinformatics

Department of Biology
Faculty of Science
University of Ottawa

February 2022

Abstract

Identifying biosynthetic gene clusters from genomic data is challenging, with many *in-silico* tools suffering from a high rediscovery rate due to their dependence on rule-based algorithms. Next generation sequencing has provided an abundance of genomic information, and it has been hypothesized that there are many undiscovered biosynthetic gene clusters within this dataset. Here, we aim to develop a machine learning tool, *ML-Miner*, that infers patterns that describe a biosynthetic gene cluster in an unbiased manner and, as such, enables the identification of new biosynthetic gene clusters from genomic data. To solve this challenging problem, we define a simpler one to predict the class of a known BGC. Specifically, *ML-Miner* receives as input the concatenation of sequences that are known or believed to be part of a biosynthetic gene cluster. Its task is to identify which class it belongs, *i.e.* NPRS, PKS terpene and RiPPs.

ML-Miner is a machine learning tool that uses Natural Language Processing, dimensionality reduction, and supervised learning to identify novel biosynthetic gene clusters. *BioVec* is a biological word embedding that we use to transform protein sequences from the highly curated MIBiG database of characterized biosynthetic gene clusters into their respective continuous distributed vector representations. Because the resulting protein vectors are of high dimensionality, a supervised Uniform Manifold and Approximation algorithm was employed to transform the high dimensional vectors into a robust lower-dimensional representation, as evaluated by Silhouette analysis, Hopkins' statistic, and trustworthiness analysis. The density-Based Spatial Clustering of Applications and Noise algorithm showed that the clusters identified from the low dimensional datasets mapped to biosynthetic gene cluster types, defined with high accuracy in the MIBiG database. A random forest classifier was then trained and evaluated using the low dimensional vectors. It was shown to classify each biosynthetic gene cluster from the MIBiG database with excellent performance metrics. Finally, the model's ability to generalize was evaluated using biosynthetic gene clusters from the antiSMASH dataset, an uncurated database containing uncharacterized biosynthetic gene clusters. The performance metrics were high, with a balanced accuracy of ~85%. After a hyperparameter search, the balanced accuracy rose to ~90%. This suggests that *ML-Miner* is a robust machine learning pipeline that can be used to identify novel biosynthetic gene clusters. Future development of a confidence score for classification and a workflow for processing bacterial genomes into gene clusters will significantly improve the utility of this tool.

Acknowledgements

First, I would like to thank my friends and family for their love, care, and support throughout my studies. They made my time during graduate school enjoyable.

I would also like to thank Christopher Boddy, Ph.D., for his expertise, mentorship, and most importantly, for being an excellent PI throughout my studies.

Finally, I would like to thank past and present members of the Boddy lab for the insightful discussions and, most importantly, for being a fantastic group.

Table of Contents

List of Figures and Illustrations

List of Tables

List of Abbrevations

A	Adenylation
AT	Acyltransferase
ARTS	Antibiotic Resistance Target Seeker
antiSMASH	Antibiotic & Secondary Metabolite Analysis Shell
BGC	Biosynthetic Gene Cluster
C	Condensation
CBOW	Continuous Bag of Words
DH	Dehydratase
DBSACN	Density-based Spatial Clustering of Application with Noise
HMM	Hidden Markov Models
HiTES	High Intensity Throughput of Elicitor Screening
ID3	Iterative Dichotomiser 3
KS	Ketosynthase
KR	ketoreductase
kNN	K-Nearest Neighbors
MCC	Matthews' Correlation Coefficient
MiBIG	Minimum Information about a Biosynthetic Gene cluster
NGS	Next Generation Sequencing
NRP	Non-Ribosomal Peptide
NRPS	Non-Ribosomal Peptide Synthetase
PCA	Principal Component Analysis
PKS	Polyketide Synthase
Pfam	Protein family

RiPPs	Ribosomally synthesized and post-translationally modified peptides
RODEO	Rapid Open reading frame Description and Evaluation Online
SK	Skip-gram
SMOTE	Synthetic Minority Oversampling Technique
TE	Thioesterase
t-SNE	t-distributed stochastic neighbour embedding
UMAP	Uniform Manifold Approximation and Projection

CHAPTER ONE: INTRODUCTION

1.1. Small bioactive molecules from nature

Nature holds a reservoir of small bioactive molecules known as natural products (NP) [1–3]. Because of the several evolutionary pressures to which the producing organisms are exposed, natural products can be found in all areas of life and are chemically varied[1,3–5]. Natural products play an essential role in an organism's ability to adapt to a new ecological niche. As a result, they are shaped by evolution and typically exhibit a potent ability to modulate specific biological responses[4,6,7]. Even though natural products are required for an organism to adapt to its environment, they are unnecessary in conventional laboratory circumstances, presumably due to a lack of competition[5,8–10].

Natural products are anabolic products of biosynthetic gene clusters (BGCs)[8,11]. BGCs are non-homologous genes close to each other in the genome that are co-regulated, enabling the concerted expression of a functional metabolic pathway[2,8,10–12]. The biosynthetic machinery's processing of the natural product precursors can be affected by mutations within the BGC, resulting in the production of a new chemical family or the creation of new analogues (Figure 1. 1) [5,11,13].

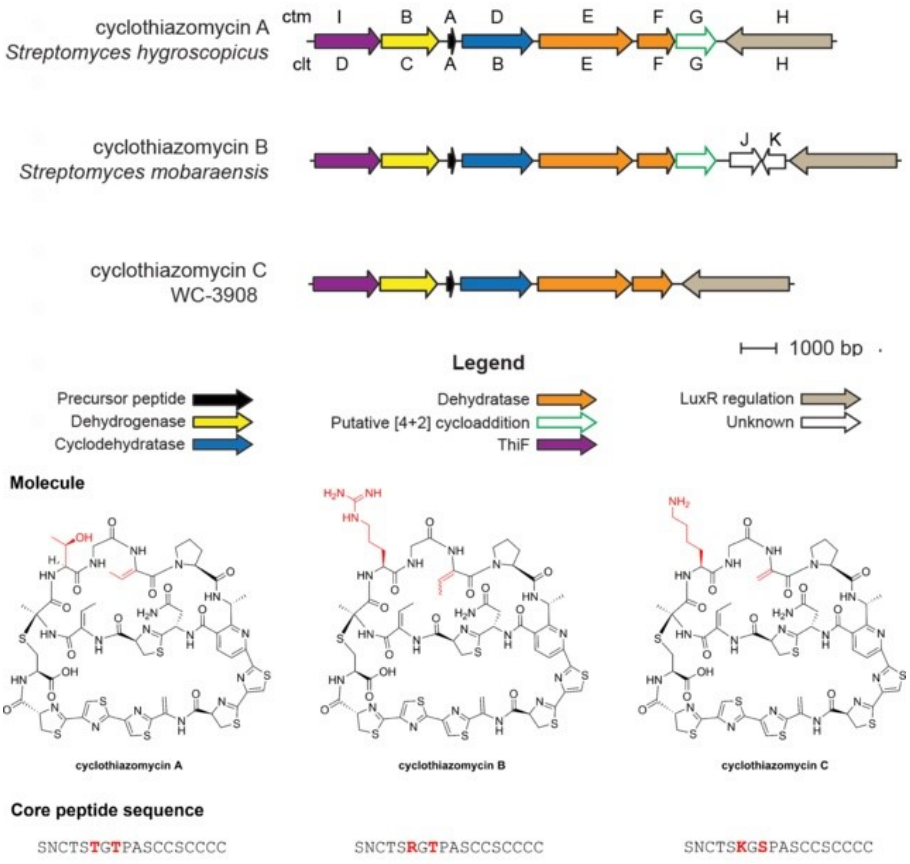


Figure 1. 1: ***Mutations within a Biosynthetic Gene Cluster lead to the production of analogs***[14]*. By mutating the core peptide of the cyclothiazomycin gene cluster, new analogues are produced by the biosynthetic machinery*[14]*, included in the thesis with permission from ACS Chem. Biol. 2014, 9, 9, 2014-2022. Copyright 2022 American Chemical Society. Further permissions related to the material excerpted should be directed to the ACS. https://pubs.acs.org/doi/abs/10.1021/cb500324n*

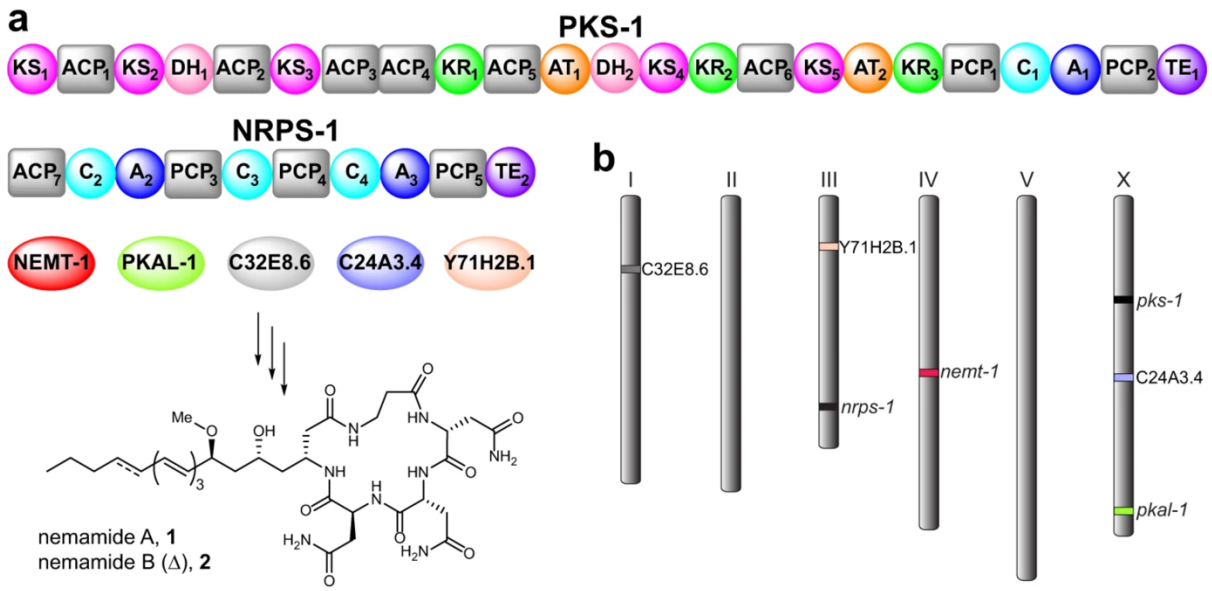


Figure 1. 2: ***Nemamide, a polyketide produced by Caenorhabditis elegans***[19]*. A) The domain organization of PKS-1 and NRPS-1 is shown, along with five additional free-standing enzymes (NEMT-1, PKAL-1, C32E8.6, C24A3.4, and Y71H2B.1) that were demonstrated in this study to be required for nemamide biosynthesis. The enzyme domains have been numbered according to the order of their appearance in PKS-1 and NRPS-1. Domain abbreviations: ACP acyl carrier protein, AT acyltransferase, KS ketosynthase, KR ketoreductase, DH dehydratase, PCP peptidyl carrier protein, A adenylation, C condensation, TE thioesterase. B) The approximate chromosomal location in* C. elegans *of pks-1, nrps-1, and the five additional genes demonstrated to be required for nemamide biosynthesis in this study*[19] *(Creative Commons). https://doi.org/10.1038/s41467-021-24682-9*

A key area of natural product research aims to develop computational and experimental tools to ease the identification of BGCs and the natural products they encode[20–22]. Computational tools focus on constructing models to understand what defines a BGC of a specific natural product type and analyze BGCs from various taxa[20]. Practical tools for natural product identification take one of two approaches; 1) a genomic approach that focuses on identifying the genomic elements of BGCs to enable natural product discovery, or 2) a chemical approach that focuses on identifying the natural product produced by the organism[21,23,24]. Integral to the first experimental approach, the more recent and practical of the two strategies is the application of computational tools.

Several computational tools have been developed to identify BGCs and characterize novel natural products[6]. These will be discussed in later sections of this introductory chapter (Section 1.5.2), as will our work on developing software that employs natural language processing, dimensional reduction, and supervised learning to identify BGCs.

1.2. A language of communication

Natural products play a critical part in an organism's ability to adapt to its ecological niche. Natural products have been tuned to mediate communication and antagonism among and across species as their BGCs have evolved, making it easier for species to adapt to their ecological niche[6].

This is exemplified by *Azteca muelleri* and *Cecropia* trees, which have a mutualistic arrangement[25]. The trees provide both shelter (parenchyma tissue) and nutritious food (Müllerian food bodies) to the ant colony[25,26]. In return for food and shelter, *A. muelleri* protects the tree from herbivores [25,26]. When an ant encounters intruders or signs of danger, it releases an acrid-smelling natural product that acts as pheromone and iridoid toxins (Figure 1. 3) [27]. Additionally, the plant communicates with ant patrollers by producing pheromones when damaged, which alerts the ants of an intruder[27,28].

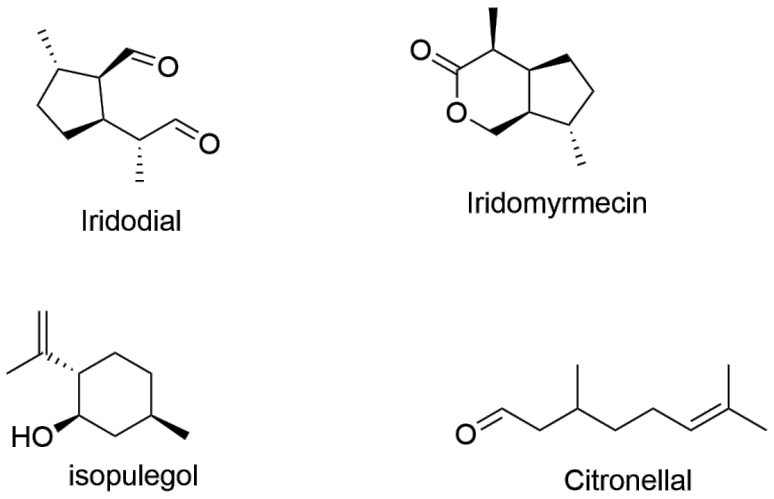


Figure 1. 3: ***Examples of compounds used for communication by ants.*** *Iridomyrmecin and iridodial are examples of toxins, while citronellal and isopulegol are alarm pheromones used to signal the presence of intruders within the* ants' *nest or domain.*

Natural products can also be used for nefarious purposes, as observed in parasitic relationships. For example, parasites can escape detection from the host's immune system by delivering anti-host factors, *e.g., type III secretion system* or toxins[29]. Mycolactones are polyketide natural products produced by the pathogenic bacteria *Mycobacteria ulcerans* (Figure 1. 4)[17,29]. Infection from *M. ulcerans* causes Buruli ulcers, characterized by persistent necrosis without an acute inflammatory response[17]. The bacteria remain undetected by the host's immune system due to the mycolactones' immunosuppressive properties[17].

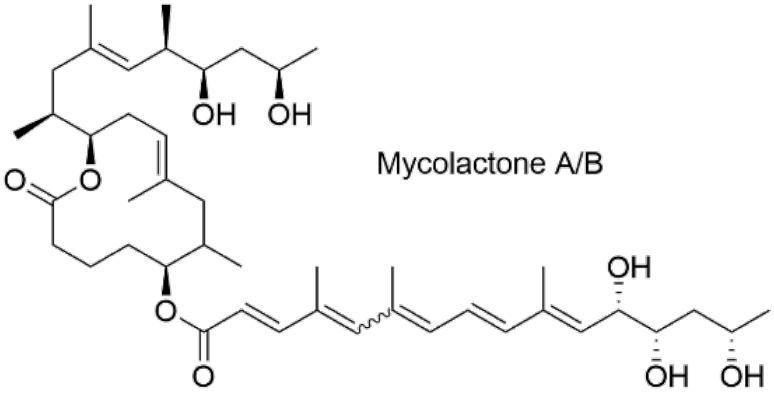


Figure 1. 4: ***Mycolactones A/B are polyketide natural products used in parasitic relations by Mycobacteria spps.*** *Mycolactones are toxins produced by mycolactone producing bacteria, such as those of the Mycobacterium genus. Mycolactones are responsible for the development of Buruli ulcers, characterized by ongoing necrosis and a lack of an immune response.*

1.3. Natural products and their analogs as drug candidates

Natural products have been developed as commercial products within the pharmaceutical and agriculture industry[1,30]. Of the 1453 new chemical entities approved by the US Food and Drug Administration between 1940 and 2013, 40% were either natural products or natural product inspired, suggesting their importance in the pharmaceutical and biotechnology industry [1,30].

Natural products have a long and storied history in drug discovery[1]. Quinine and artemisinin are natural products drugs identified through traditional medicine and are currently used to treat malaria (Figure 1. 5) [1]. The combination of both traditional knowledge and bioprospecting efforts has enabled the discovery of a vast array of antimicrobial, anti-cancer, and immuno-modulatory drugs. Here we will briefly explore the usage of NPs for medicinal purposes[1].

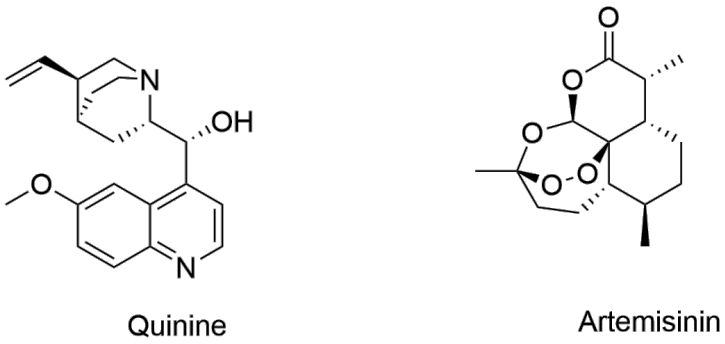


Figure 1. 5: ***Quinine and Artemisinin are NP drugs extracted from plants with roots in traditional medicine.*** *Quinine and artemisinin are both alkaloid molecules used to treat malaria. These molecules have strong ties to conventional medicine and have been used to treat malaria and fever for more than a century.*

Antimicrobial agents

Antimicrobial agents are small molecules that either kill or stop the growth of microorganisms[31]. Antibacterial and antifungal compounds are two types of chemicals that act on bacteria and fungi, respectively[31,32]. Antimicrobial agents are among the most sought-after natural product families due to their usefulness in health care and agriculture[32].

Streptomyces and *Aspergillus* genomes are some of the most prolific reservoirs of antibacterial and antifungal natural products[32,33]. Antimicrobial agents isolated from these microorganisms and their derivatives have assisted in combatting antimicrobial resistance[32]. For example, the natural product penicillin G is a potent antibiotic. Natural products can be chemically derivatized to generate analogs with superior pharmacological properties, such as the semi-synthetic antibiotic penicillin V (Figure 1. 6) [32]. Other antimicrobial natural products used in the clinic include vancomycin and erythromycin, which inhibit cell synthesis and protein synthesis in bacteria, respectively(Figure 1. 6) [32].

2.

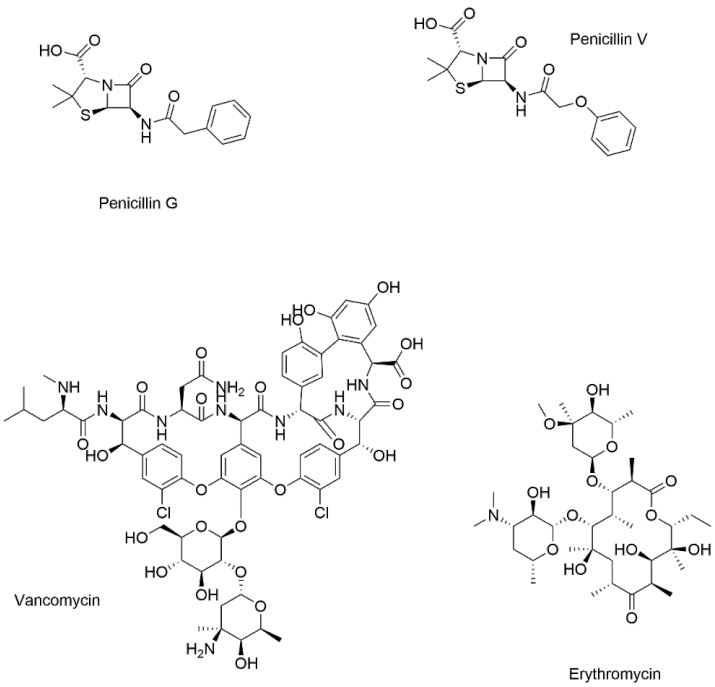


Figure 1. 6: ***Small molecules used as antibiotics.*** *Penicillin is a broad-spectrum antibiotic. Penicillin G is a natural product, and Penicillin V is a synthetic derivative. The polyketide natural product erythromycin is also a broad-spectrum antibiotic, as is the glycosylated non-ribosomal peptide vancomycin.*

Anti-tumor agents

Natural products and their derivatives have been shown to exhibit anti-tumour activity[30]. The efficacy of NPs as anti-tumour can be attributed to their ability to modulate enzymatic cascades[30]. For example, Doxorubicin and Bleomycin are NPs that inhibit DNA replication either by binding to DNA or DNA-associated enzymes (Figure 1. 7)[34,35]. Doxorubicin intercalates within DNA base pairs and inhibits the enzymes involved in DNA replication, causing the cell to initiate necrosis or apoptosis (Figure 1. 7)[35]. On the other hand, Bleomycin acts by binding to DNA and abstracting a hydrogen atom from DNA, leading to strand scission. The proteasome is also a common protein target of natural products in cancerous cells due to its ability to inhibit angiogenesis and further induce apoptosis *in-vivo*[34]. Some NP drugs that target the proteasome and are used clinically include epoxomicin and salinosporamide A (Figure 1. 7)[15,36].

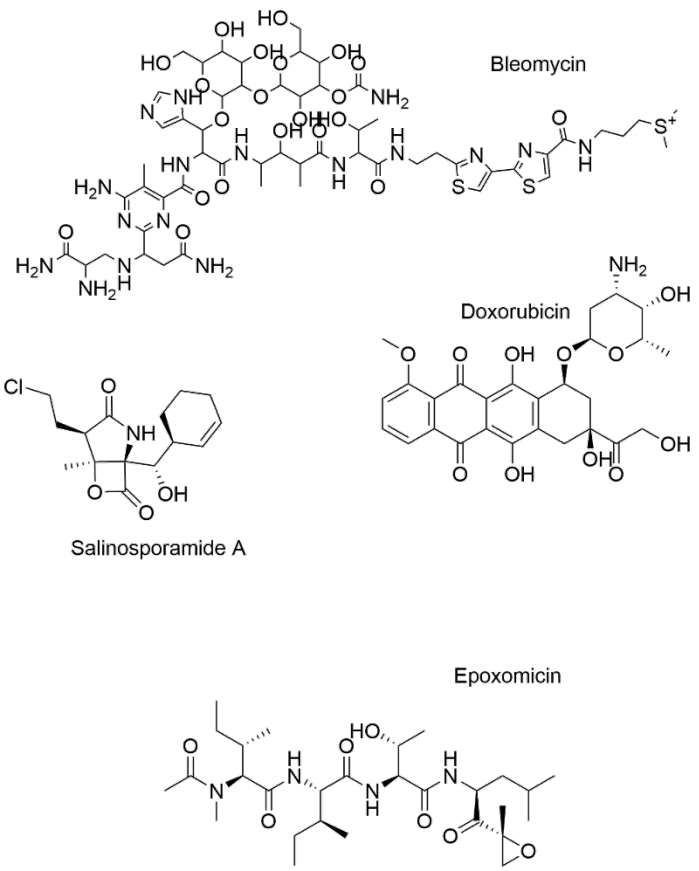


Figure 1. 7: ***Natural Products have been applied as oncological drugs due to their anti-tumorigenic activities.*** *Bleomycin and doxorubicin are anti-tumour drugs that inhibit DNA replication, forcing cells to undergo apoptosis. On the other hand, salinosporamide A and epoxomicin target the proteasome, where they impede angiogenesis and cause cells to die either due to lack of oxygen or a food source.*

1.3. Biosynthetic Gene Clusters use a variety of building blocks

Building blocks, biosynthetic chemistry, and tailoring enzymes are responsible for natural products' chemical variety[3]. Primary metabolites are redirected into secondary metabolism by certain environmental signals[3]. These secondary metabolism pathways combine primary metabolites via the action of core biosynthetic enzymes to create complex and novel structures, which can be further processed by tailoring enzymes to make the final natural products[3]. Natural products are classified based on the core enzymes responsible for their biosynthesis. Four of the most common types of BGCs are described below.

Ribosomally synthesized and post-translationally modified peptides (RiPPs) are natural products built from proteinogenic amino acids by the ribosome[3,23,37]. RiPP precursor peptides are comprised of a short core peptide region linked to a leader peptide, which regulates post-translational modification and binding affinity to the biosynthetic enzymes[37]. The leader peptide plays a regulatory role within biosynthetic machinery, ensuring all post-translational modifications occur in the correct order and keeping the maturing NP inactive and thus non-toxic to the producing organism (Figure 1. 8) [3,23,37]. A wide diversity of enzymes assist in post-translation modification of the precursor peptides during RiPP

biosynthesis[3,23,37]. RiPP diversity is attributed to the different core peptides, as other sub-classes of core peptides undergo different types of post-translational modifications [3,23,37].

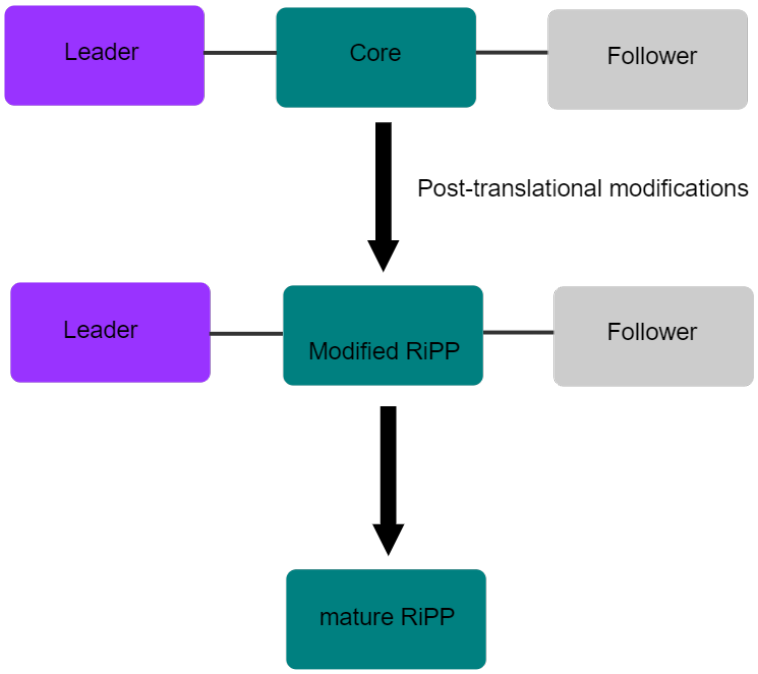


Figure 1. 8: ***Illustration of the biosynthetic machinery of ribosomally and post-translationally modified peptides****. RiPP precursor peptides are divided into a leader peptide and a core peptide. The core peptide undergoes post-translationally modifications to generate the mature RiPP that at some stage requires a leader/follower peptide for activation. In some cases, a follower peptide is present.*

Polyketides are a large family of NP, assembled using malonyl-CoA and substituted malonyl-CoA by multifunctional mega-enzymes known as polyketide synthases (PKS)[3,4,21]. PKSs are functional homologues of fatty acid synthases[4]. They are the prominent architect behind polyketide chemical diversity and are divided into three enzyme classes[3,4,21].

Type I PKS (PKS I) are modular mega-enzymes composed of an acyltransferase domain that recruits and activates a malonyl-CoA extender unit, an acyl-carrier protein (ACP) domain that anchors the elongating polyketide chain and ketosynthase domain that catalyzes the condensation of the polyketide chain with the extender unit [4]. Furthermore, the modules are arranged linearly in an assembly-line fashion, with reactions occurring non-iteratively [4].

On the other hand, type II PKSs (PKS II) are involved in the biosynthesis of aromatic polyketides in an iterative manner [4]. They consist of heterodimeric α-ketosynthase β-ketosynthase and an acyl-carrier peptide [4]. In conjunction with these enzymes, they also harbour tailoring enzymes such as cyclases, aromatases, methyltransferases, and glycosyltransferases [4]. Lastly, type III PKSs are the simplest of all. Here, a single enzyme performs all enzymatic functions. They lack an ACP domain for substrate anchoring and rely on free CoA thioesters [4]. While PKS are highly diverse, they all possess homologues of ketosynthases domains regardless of the type.

Non-ribosomal peptides (NRP) are natural products produced by multi-modular enzyme complexes known as non-ribosomal peptide synthetases (NRPS) that connect amino acids to build a peptide backbone, which is further processed by tailoring enzymes[4,38]. The minimal NRPS domain

consists of the adenylation domain, which activates the amino acid building block and anchors it to the peptidyl-carrier-protein (PCP) domain [4,38] and a condensation domain, which catalyzes peptide coupling [4,38]. The chemical diversity of NRPs can be attributed to the variety of building blocks incorporated, and the tailoring enzymes present within the biosynthetic pathways [4,38].

Finally, terpenes are one of the largest groups of natural products[39,40]. Their chemical diversity is mediated by terpene synthases, which catalyze complex carbocation-based cyclization and rearrangement cascades with linear and cyclic substrates[39,40]. The substrates used by terpene synthases are built from isopentenyl pyrophosphate and dimethylallyl pyrophosphate units [39,40].

The utilization of a variety of chemical moieties has increased the chemical diversity of NPs and their potential to modify a broad range of targets[3]. Each BGC family uses a discrete set of building blocks (**Error! Not a valid bookmark self-reference.**). In some cases, specific biosynthetic clusters have developed the ability to fuse biosynthetic types and generate molecules with several building blocks, such as hybrid polyketide-non-ribosomal peptide natural products [3].

*Table 1. 1**: **Summary of natural products and their respective building blocks.** Natural product diversity has been linked to the variety of building blocks they can incorporate into secondary metabolism.*

Natural Product	**Building block**
Polyketides	Malonyl-CoA
Non-ribosomal peptides	Proteinogenic and Non-proteinogenic amino acids
Terpenes	Isoprene
RiPPs	Proteinogenic amino acids

1.4. Evolution of Biosynthetic Gene Clusters

Several mechanisms have been proposed to explain BGC evolution[3]. Horizontal gene transfer, recombination processes, and gene duplication are among the mechanistic hypotheses[41,42]. These mechanisms help explain the difference in gene cluster organization and the existence of homologues of a particular BGC[43]. Furthermore, these mechanisms can work together in any order throughout BGC evolution [43].

The Gene Duplication Hypothesis

Gene duplication is a mechanism that generates new genetic information during molecular evolution[44]. It is usually defined as duplicating a gene, group of genes or whole genome [44]. It is hypothesized that essential genes, or those involved in primary metabolism, are repeated, and the extra copy is incorporated into secondary metabolism [44]. This phenomenon is commonly observed in antibiotic-producing BGCs, where the resistance gene is highly similar to a core gene involved in primary metabolism [44].

This is exemplified by salinosporamide A biosynthesis. Salinosporamide A is a proteasome inhibitor produced by *Salinispora tropica*, which possesses a proteasome that salinosporamide A[15,16] can inhibit. *SalI,* a subunit of the salinosporamide BGC, shares 58% sequence similarity with the β-subunit gene of the core proteasome [15,16]. While these β-subunits differ by two amino acids at the protein level, when *SalI* is combined with the α-subunit of the proteasome, a proteasome with decreased binding affinity to salinosporamide A is formed, thereby protecting the producer from the antibiotic (Figure 1. 9) [15,16].

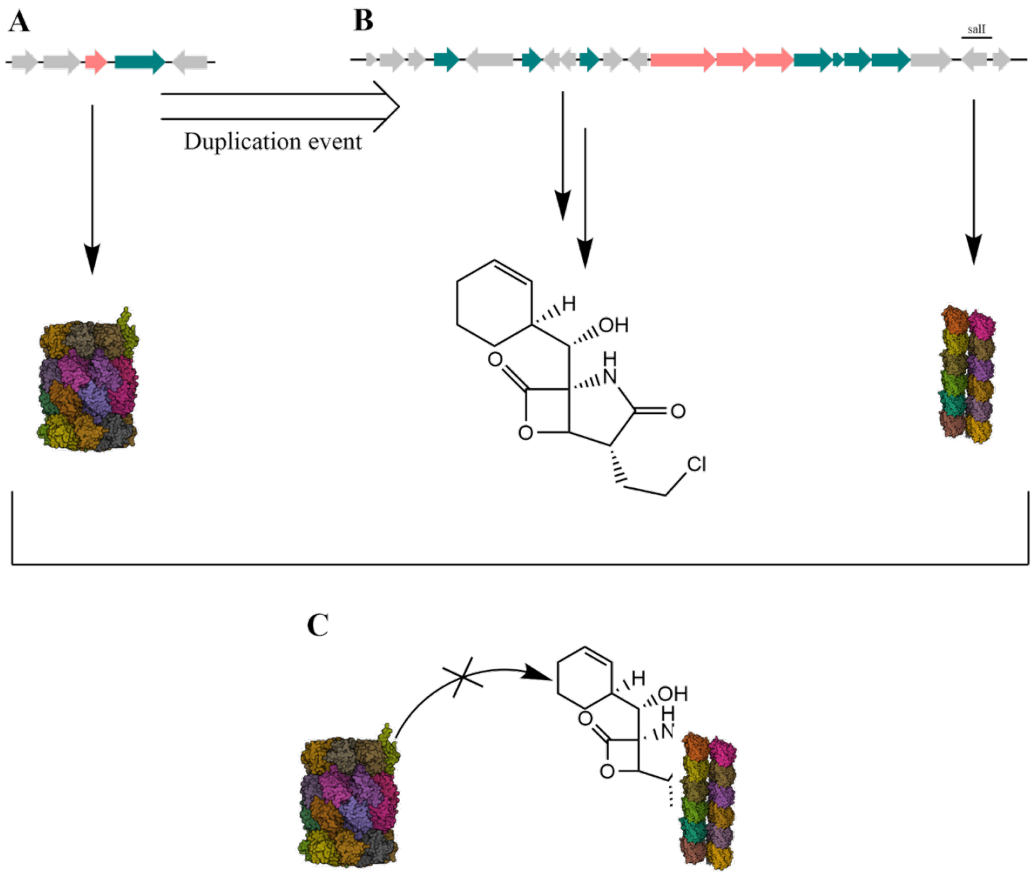


*Figure 1. 9**: Illustration of the evolution of the salinosporamide gene cluster adapted from Tran et al***[18]*. (A) The β-proteosome encoding gene underwent a duplication event. (B) The duplicated gene merged with a nascent BGC for salinosporamide formation in S. tropica, protecting the organism from the antibiotic activity of salinosporamide. (C) The SalI containing proteasome has greater resistance to salinosporamide than the core proteasome, protecting the organism from salinosporamide.*

Horizontal Gene Transfer Hypothesis

Horizontal Gene Transfer is one of the leading forces that drive genetic diversity in bacteria[7,18]. Non-vertical transmission of gene material allows organisms to expand their evolutionary repertoire and adapt to their ecological niche (Figure 1. 10)[7]. Natural products play an essential role in adaptation, making bacteria bound to accept BGCs horizontally in a new environment[7]. The production of natural products has been closely linked with environmental advantages. Antibiotics are often secreted to hinder the growth of other species that share the same ecological niche or to acquire nutrients from the environment[45].

Analysis of *Streptomyces* lineages suggests that most of the genes involved in xenobiotic metabolism are acquired laterally. Nonetheless, a small percentage is obtained by horizontal gene transfer[46]. This phenomenon has also been observed within the *Actinobacteria* and *Salinospora* genera, which are also notable secondary metabolite producers [46].

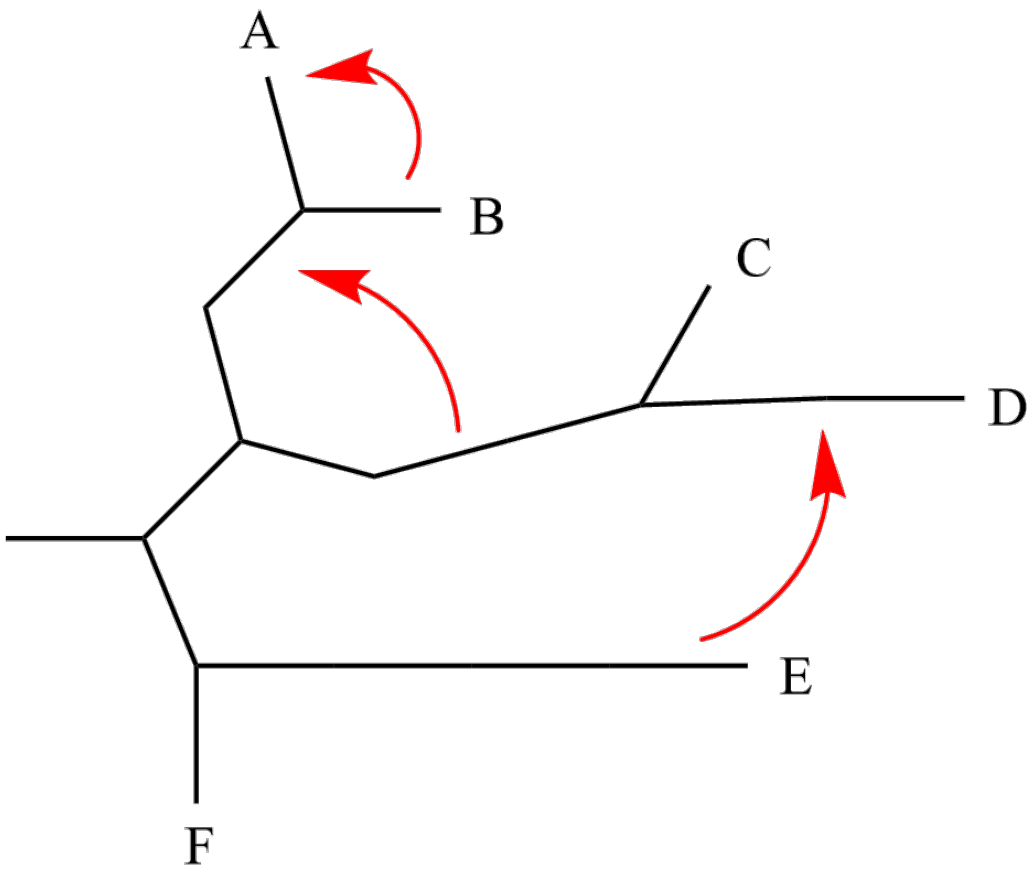


Figure 1. 10: ***Illustrating how horizontal gene transfer occurs between organisms.*** *Horizontal Gene Transfer allows organisms to gain genes from their environment to better adapt to their ecological niche. Recent horizontal gene transfer can often be detected between organisms with different phylogenetic histories.*

Recombination Events

Diversity in NRPs has been attributed to the co-evolution of adenylation and condensation domains, with the former playing a role in substrate selectivity during non-synthesis[43]. However, a simple model of genetic drift-based evolution cannot explain the plethora of substrate diversity that exists from NRP biosynthesis [43]. Recombination events have been studied as a model for NRP diversity, as non-ribosomal peptides are prime members for recombination due to their modular nature [43]. Microcystins, a class of hepatotoxins, are NRPs that exhibit this phenomenon [43]. By analyzing the structures of related NRP and their BGCs, it was proposed that gene conversion of the A domain was occurring through homologous recombination to diversify the A domains leading to NRP chemical diversity (Figure 1. 11) [43].

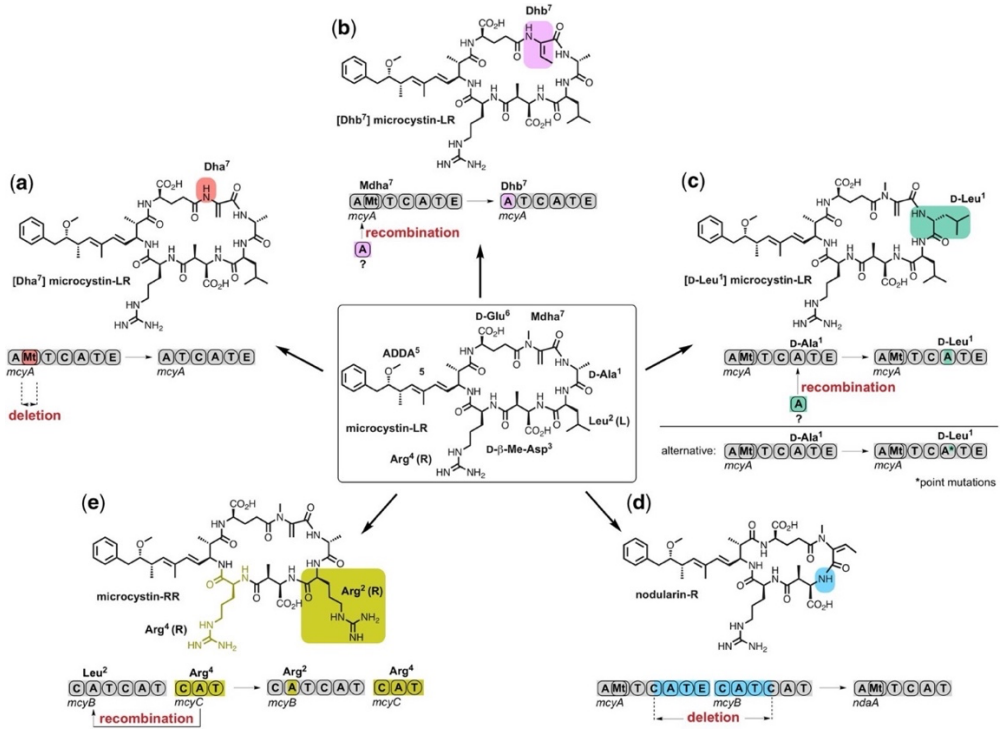


Figure 1. 11: ***Evolution of microcystin diversity by recombination, DNA deletion, and point mutations[43].*** *(a) Compared with microcystin-LR, [Dha7] microcystin-LR is produced by strains carrying a mcyA gene in which a segment encoding an N-methyltransferase was deleted[48]. (b) Recombination, in part encoding the first A domain of mcyA, likely gave rise to strains producing [Dhb7] microcystin-LR[49]. (c) Recombination of the feature encoding the second A domain of mcyA likely gave rise to strains producing [D-Leu1] microcystin-LR. Alternatively, point mutations may have led to the same chemotype[50]. (d) Deletion of two modules encoded by mcyA and mcyB likely was involved in the evolution of microcystin-like nodularins[51] (e) An intragenomic recombination event between mcyB and mcyC probably gave rise to the development of strains producing microcystin-RR[41,48,52] Gene segments encoding modules are divided into adenylation (A), condensation (C), thiolation (T), and, if present, methylation (MT) domains. (M)Dha,*

(N-methyl) dehydroalanine; Dhb, dehydrobutyrine[43] *(Creative Commons). Illustration obtained from Baunach et al.* [43]*https://doi.org/10.1093/molbev/msab015*

A critical observation from this analysis of the evolution of BGCs is that through gene duplication, horizontal gene transfer, recombination events, and genetic drift, BGCs possess a vast array of different genes and sequence diversity. This feature makes their detection via *in silico* methods highly challenging.

The intriguing question is, *why do biosynthetic genes cluster?* The parsimony hypothesis suggests that clustering genes allow the organism to transcribe and assemble the biosynthetic machinery quickly. Furthermore, as this section has shown, transferring features, genes, and groups of genes is essential in BGC evolution and function, and these processes are facilitated by clustering. Furthermore, it is proposed that clustering enables effective transfer or duplication of genes encoding proteins co-evolved to interact biochemically, a vital aspect of the function of multi-protein biosynthetic pathways. Regardless of the reason, the clustering of biosynthetic genes cluster provides critical information for designing *in silico* tools for BGC identification.

1.5. Methods used in the identification of Biosynthetic Gene Clusters

Before 2000, very few BGCs had been discovered[1]. On the other hand, natural product discovery was a well-established discipline with incredible successes in new natural product identification from the 1940s to the 1970s. This era is often referred to as the golden age of natural product discovery[1]. However, it was not until the late 1980s and early 1990s that natural product biosynthetic gene clusters were first identified[1].

Advancement in sequencing technologies and, most importantly, Next-Generation Sequencing (NGS) was a turning point in natural product research[1,20,22,42]. NGS provided an enormous wealth of sequence data and gave new insight into secondary metabolism[1,2,8,9,47]. NGS provided an abundance of sequences from a variety of environments[48,49]. Using 32,380 natural product structures, the Tanimoto coefficient, which measures similarity between two sets of elements, was calculated between chemical structures of known natural products and newly discovered ones to evaluate the novelty discovery rate[48,49]. This analysis revealed increased chemical diversity in freshly discovered natural products, suggesting the biosynthetic diversity space is vast[48,49]. Combining NGS with advances in bioinformatics, structural biology, and our updated knowledge of bacterial metabolism, *in*-silico and *in*-vivo tools were developed to further natural product research and discovery[5].

In the modern age of natural product discovery, most natural products are discovered from silent biosynthetic gene clusters[1]. Silent BGCs are not expressed under standard laboratory conditions and need specific cues from the organism's environment to be activated[10,18,20]. Determining these environmental cues is very challenging as they can be a wide variety of signals ranging from molecules within the organism's environment, cell-cell interactions, physicochemical changes, or any combination of the aforementioned[9]. Identifying or understanding the conditions to initiate transcription of these silent BGCs has been a significant roadblock in wet-lab natural product discovery[9,20].

1.5.1. Wet-lab strategies used to identify Natural Products

Because identifying new natural products is a significant challenge, many wet lab strategies have been developed to enable further compound identification. Two essential methods are briefly discussed below. In general, wet lab-based identification of natural products most frequently results in the re-isolation of known compounds. This rediscovery problem is a significant concern.

Co-culturing

Natural products are used as a device of communication between microorganisms[6,50]. It has been hypothesized that by culturing different microorganisms together, novel natural products can be produced due to microorganisms' interaction and as they compete for resources[6,50]. Consistent with this hypothesis, co-cultures of marine and terrestrial bacteria have led to identifying several new molecules with antibiotic potential[10]. This strategy provides a practical approach for eliciting the production of novel natural products and can be used to induce silent BGCs. Their products are typically produced with titers too low for experimental detection [10].

Chemical Elicitors

Many BGCs are silent under standard laboratory conditions. However, small molecules called chemical elicitors had been found in some cases to activate silent BGC expression[9,10,51]. Known chemical elicitors include natural products and synthetic drug-like molecules [9,10,51]. Recent high-throughput Elicitor Screening (HiTES) enabled the identification of elicitors for the activation of silent BGCs under specific culture conditions[9,51]. This strategy was used on *Burkholderia thailandensis* to discover 14 novel small molecules [9,51]. Thus, elicitors are a particularly promising strategy for wet lab-based new natural product discovery[9,51].

1.5.2. Bioinformatics strategies for NP identification

While the rediscovery of known natural products in the wet lab suggests natural product diversity is limited, genomic analysis has revealed that the natural product chemical space is, in fact, much larger than previously thought. However, most of these BGCs are not expressed under laboratory conditions, hence the need to develop *in-silico* tools for BGC identification. This section presents the most prominent tools used for bacterial BGC identification *in-silico*.

The antibiotic and secondary metabolite analysis shell (antiSMASH)

antiSMASH is the most used tool in identifying biosynthetic loci, as it can locate a wide range of known secondary metabolite BGCs, including those encoding RiPPs, polyketides, NRPs and terpenes[22]. Due to the conserved nature of core biosynthetic genes, profile Hidden Markov Models (pHMM) have been generated for each NP type. This pHMM is based on multiple sequence alignments of experimentally characterized signature proteins or protein domains in secondary metabolism[20,22]. antiSMASH analyzes DNA sequence data for the presence of matches to these core pHMMs. The sequence data 5' and 3' of each core pHMM is further investigated for additional biosynthetic genes, creating an annotated BGC following various rules that differ for each BGC type.

As the platform grew and became popular, a module for dereplication was introduced within the pipeline[22,52,53]. This module assisted in reducing the rediscovery rate of specific natural products and gave insight into the function of genes inferred from homology[22]. Furthermore, additional pHMMs and support vector machines were incorporated to predict stereochemistry and the substrate's specificity for some classes of biosynthetic machinery [20,54,55]. A good example showcasing the power of antiSMASH in BGC mining was its application to identify and characterize eight new RiPPs BGCs [20,54,55].

With its combination of several modules to ease analysis and detection of BGCs, its modularity to facilitate the incorporation of new capabilities, and a simple graphical user interface, antiSMASH 6.0 is a robust and one of the most broadly used tools in NP research[22,52,53,56–58].

The Antibiotic Resistant Target Seeker (ARTS)

Organisms that produce toxic molecules such as antibiotics have evolved methods to protect themselves. One of them is the presence of a resistance gene within the BGC[18,23,33,47]. These resistance mechanisms can be inactivation and export of the target molecule or modification of the target protein[33,55]. Such resistance genes can be seen as genomic signatures for the presence of antibiotics, hence aiding in genome mining of novel antibiotics[10,33,55].

ARTS is a genome mining tool that aims at identifying novel antibiotics among *Actinomycetes*[20,54,55]. It utilizes three criteria for antibiotic discovery: duplication of an essential gene, localization of the duplicate within a BGC that has been identified by antiSMASH, and evidence of horizontal gene transfer, allowing for phylogenetic reconstruction [20,54,55]. Even though these criteria do not imply resistance, *e.g.*, duplication due to adaptive biosynthesis, it has been shown that integration of multiple criteria can serve to highlight those involved in resistance [20,54,55].

Rapid ORF Description & Evaluation Online (RODEO)

RODEO is a software that focuses on ribosomal natural products (RiPPs) discovery[13]. Mining for RiPPs is a complex computational task due to the lack of a conserved protein domain that defines members of the RiPP family and the very short leader-core genes[13,22,24]. RODEO aims to rapidly assess gene neighbourhood information to identify biosynthetic genes that can aid in isolating and predicting the properties of cluster[13,24]. To achieve this, RODEO uses HMMER to build Hidden Markov Models of specific core RiPP biosynthetic genes as described in the Pfam database[13,22]. Moreover, leader-core gene coding sequence prediction is performed using a pHMM and alignment using the six open-reading

frames (ORF) that may encode potential precursor peptides. This is necessary since the leader-core sequences are too short and give low e-values [13,22]. The hypothetical ORF is then scored using a combination of heuristics, MEME motif analysis, and support vector machine classification [13,22]. Each is tailored toward a specific RiPP class and further identifies precursor peptides and peptide cleavage sites between the leader and core peptides [13,22].

RODEO was first used to discover lasso peptides BGCs, revealing over 1400 lasso peptide BGCs, with previously uncharacterized tailoring enzymes[13]. This tool has significantly impacted the discovery of RiPP BGCs, which has otherwise been more challenging than the well-defined polyketide, NRPs, and terpenoid BGC[13].

DeepBGC

DeepBGC is the first program to employ deep and supervised learning strategies to identify BGCs[42]. It combines natural language processing (NLP) to build an embedding, Pfam2vec, using sequences from the Pfam database [42,59]. Pfam2vec utilizes the skip-gram algorithm to determine the relationship between clusters [42]. It is hypothesized that the resultant vector is encapsulated within the genomic context, allowing for the contextual similarity between Pfam domains and BGCs [42,59]. Because DeepBGC uses vector space defined by Pfam2vec, an implicit assumption of this system is that BGCs are determined by the identity of their functional domains rather than sequence-specific information about those domains [42].

Not all core proteins within BGCs can be classified within a protein family as with RiPPs leader-core peptides [42]. Due to their hyper-variability and transient nature, RiPPs are not suitable for this approach[42,59]. However, polyketide synthases and non-ribosomal peptide synthases have well-conserved core proteins and domains, making such a strategy suited for their discovery [42].

Pfam2vec provided a condensed and meaningful representation of individual domains and, therefore, a functionally input to the recurrent neural network[42]. The BGC vector space described by Pfam2vec was analyzed using a Bi-directional Long Short-Term Memory (Bi-LSTM), which provides a score on whether the Pfam domain is part of BGC or not [42]. The potential class of a candidate BGC is then determined using a Random Forest [42].

DeepBGC was shown to outperform ClusterFinder, an *in-silico* tool that relies on pHMM, under three criteria, A) the ability to identify BGC positions within whole bacterial genomes, B) discriminate between BGCs and artificially created non-BGC sequences, and C) identification of novel BGC classes to which it has not been exposed to [42].

Importantly, DeepBGC was able to predict novel antibacterial BGCs that both antiSMASH and ClusterFinder failed to identify [42]. Using 3376 bacterial genomes described by RefSeq, ~227 candidate BGCs were built consisting of 5 or more Pfam domains and classified based on their compound class and molecular activity [42]. The first molecule was a lycopene-like carotenoid cluster identified as a terpene. antiSMASH identified it as a primary metabolite [42]. Interestingly, five of the putative BGCs uniquely identified by DeepBGC were from diverse taxa [42]. These results suggest DeepBGC can identify novel BGC missed by previous methods and aid in discovering novel BGCs [42].

DeepRiPP

The use of only sequence information for BGC identification can be limited, as it does not provide a complete picture of what is necessary for the identification of novel BGCs[24,60]. DeepRiPP is a strategy that combines both bioinformatics and experimental approaches to prioritize the discovery of new RIPPs molecules [24,60].

DeepRiPP consists of three modules: the NLPPrecursor module, which identifies RiPPs independent of genomic context and neighbouring biosynthetic genes. The Basic Alignment of

Ribosomal Encoded Products Locally (BARLEY) module prioritizes loci encoding novel compounds and the Computational Library for Analysis of Mass Spectra (CLAMS) module, which automates the identification of their corresponding mass spectra data from liquid chromatography-mass spectrometry analysis of complex bacterial extracts[24,60]. As previously discussed, RiPPs lack a well-defined and conserved core biosynthetic protein, making their identification a challenging task [24,60]. Combining both genomic and metabolomic data aims at solving that problem [24,60]. In addition, it also addresses the challenge of correctly linking biosynthetic loci and their products through metabolomics data [24,60].

The NLPPrecursor uses NLP to identify precursor peptides independent of genomic context and predicts cleavage patterns[60]. Limitations in genome mining associated with fragmented assemblies or the presence of distantly encoded and non-clustered modification enzymes are overcome by the NLPPrecursor, which captures a broader diversity of RiPPs [60]. Peptides resulting from this analysis are passed through RiPP-PRISM, which aids in the prediction of the complete chemical structure and tailoring enzymes involved in the enzymatic cascade [60].

BARLEY allows for retro-biosynthetic processing of known RiPP structures with local alignment to genomic information, to assign a novelty index to candidate RiPPs identified by genome mining and to dereplicate available products due to the module's ability to compare genes to genes, chemical structure to chemical structures and genes to chemical structures [60].

The CLAMS module provides an algorithm that integrates disparate sources of mass spectral information, including isotopic distributions, intensity, exact mass, fragmentation patterns, and comparative metabolomics to identify mass features consistent with the predicted RiPP biosynthetic loci[60]. This module allows for the exploitation of a broader range of metabolomics information [60].

The authors analyzed ~65,000 bacterial sequences, from which they identified ~19,000 unique unknown RiPPs, expanding the number of RiPP natural products by a factor of 6 from previous estimates [60]. Three new products from a single host were identified by linking metabolomic and proteomic data [60]. These results suggest that DeepRiPP has the potential in exploring the RiPP chemical space and identifying novel RiPPs [60]. Furthermore, this combination of experimental and bioinformatics analysis can provide a more robust strategy for analyzing the NP chemical space and identifying novel NPs [60].

1.6. Databases of Biosynthetic Gene Clusters and Natural Products

Natural products research has become an enormous field in the last 20 years[61,62]. This is because of the diverse talent it attracts, from chemists who study enzymes and ways to engineer them to computational scientists who build models to identify new biosynthetic gene clusters [61,62]. This has led to the generation of an abundance of data [61,62]. Several repositories have been developed to store various information about biosynthetic gene clusters and their associated natural products[61–66]. While all contain basic BGC information, they differ on the level of data curation and thus trustwothiness[65,66]. The two primary databases, antiSMASH and MIBiG, are discussed in this section.

1.6.1. The antibiotic and secondary metabolite analysis shell (antiSMASH) database

As discussed above, antiSMASH is the most widely used method for BGC identification. The antiSMASH database was developed to capture the annotated BGCs identified by antiSMASH from many microbial genomes, thus providing a resource to address questions such as the evolution of gene clusters[22,52,53,56–58]. The database consists of approximately 150 000 BGCs from Archeal, Bacterial and Fungal genomes. The database is heavily skewed towards bacterial biosynthetic gene clusters (*Figure 1. 12*). It captures the BGC in GenBank format, including annotating all coding sequences with matches to standard pHMM, such as Pfam domains and antiSMASH specific pHMM that predict unique biosynthetic functions. In addition, antiSMASH assigns a BGC type such as polyketide, non-ribosomal peptide, RiPP, or terpene to each BGC captured in the database [22,52,53,56–58].

1.6.2. The Minimum Information about a Biosynthetic Gene Cluster (MIBiG) database

As natural product research has grown over the past decades, developing a highly curated dataset that meets specific standards has become critical due to advancements in computational and genomic tools. The Minimum Information about a Biosynthetic Gene Cluster (MIBiG) Data Standard and Repository addresses this concern by incorporating only BGCs that have been experimentally characterized, the BGC type, complete BGC sequence, genomic loci as identified by the antiSMASH software, or experimentally characterized, natural product chemical structure, and literature references[65]. The MIBiG database consists of 1170 biosynthetic gene clusters from both bacteria and fungi that have been experimentally validated (*Figure 1. 12*)[65]. It should be noted that MIBiG is a validated subset of the antiSMASH database[57,65].

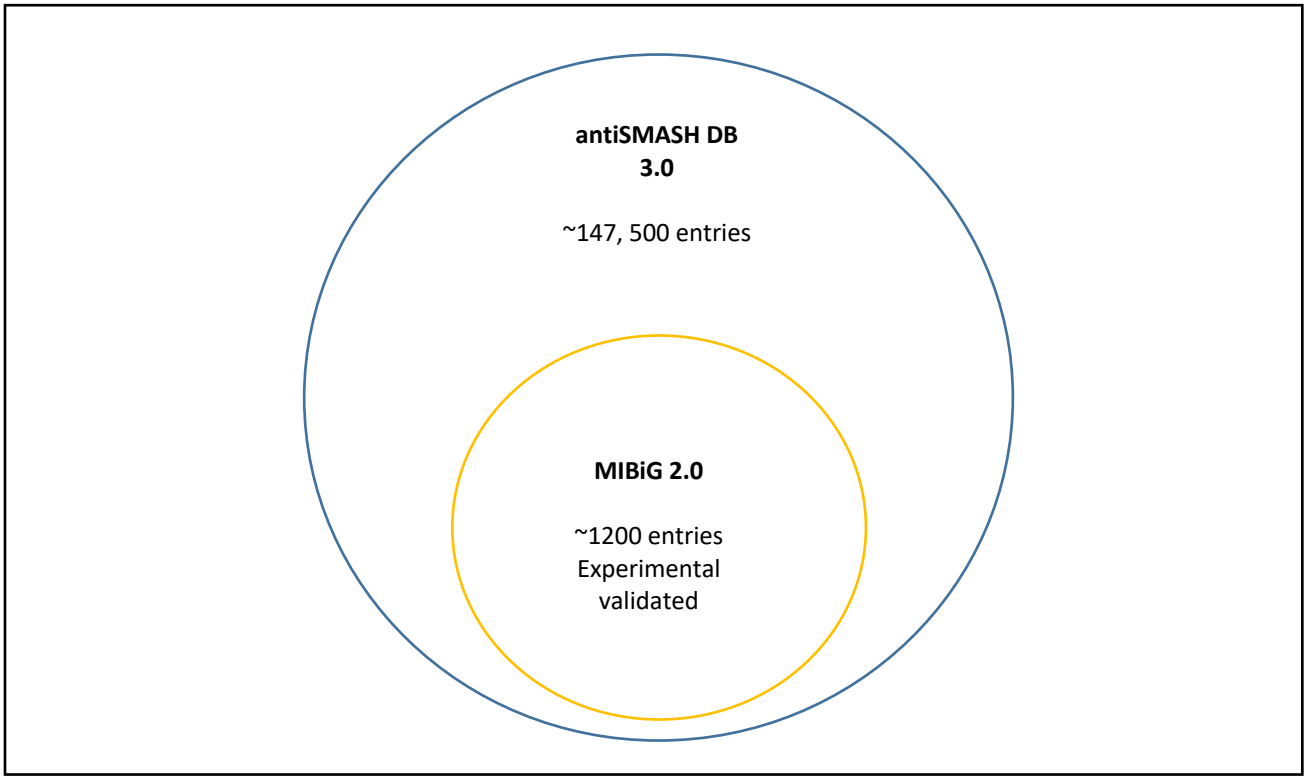


Figure 1. 12: ***Databases that have compiled information about Biosynthetic Gene Clusters.*** *The antiSMASH database is the largest repository of BGCs, with ~147,500 entries. Each entry consists of a BGC type, genomic and protein sequences and a predicted chemical structure for Polyketide Synthases. Entries in the MIBiG database are experimentally validated, and a chemical structure for the NP is associated with each entry.*

1.7. Background Information to Machine Learning

Machine learning is an umbrella term for algorithmic strategies that solve problems by extracting patterns from raw data, acquiring knowledge, and defining their rules[67]. Machine learning algorithms strive to create systems that learn from the data presented and endeavour to reduce prediction mistakes while increasing true predictions[67]. Machine learning is sub-divided into several classes based on their application, with natural language processing and data analysis being among them[67]. Deep learning focuses on developing artificial neural networks and their application to solving problems requiring massive datasets and substantial computational power. In contrast, natural language processing focuses on developing algorithms that allow computers to understand, analyze and manipulate language[67–69]. In recent years machine learning has been successfully applied to solve problems in several disciplines, such as protein structure prediction[68].

A common challenge in machine learning is building generalizable models. Generalizable models are those that, when presented with new information, can make accurate predictions [67]. Several strategies have been developed to build models that generalize adequately, such as choosing models with appropriate bias and variance, cross-validation, or searching for hyperparameters [67]. Bias refers to errors incurred from erroneous assumptions by the learning algorithm [67]. On the other hand, variance refers to errors incurred from sensitivity to small fluctuations in the training set[67]. This introduces the bias-variance tradeoff that aims to minimize the two error sources simultaneously beyond the training set during a supervised learning task[67]. Cross-validation is a resampling strategy that uses different data

portions to test and train a model on different iterations[67]. Lastly, hyperparameters are parameters whose value controls the learning process. They are external to the model, and their value cannot be estimated from the data[67].

Hidden Markov Models

A Markov Model is a statistical stochastic model that can describe the evolution of observable events that depend on internal factors, which are directly observable[70]. They rely on the Markov Property, which allows models to predict future outcomes based on their current state (Assumption). Markov Models are utilized in the following scenarios: The data consists of a sequence of observations, the observations depend probabilistically on the internal state of a dynamic system, and the actual state of the system is unknown, *i.e.* it is a hidden or latent variable.

The "Hidden" in Markov Models (HMMs) refers to the fact that we do not know the state that matches the physical event. Instead, each state matches a given output whose results are determined by the sequence of states are determined over time[68].

The number of biological sequences (DNA, RNA, and protein) has increased exponentially due to the development of Next Generation Sequencing (NGS)[70,71]. The need to analyze, identify patterns, and make sense of this data is of primary importance[70,71]. HMMs have been presented as a potential tool to address this problem due to their success in modelling the correlations between adjacent symbols, domains, or events in various engineering fields[70,71]. HMMs have successfully identified intron-exon boundaries, hence gene prediction, protein secondary structure prediction, and RNA alignment, to name a few[70,71].

BGCs are classified by types based on the building blocks, final products, and biosynthetic machinery[70,71]. HMMs have played a significant role in BGCs identification and annotation. Modelling domains using HMMs allows for their identification[70,71]. The antiSMASH software suite has applied this strategy to identify sequences that have the potential of producing secondary metabolites [70,71].

Natural Language Processing

Natural Langauge Processing (NLP) refers to computer systems that analyze, attempt to understand or produce one or more human languages [72]. NLP has been used in bioinformatics to extract information from genomic and proteomic sequence data to assess how linked they are, classify them, and predict enzyme commissions and gene ontologies[42,73,74]. The technologies enable the creation of systems that can comprehend language.

Continuous bag-of-words (CBOW) and skip-gram (SK) are two data structures for generating word embeddings [73,74]. Each model represents the semantic features of words, with word vectors positioned in the vector space so that words with similar contexts in the corpus are close together [73,74]. CBOW defines a term by its context, allowing the algorithm to learn better syntactic relationships. In contrast, SK tries to predict several context words from a single input word, allowing for stronger semantic linkages to be captured [73,74] .

Word2vec is a natural language processing (NLP) program that uses either data structures and a neural network to learn the relationships between words in a considerable text corpus[72,73]. It allows us to deduce that words such as 'cat' would retrieve morphologically similar words like its plural as closest vectors. At the same time, SK considers morphologically different but semantically relevant words such as dogs [73,74].

BioVec is an unsupervised data-driven distributed representation of biological sequences used for feature extraction and used as input for machine learning methods[74]. Feature extraction is an essential step since it aims at finding an interpretable representation of data for machine learning that

increases the performance of the learning algorithm [74]. *BioVec* was shown to capture both biophysical and biochemical properties of proteins within an n-dimensional vector [74]. *BioVec* was applied to classify protein sequences within the Pfam database and showed an average accuracy of 93% ± 0.06% with respect to the assigned label from the Pfam database 73. Furthermore, it was also used to predict unstructured regions of disordered protein sequences and determine the sequence patterns featured in disordered regions[74]. Protein sequences were obtained from both the PDB database for structured proteins and a collection of disordered proteins for FG-Nups, containing about 1,138 proteins or protein domains [74]. Using the defined representation, the algorithm had an accuracy of 99.81% with a specificity and sensitivity of 99.87% and 99.74%, respectively [74]. These results show that *BioVec* effectively encapsulates both biophysical and biochemical properties of proteins[74].

Types of machine learning algorithms

Machine learning comprises a diverse set of algorithms with varying underlying assumptions that can be used to solve a variety of issues[67]. This section examines the many forms of machine learning, the problems they solve, and the algorithms associated with them[68].

Supervised learning

Supervised learning is a type of machine learning that employs labelled information for classification or prediction pruposes[68]. Supervised learning aims to create a model that can accurately determine the label when given a set of features [68]. In classification, the model is trained to assign a label [68]. This label is part of a finite class, whether binary or multiclass[68]. On the other hand, regression models predict a target value from an unlabeled example [68].

To obtain the best performance from a classification model, they must be trained using a balanced dataset[67]. Several sample procedures have been developed to address this issue[75,76]. To produce a balanced dataset for classification, the Synthetic Minority Oversampling Technique (SMOTE) algorithm provides approaches for either oversampling or under-sampling strategies[67,76].

Decision trees are fundamental algorithms in machine learning. A decision tree is an acyclic graph used for decision making[67]. A specific feature vector *j* is examined when branching at a node [67]. If the value of the particular feature is below a threshold value, the left branch is explored. Otherwise, the right branch is explored [67]. When a leaf node is attained, the class to which the example belongs is decided or known [67].

Iterative Dichotomiser 3 (ID3) is a popular algorithm used to generate decision trees. The optimization criterion employs an average log-likelihood[67,77]. To find an optimal solution to the optimization criterion, the ID3 algorithm optimizes it approximately by constructing a nonparametric model [67,77]. Entropy is the criterion used to evaluate how good a split is [67,77]. It is a measure of uncertainty about a random variable that achieves its maximum when the random variables' values have an equal probability of occurring and a minimum when the random variable can only have one value [67,77].

The ID3 algorithm aims to minimize entropy with a split at each leaf node[67,77]. Because the algorithm does not guarantee an optimal answer, strategies like backtracking can help improve the search for an optimal decision tree at the cost of taking longer to create a model[67,77].

Random Forests are a collection of decision tree predictors. Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest[78,79]. Each tree in the random forest spits out a class prediction, and the class with the most votes becomes our model's prediction [78,79].

The key to the success of random forests, is the low correlation between the decision trees[67,78,79]. This is because uncorrelated trees produce predictions that are more accurate than any of

the individual predictions – hence protecting each other from their individual errors [67,78,79]. The prediction of an outcome x can be described as the average of B predictions:

$$y \leftarrow \hat{f}(x) \overset{def}{=} \frac{1}{B}\sum_{b=1}^{B} f_b(x)$$

A primary reason why random forests are a popular machine learning algorithm is that, by using several trees, the variance of the model is reduced, thereby reducing the chances of overfitting[67,78,79].

k-Nearest Neighbors (kNN) is a nonparametric algorithm commonly used for supervised learning. Unlike other algorithms, it does not discard the training set after the model has been trained[67]. It uses distance functions such as negative cosine similarity to evaluate the closeness of two given examples [67]. The cosine similarity is a measure of similarity between two vectors [67]. If the angle between them is zero, then the vectors are orthogonal. One would suggest that vectors are proportional and -1 the vectors are opposite[67]. The choice of distance metric depends largely on domain knowledge or can be investigated using hyperparameters [67].

$$s(x_i, x_k) = \cos\big(\angle(x_i, x_k)\big) = \frac{\sum_{j=1}^{D} x_i^j x_k^j}{\sqrt{\sum_{j=1}^{D}(x_i^j)^2}\sqrt{\sum_{j=1}^{D}(x_k^j)^2}}$$

Adaptive Boosting, or AdaBoost, is a boosting algorithm that works well when combined with decision trees[78]. Boosting describes the combination of many weak classifiers into one very accurate one by sequentially refitting the classifiers with different weighted realizations of the data set. Each model learns from previously made mistakes [78]. In the case of AdaBoost, it learns from previously made mistakes by increasing the weight of misclassified data points [78]. Therefore, it makes a new prediction by adding the weight of each tree and multiplying them by the prediction of each tree, such that the tree with the highest weight will have more power of influence when making the final decision [78]. AdaBoost is relatively robust to overfitting in low noise datasets and only has a few hyperparameters to be tuned to improve model performance [78]. However, its performance when subjected to noisy data is debated. Moreover, it is not optimized for speed [78].

Unsupervised learning

Unsupervised learning aims to extract patterns from unlabeled datasets[67]. Unsupervised learning is mainly used for exploratory data analysis by providing information on the distributions and features of the dataset[67]. Unsupervised learning aims to create a model that takes an input and produces an output that solves a problem[6768].

Clustering, density estimation, and outlier detection are unsupervised learning tasks[6768]. Clustering techniques help determine the likelihood of discovering groups in data. In contrast, density estimation algorithms help summarize the data distribution, and outlier detection tells how distinct an input is from the norm as specified in the dataset[6768].

A similarity criterion identifies similar data points within the dataset. The criterion uses distance metrics such as Euclidean, Mahalanobis, or any other unknown distance to cluster the data points[6768]. Finally, cluster evaluation is challenging because labels are not present at the start[6768]. In contrast to supervised learning, where labels are provided, the number of clusters is unknown at the outset6768. When working with unsupervised learning algorithms, this poses several issues[6768].

k-means is a centroid-based clustering algorithm that aims to partition the data into k clusters so that data points in the same cluster are similar while being different from those other clusters by using several distance metrics such as Euclidean to determine how similar data points are[67].

Density-Based Spatial Clustering of Application with Noise (DBSCAN) is a density-based clustering algorithm[67]. Unlike k-means, which asks for the number of clusters *a priori*, this algorithm finds core high-density samples and expands clusters [67]. This requires two hyperparameters, *n* and e, where the latter is the maximum distance between two points and the former is the number of points belonging to a given cluster [67].

It should be noted that k-Means and DBSCAN are hard clustering algorithms, meaning data points belong to a single cluster [67]. Moreover, density-based algorithms can build clusters of arbitrary shapes. In contrast, centroid-based clustering algorithms can only create clusters of a hyper-spheric form, limiting the types of clusters made [67].

Dimensionality reduction

Dimensionality reduction is a branch of machine learning that seeks to transform high-dimensional data into a meaningful representation of reduced dimensionality80–83. Dimensionality reduction algorithms have been applied to facilitate tasks such as clustering, visualization, and compression of high-dimensional data[83–85]. This technique is helpful in bioinformatics since numerous laboratory techniques yield high-dimensional datasets [83–85].

Dimensional reduction algorithms are either linear or non-linear[80,85]. Furthermore, they can either analyze the dataset to extract the most informative features (feature selections) or transform the dataset of high dimensions to lower dimensions (feature engineering) [80,85]. These attributes make them well suited for intermediate steps in other analyses, reducing noise, and data validation [80,85].

Principal Component Analysis (PCA) is a linear dimensional reduction approach that is both unsupervised and deterministic[80,81]. It embeds the data into a linear subspace of lower dimensionality [80,81]. The projected principal components (features) aim to maximize the explained variance within the dataset by launching the data to a set of orthogonal axes using a matrix decomposition strategy to produce a new dataset with fewer dimensions[80,81.]

The t-Distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear and nondeterministic dimensional reduction method that captures much of the local structure of the dataset while revealing global structures such as the presence of clusters[83,85]. To perform dimensionality reduction, it generates a Gaussian distribution to find neighbours, i.e., data points with similar characteristics, using a defined metric such as Euclidean distance [83,85].

The Uniform Manifold Approximation and Projection (UMAP) algorithm, like t-SNE, is a nonlinear, nondeterministic dimensions reduction approach[80,84]. However, it uses a combination of manifold learning and topological data analysis to reduce the high-dimensional data to a lower one [80,84]. UMAP creates a high-dimensional graph representation of the data before optimizing a low-dimensional graph to be structurally comparable[80,84].

The fundamental difference between UMAP and t-SNE is balancing local and global structures. UMAP preserves the global structure of the dataset better than t-SNE[80,85]. Thus, inter-cluster linkages created by UMAP are more essential than those produced from t-SNE [80,85]. Nonetheless, because both approaches distort the high-dimensional space, they are not directly interpretable in the way that a PCA is [80,85].

Evaluation of machine learning algorithms

Evaluation of machine learning algorithms is an essential step in data analysis, as it gives a measure of how well the model performs when exposed to novel data[82]. This section will discuss the various metrics used to evaluate these models.

There exist numerous challenges when implementing unsupervised models. Several factors need to be considered when evaluating clustering algorithms, such as clustering tendency, number of clusters, and quality[86].

Clustering tendency is a technique for assessing clustering algorithms[67]. The Hopkins statistic reports on the clustering tendency of a dataset. This test assumes the null hypothesis that a dataset is generated by a Poisson point process and is thus uniformly and randomly generated [87]. The statistic measures how well the null hypothesis can be refuted. A number near zero suggests that the data is strongly clustered, whereas data with a value around 0.5 is random, and data close to one is uniformly distributed [87].

$$H = \frac{\sum_{i=1}^{m} u_i^d}{\sum_{i=1}^{m} u_i^d + \sum_{i=1}^{m} w_i^d}$$

where, u_i is the distance of $y_i \in Y$ from its neighbor in X, and w_i, the distance of m number of randomly chosen x_i, with $x_i \in X$ from its nearest neighbour in X.

Under the null hypothesis, the Hopkins statistic follows a $Beta(m, m)$ distribution [87].

A challenging task in clustering problems is determining the number of optimal clusters within a dataset[86]. Finding the number of optimal clusters gives better insight into the dataset[86]. Two approaches are often used to identify the number of optimal clusters, either domain knowledge or a data-driven approach [86]. Using domain knowledge, the number of number optimal clusters can be determined a priori [86]. A data-driven approach allows for modelling the dataset to find the number of clusters [86]. The Silhouette method is a strategy used to identify the natural clusters [86]. The silhouette of a data instance measures how closely related data is to data within its cluster versus data in a neighbouring cluster [86]. A dataset with a silhouette of one suggests that the data forms a cluster [86].

$$s(i) = \begin{cases} 1 - \dfrac{a(i)}{b(i)} & if\ a(i) < b(i) \\ 0 & if\ a(i) = b(i) \\ \dfrac{b(i)}{a(i)} - 1 & if\ a(i) > b(i) \end{cases}$$

where, a(i) is a measure of how dissimilar i is to its own cluster, and b(i) implies that I is badly dissimilar to its neighbour cluster.

Several methods have been proposed to evaluate dimensional reduction algorithms[80,85]. Here we would present three methods, reconstruction error, continuity, and trustworthiness analysis[80,81,85]. Reconstruction error measures the distance between the original data point and its projection onto lower-dimensional subspace[81]. Trustworthiness measures the proportion of data points close together in the low-dimensional space[85].

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{1}^{n} \sum_{j \in U_i^k} (r(i,j) - k)$$

The trustworthiness measure is defined as where $r(i,j)$ represents the rank of the low-dimensional datapoint j according to the pairwise distances between the low-dimensional data points [85].

The variableU_i^k indicates the set of points among the *k* nearest neighbours in the low dimensional space but not in the high dimensional space [85].

Continuity on the other hand, is a defined as

$$C(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in V^k} (\hat{r}(i,j) - k)$$

where $\hat{r}(i,j)$ represents the rank of the high dimensional datapoint *j* according to the pairwise distances between the high dimensional datapoints [85]. The variable V_i^k indicates the set of points that are among the k nearest neighbors in the high dimensional space but not in the low dimensional space [85].

Evaluation of multiclass classification models involves the calculations of statistics that explain how well the models perform when exposed to new information[88]. It begins with the construction of a confusion matrix, which summarizes the algorithm's performance by reporting the number of true positives and negatives, with that false positives and negatives[88]. Using this table, several statistics can be derived, such as accuracy, balanced accuracy, Kappa statistic and Matthews' correlation coefficient.

Accuracy summarizes the performance of a classification task by dividing the total of correct predictions over the total number of predictions made by the model[88].

$$accuracy = \frac{tp + tn}{tp + fp + tn + fp}$$

where, tp: true positives, tn: true negatives, fp: false positives and fn: false negatives

Balanced accuracy is another performance metric calculated using the arithmetic mean of sensitivity and specificity and is mainly employed when dealing with unbalanced data[89]. Sensitivity is the percentage of positive cases the model can detect, while specificity is the percentage of negative cases the model detects [89].

$$balanced\ accuracy = \frac{sensitivity + specifity}{2}$$

Matthews' correlation coefficient (MCC) is a statistical test that measures the differences between actual and predicted values[90]. It considers all four categories of a confusion matrix and produces a high score if the classes are predicted well [90]. The score ranges from -1 to 1, where one indicates the predicted and actual labels are in best agreement[90]. Zero means no agreement at all, revealing the predictions are random[90].

$$MCC = \frac{(tn * tp) - (fn * fp)}{\sqrt{(tp + fp) * (tp + fn) * (tn + tp) * (tn + fn)}}$$

where, tp: true positives, tn: true negatives, fp: false positives and fn: false negatives

Kappa statistic tests for interrater reliability for categorical items, i.e., it measures the degree of agreement among two deliberately chosen observers who assess the same phenomenon[91]. It ranges from 0 to 1, with zero being an equal chance of agreement, while a score of one suggests perfect agreement [91].

$$Cohen's\ Kappa\ \frac{2 * (tp * tn - fn * fp)}{(tp + fp) * (fp + tn) * (tp + fn) * (fn + tn)}$$

where, tp: true positives, tn: true negatives, fp: false positives and fn: false negatives

1.8. ML-Miner: An unbiased strategy for mining biosynthetic gene clusters using machine learning

The next chapter describes a machine learning strategy for mining BGCs[42]. To achieve this, an NLP embedding is generated using *BioVec*. *BioVec* uses a skip-gram algorithm to produce machine learning ready vectors from biological sequences. Since the vectors produced from *BioVec* are 300 dimensions, supervised UMAP is adopted to transform the dataset to a low dimensional representation to increase classification accuracy[74,84]. The low-dimensional vectors produced by UMAP were used to train several classifiers, with a random forest being the most successful at classifying each biosynthetic gene cluster protein vector into its respective class[79].

Figure 1. 13: ***Illustration of ML-Miner pipeline. The pipeline relies on three facets of machine learning to analyze Biosynthetic Gene Clusters****. The first module incorporates Natural Language Processing to transform biological sequences into a machine learning-ready input. This is followed by dimensionality reduction to produce a low dimensional embedding, and, finally, the classification module aims to assign each BGC vector to its class.*

BGCs were downloaded from the MIBiG and antiSMASH database[56,65]in GenBank format. Because the BGCs are groups of genes, the biosynthetic proteins encoded by a single BGC were concatenated, generating a single protein sequence for each BGC. These sequences served as the input for the generation of the vector representations.

BioVec enabled the concatenated sequence to be converted into a high-dimensional vector. While BioVec has been used extensively, only the program is available on GitHub. Using up-to-date sequences from UniProt version 2021_04, a new *BioVec* embedding was trained[74,92]. The concatenated MIBiG BGC sequences were then converted into their vector representations.

Classification of the high dimensional vectors using the BGC types assigned by the MIBiG database as the labels[65] performed poorly *Table 3. 1*. The high dimensional vectors were then transformed into a robust low dimensional representation with UMAP and t-SNE[83,84]. The models were subjected to a cluster analysis using both clustering tendency and unsupervised learning[85,86].

The three-dimensional vectors were used to train a classifier, with the MIBiG BGC type being used as a label. The classifiers were evaluated using accuracy, balanced accuracy, the Kappa statistic, and MCC [67,90,91]. Finally, the complete pipeline was assessed using a subset of biosynthetic gene clusters from the antiSMASH database, not including sequences from the MIBiG databases[66].

ML-Miner thus consists of three steps: NLP, Dimensionality Reduction, and Classification. The NLP module was shown to conserve biological information stored within protein sequences, while the dimensionality reduction module retained the biological information stored within the protein vectors. Finally, the classification module was shown to demonstrate excellent classification metrics. The development of *ML-Miner* is described in the following chapters.

CHAPTER TWO: MATERIALS AND METHODS

This chapter describes the approaches used to generate and process the datasets used during the analysis in Figure 2. 1. First, the details necessary for gathering the sequences of raw BGCs and then processing the sequences for feature extraction using BioVec to produce a machine learning-ready input are described. Next, the high dimensional vectors produced by BioVec are transformed to a robust low dimensional representation using dimensionality reduction. At each step of the analysis, the modules were evaluated. BioVec was assessed using several statistical tests, the dimensionality reduction algorithms were evaluated by determining the datasets clustering tendency. Finally, the classification models were evaluated using several classification metrics. Several classifiers are used to build a robust classification scheme for low dimensional vectors of BGCs.

Source code available at https://github.com/poloarol/secondary_metabolite.git

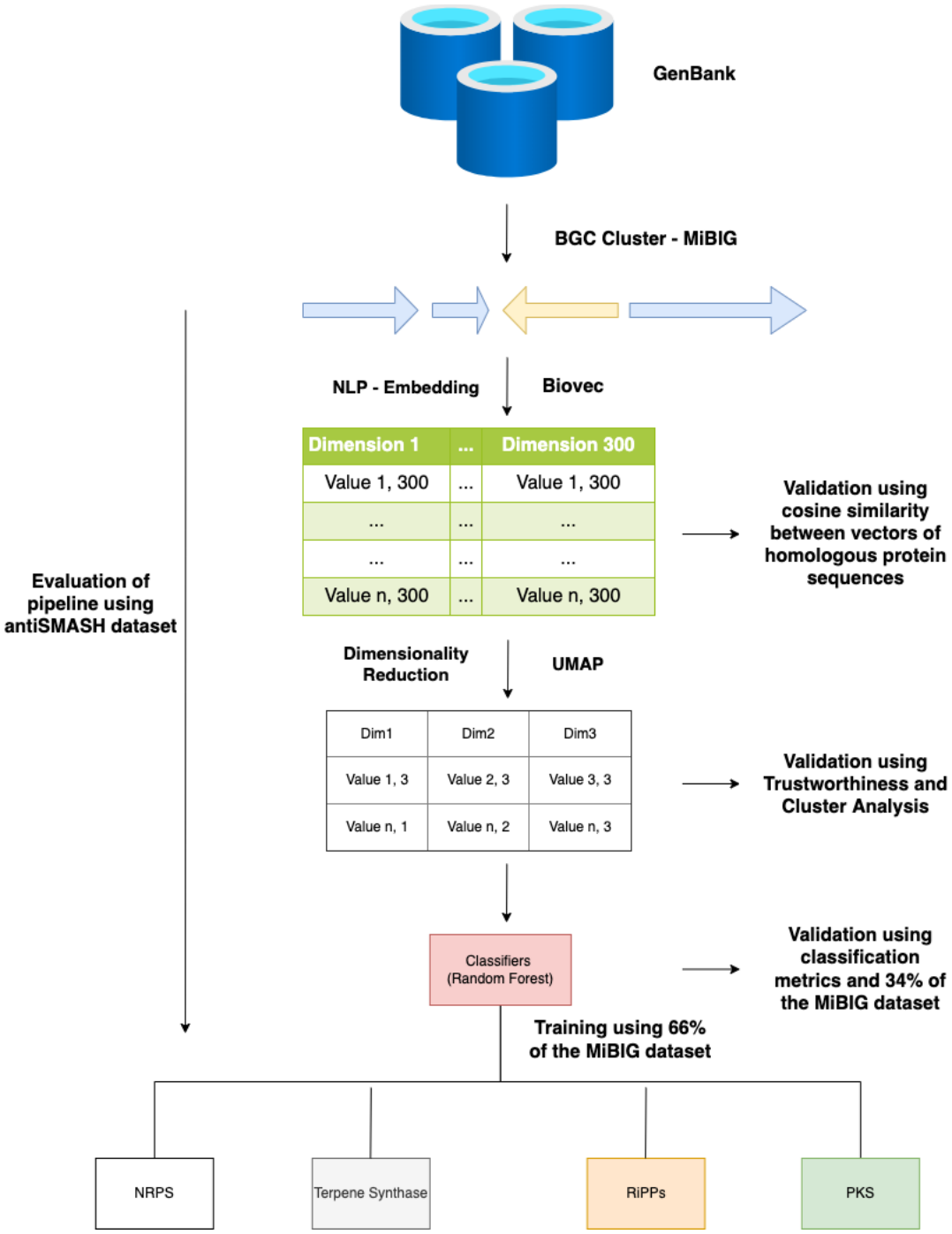


*Figure 2. 1**: Illustration of the ML-Miner pipeline. By curating the NCBI database, the MIBiG database stores Biosynthetic Gene Clusters, experimentally validated***[65,93]*. The gene clusters were then converted to their respective vectors using a BioVec model, trained with protein sequences from the UniProt database*[74,92]*. The 300-dimensional vectors obtained after transforming the sequences with BioVec were reduced to two or three dimensions*[80,82–85]*. A cluster analysis was performed on the low dimensional vectors to determine if they formed clusters. A trustworthiness analysis was used to determine how the low-dimensional vectors mapped to the high dimensional ones* [80,82–85]*. Finally, the dataset was used to train several classifiers to assess how well they could identify the various classes from the low*

dimensional vectors[67,89]. The pipeline was finally evaluated using biosynthetic gene clusters from the antiSMASH database[66].

2.1. Data Collection

Using the built-in search presented in the MIBiG and antiSMASH databases, biosynthetic gene clusters identifiers were obtained for PKSs, NRPSs, Terpenes and RiPPs (**Error! Reference source not found.**)[65,66]. The process was as follows:

1. Super-kingdom was set to Bacteria
2. The BGC-type of the desired cluster was set chosen e.g., PKS
3. Additional items were added using the EXCEPT Tag to exclude hybrids from the analysis e.g., NRPS

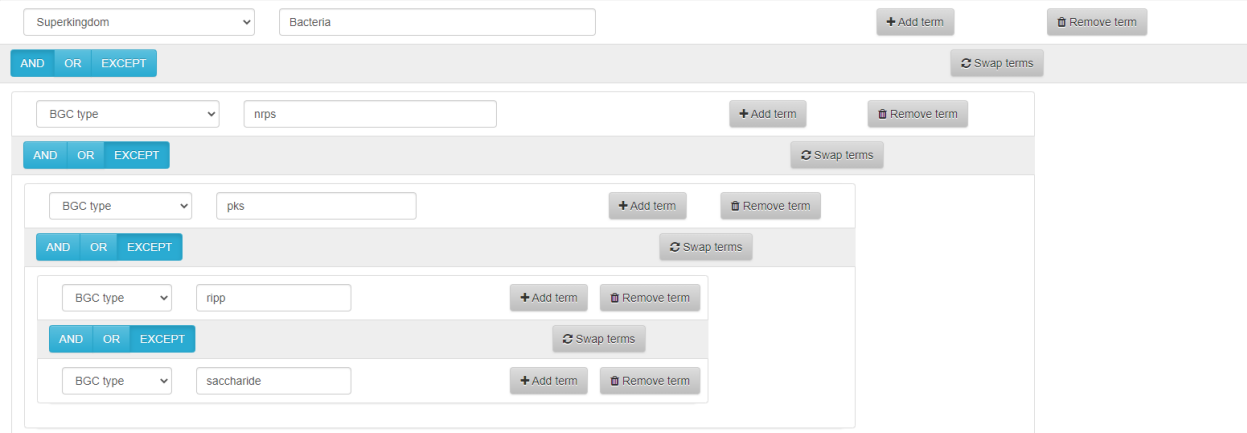


Figure 2. 2: ***Depiction of the built-in search method provided by the MIBiG website. A query was built to obtain information about all NRPS gene clusters from bacteria genomes****. An except tag was used to exclude hybrids such as PKSs, RiPPs and Saccharide.*

After the submitting the query, columns containing the Accession and Download link for the MIBiG and antiSMASH database were saved into a comma delimited file and used as input for web scrapers.

2.2. Converting protein sequences to numerical vectors, using *BioVec*

Natural language processing allows for the development of algorithms that understand languages. Here we use BioVec 0.2.7 to model the language of proteins and convert them to their respective numerical vectors[65,74,93]. The algorithm was trained using sequences from the UniProt 2021_02 release[92]. Default settings were used to train the model as described by the authors[74,92].

2.3. Evaluating *BioVec*'s ability to capture biological relevant information

Here we hypothesized that BioVec could successfully capture information pertaining to protein sequences wihtin the encoded biological vectors[74].

2.3.1. Calculating cosine similarity between eukaryotic and prokaryotic sequences

To evaluate BioVec's ability to capture relevant biological information, homologous sequences of housekeeping proteins from eukaryotes and prokaryotes were obtained from the GenBank database

(Supplementary Table 1)[74,93]. The protein sequences were converted into their respective numerical vector, and the cosine similarity score was calculated between each protein and its human homologue.

2.3.2. Using a one-way ANOVA to determine if there is a statistical difference between cosine similarity between human proteins and their homologues

The cosine similarity scores between the human protein and its homologues from prokaryotes and eukaryotes were analyzed using one-way ANOVA. The model was fitted, all assumptions were evaluated to determine whether they were not violated. Finally, a Tukey Honest Significance Difference test was performed to correct multiple comparison and control type I error rates.

2.4. Preparing BGCs for transformation using *BioVec*

Because BGCs are made of several discrete genes, the protein sequences within the GenBank files were concatenated into a single sequence and used as input for the *BioVec* embedding.

2.5. Analysis of the high dimensional BGC dataset

Dimensional reduction allows for the analysis of the high dimensional dataset by converting it to a low dimensional representation[85]. BGCs were built as described in section 4.4 and transformed to their numerical vector representation of three vectors of 100 dimensions using the trained BioVec model[74]. The obtained vector is then flattened to a row vector of 300 dimensions. Both supervised and unsupervised dimensional reduction strategies were employed for the analysis [85].
As implemented in the scikit-learn library, a robust scaler was trained using numerical vectors obtained from the MIBiG databases[65,94]. The scaled data was then used to fit the following models PCA, t-SNE and UMAP [67,83,84]. The models were then subjected to hyperparameter search, using parameters described in the appendix, to determine which parameters allowed for a robust representation of the MIBiG vector dataset.

A scree plot was used to evaluate the PCA to determine the contribution of each principal component in describing the variance within our dataset[81,85]. Both t-SNE and UMAP are non-linear dimensional reduction strategies[83–85]. A trustworthiness analysis was performed to determine how the low dimensional vectors mapped to their higher-dimensional counterparts [83–85]. A hold-out validation was performed. Ten replicates of 80% of the data were randomly chosen and analyzed to obtain a trustworthiness score. The models with sets of parameters that produced a trustworthiness score closer to 1 were considered more robust as they conserved the most information within the dataset [83–85].

2.6. Identifying the presence of clusters within the low dimensional space

Clustering algorithms aim at finding groups of data points based on their similarity[87]. To determine whether the groups observed from the dimensional reduction analysis were actual clusters, the low dimensional dataset was analyzed and evaluated using a variety of clustering algorithms.

The low dimensional vectors were used to train k-Means and DBSCAN algorithms[67]. To evaluate the presence of actual clusters within the dataset, the models were optimized to determine clustering tendency using both the Hopkins' statistic and Silhouette score[86].

2.7. Classification and evaluation of low dimensional BGC vectors

Several classifiers were trained to determine the class to which each BGC belonged. Numerical vectors representing BGCs from the MIBiG dataset were used for training, while numerical vectors from the antiSMASH database were used for testing[65,67]. The training set was subjected to an oversampling strategy to generate a balanced dataset implemented in the SMOTE library[76]. A random forest model, k-Nearest neighbour model, a multi-layer perceptron model, and a pipeline consisting of a random forest

and AdaBoost were trained [67,79,94]. The models were then evaluated using the following metrics: accuracy, balanced accuracy, Kappa statistic, and MCC[67,89–91,95]. The average and standard deviation for each metric were obtained by stratified k-fold cross-validation with three dataset splits and ten repeats.

2.8. Evaluation of the complete artificial intelligence pipeline

The entire pipeline was evaluated for its ability to classify BGC vectors it had not been exposed to using sequences from the antiSMASH database that do not appear within the MIBiG database[65,66,93]. Biosynthetic gene clusters were downloaded from the antiSMASH database, described in section 0[66]. The gene clusters GenBank files were extracted, and the protein sequences were concatenated as they appeared in the GenBank file. The resulting protein sequence was converted to its numerical vector using the BioVec model, and dimensional reduction was then performed to obtain the low dimensional vectors[74]. These vectors were passed through a classifier to determine how well our models identified the correct BGC class. Classification metrics were calculated for each model.

CHAPTER THREE: RESULTS

This chapter describes the development of a machine learning-based pipeline, *ML – Miner*, that can classify BGC type starting from GenBank files containing BGCs. The pipeline consists of *BioVec*, a dimensional reduction module, and a classifier (Figure 3. 1) [67,74,92].

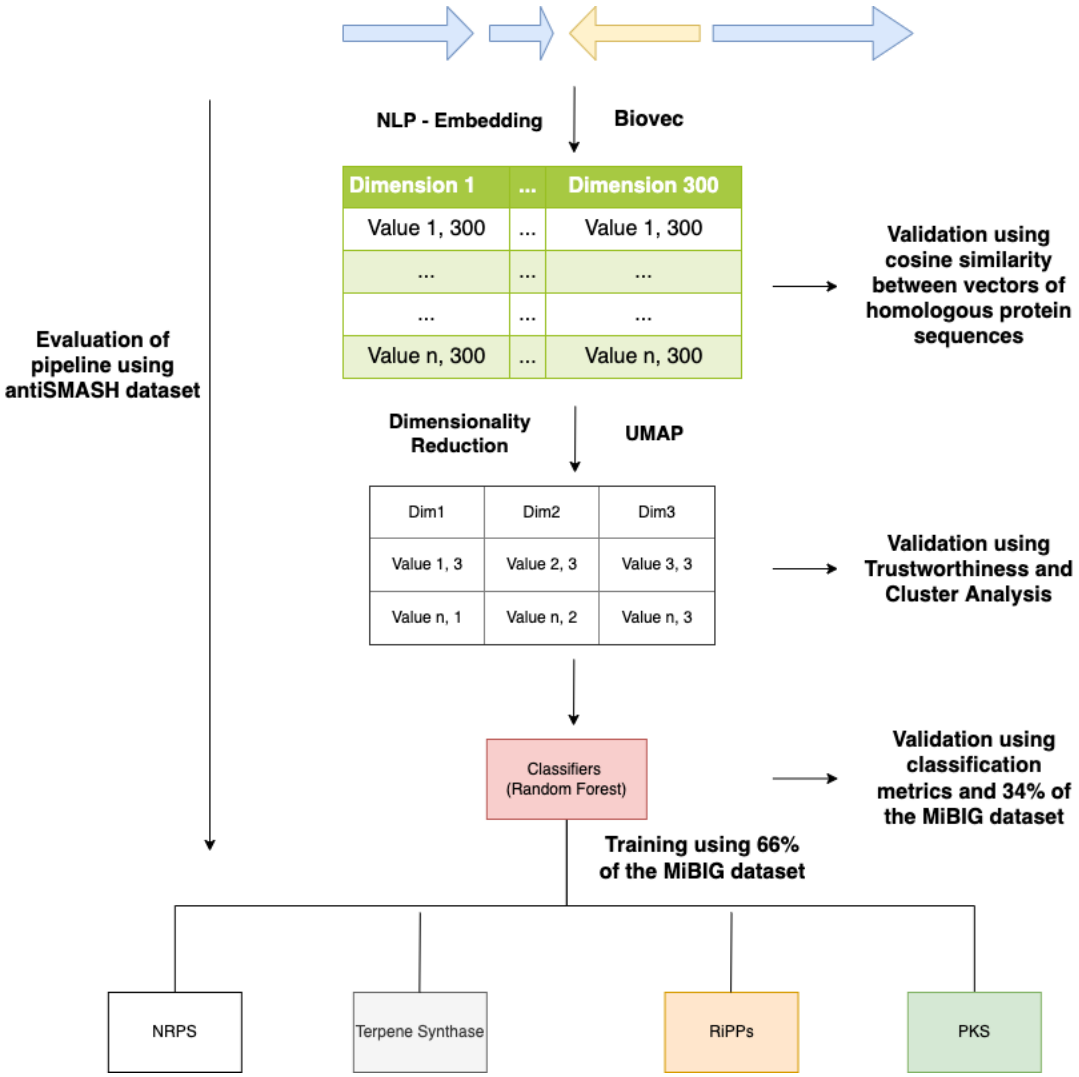


*Figure 3. 1**: **Workflow of the ML-Miner pipeline. Using BioVec, concatenated protein sequences of individual genes are transformed into continuous distributed vectors**. The resulting 300-dimensional vectors were reduced to two or three dimensions using supervised UMAP. The low dimensional vectors are used to train and evaluate a random forest.*

3.1. *BioVec* embedding captures biological information stored within protein sequences

The first module is that this pipeline converts BGCs, readily available in the MIBiG and antiSMASH databases as nucleotide or protein sequence data, into machine learning-ready vectors. BGC protein sequence data was selected as the appropriate input for analysis since it depends on organism-specific codon usage and Guanine-Cytosine content. A natural language processing-based protein embedding that transformed protein sequences into vectors was investigated.

Unfortunately, while the *BioVec* model is available on GitHub, the initially trained model is not. We retrained the model using sequences from the UniProt_2021_02 release[74,92]. We then confirmed that our retrained model captured biologically meaningful relationships. Eukaryotic and prokaryotic housekeeping proteins (Supplementary Table 1) were obtained from the NCBI databases[93] and transformed to vectors using *BioVec* to test this hypothesis73. The cosine similarity between each protein and its human homologue was calculated. We hypothesized that *BioVec* captures sufficient information within the protein sequence that the embedded vectors can reproduce known phylogenetic relationships[74]. We expected the cosine similarity of the human housekeeping proteins with other eukaryotic homologues to have a higher cosine similarity than for the human proteins and their bacterial homologues. As can be seen in Figure 3. 2 the average cosine similarity between the human and mouse and human and yeast homologues is greater than the cosine similarity scores between the human housekeeping proteins and the bacterial homologues

More quantitatively, an ANOVA revealed that BioVec captured a statistically significant difference between the distribution of cosine similarity scores of vectors for human protein sequences and different organisms (Figure 3. 1 and Figure 3. 2). A posthoc analysis was performed using a Tukey Honest Significant Difference test to determine which groups were statistically significantly different. It was observed that cosine similarity scores between human homologues and homologues from the Gram-negative bacterial Pseudomonas aeruginosa were not significantly different from those between human homologues and the Gram-negative bacteria Escherichia coli. The distribution of cosine similarity scores between human and mouse homologues was shown to be significantly different than between human and Pseudomonas aeruginosa homologues ($p <<< 0.005$). A detailed list of all significant and non-significant differences and p-values from the analysis can be found in Supplementary Table 4. These results suggest that our retrained embedding method captures biological information between protein sequences and could be used to embed BGCs.

As BGCs consist of multiple proteins and BioVec converts a single protein into a vector, we had to evaluate how the BGC would be converted into a single vector to enable downstream analysis. Two strategies were envisioned[74]. The first involved concatenating all the proteins from the GenBank file for a BGC into a single sequence and converting this into a *BioVec* encoded vector. The second involved converting each protein from a BGC separately into a vector and then summing the set of vectors. Since initial analysis showed both strategies yielded similar results, the ≈1200 BGCs with BGC type of PKS, NRPS, RiPPs, and terpenes from the MIBiG database 66 were thus processed using this first strategy to generate our dataset.

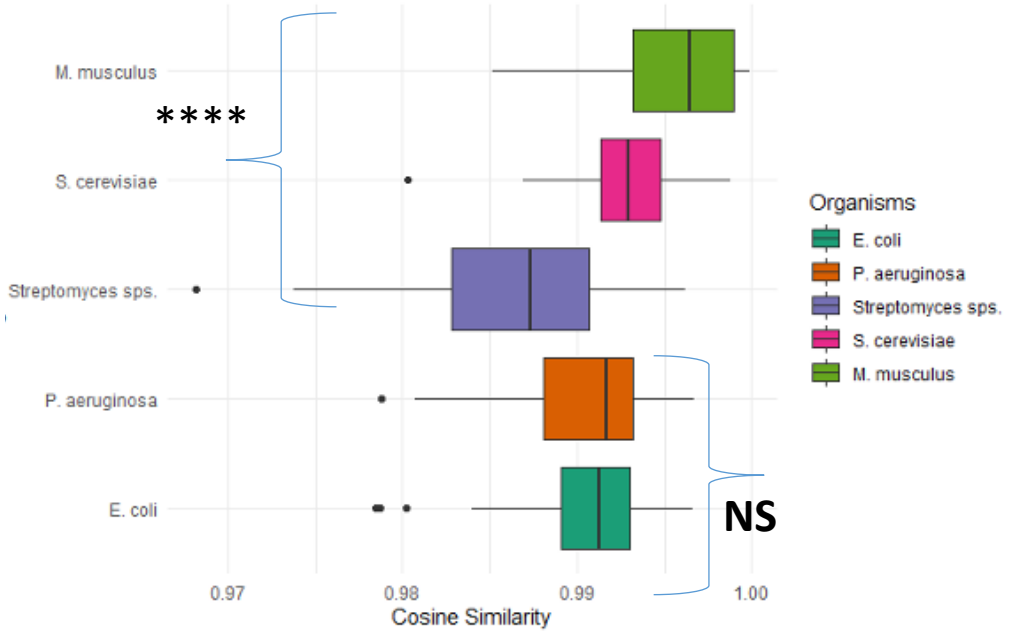


Figure 3. 2: ***Distribution of cosine similarity scores between housekeeping proteins. Cosine similarity was calculated using housekeeping proteins from humans and homologues from prokaryotes and eukaryotes****. The protein sequences were obtained from the GenBank database. A One-way ANOVA revealed a statistical difference, with a p-value <<< 0.05 and n=26.*

3.2. Classification of high-dimensional vectors obtained using *BioVec*

The analysis of high dimensional vectors can be difficult to analyze and, at times, computationally intractable[74,83,84,97]. We thus set out to evaluate the ability of machine learning algorithms to analyze our initial MIBiG dataset[65]. Several multi-classification machine learning models were fitted and assessed. While computationally tractable, these models were shown to have poor performance with balanced accuracies well under 65% and the Kappa statistic and MCC well below 0.7 (Table 3. 1). As such, the strategy of classifying the high dimensional vectors was insufficient for ML-Miner. Our requirements for moving forward with the pipeline were set at a balanced accuracy of >80% and Kappa statistic and MCC above 0.7.

The poor performance of the models used in this experiment (Table 3. 1) was hypothesized to be caused by the relatively high number of features. Random Forest, Multilayer perceptron, kNN and Gradient boosting may not have sufficient capacity to model the complex relationships between the high dimensional vectors of BGCs. Hence, they generate a model that generalizes poorly[67,74]. Models with higher capacities, such as deep learning models like Recurrent Neural Networks, which can approximate the function defined by the vector space of Biosynthetic Gene Clusters using *BioVec*[67,74] would likely perform better with these high dimensional vectors. However, insufficient data was available to train a recurrent neural network effectively with 300 features per vector and only 1,200 vectors from the MIBiG dataset. Alternatively, dimensionality reduction may produce a transformed dataset that lower capacity machine learning models can accurately model[98]. Dimensionality reduction algorithms have been shown to improve classification by denoising the dataset by handling correlated variables, perturbing the data manifold, or removing unwanted nonlinear degrees of freedom[98]. We thus chose to implement a dimensionality reduction step.

Table 3. 1: ***Classification metrics of the MIBiG database using several classifiers****. The high dimensional dataset produced by BioVec were classified into their respective classes, and classification models were evaluated. The classification metrics were calculated by performing a repeated stratified k-fold cross-validation, with three splits and ten repetitions.*

Model	Accuracy (%)	Balanced Accuracy (%)	Kappa statistic	MCC
Random Forest	67.99 ± 5.49	61.20 ± 6.53	0.649 ± 0.034	0.655 ± 0.032
AdaBoost + Random Forest	68.06 ± 5.95	60.81 ± 7.2	0.643 ± 0.0311	0.646 ± 0.031
Multi-Layer Perceptron	71.13 ± 4.35	64.08 ± 6.13	0.642 ± 0.014	0.651 ± 0.04
k-NN	65.81 ± 4.01	56.18 ± 4.82	0.556 ± 0.037	0.558 ± 0.036
Gradient Boosting	69.77 ± 4.62	61.79 ± 6.87	0.605 ± 0.048	0.611 ± 0.48

3.3. Evaluation of dimensionality reduction algorithms using cluster analysis and comparison to phylogeny

High dimensional vectors, like those generated by *BioVec*[74], can be challenging to model, as seen from above. Therefore, we implemented developed dimensionality reduction algorithms to reduce the complexity of the dataset (Figure 3. 3).

Dimensionality reduction aims to transform high-dimensional data into a robust low-dimensional representation[85]. Several dimensionality reduction algorithms were used to build a three-dimensional representation of the 300 dimensional dataset[74,85].

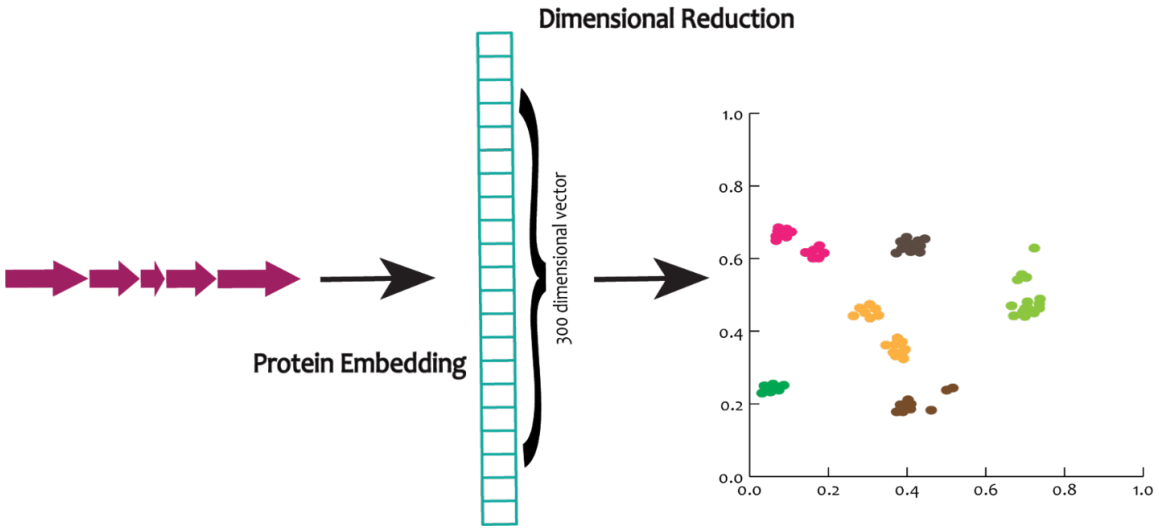


Figure 3. 3: ***Dimensionality reduction was used to create a robust low dimensional representation of the continuous distributed vectors of biosynthetic gene clusters****. The three-dimensional vectors were*

subjected to cluster analysis using DBSCAN and k-Means to examine if the clusters formed mapped to the known BGC type available from the MIBiG database.

A single value decomposition (SVD) analysis was performed to determine the number of principal components which explain most of the variance (Supplementary Figure 4)[85]. This strategy was unsuccessful as the first principal component explained more than 95% of the variance (Figure 3. 4). The root cause of this observation was not apparent, though we hypothesized that it was not that our BGC vectors were unidimensional. We thus examined alternative dimensionality reduction strategies.

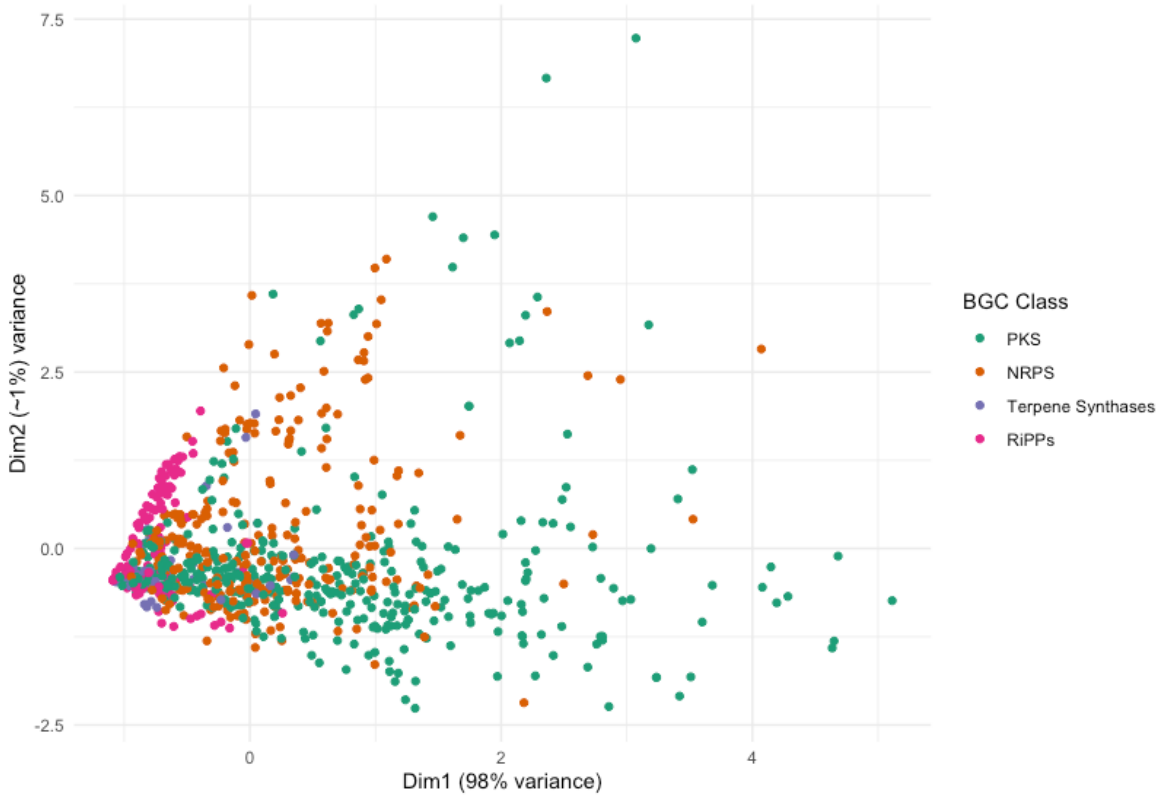


*Figure 3. 4**: Vectors were obtained by transforming BGC protein sequences obtained from the MIBiG database and BioVec**. Using the PCA model, the data was reduced to two dimensions and visualized.*

Non-linear dimensionality reduction algorithms can be practical for analyzing high dimensional dataset[80,83,84]. We employed two models, t-SNE and UMAP, as they rely on different assumptions [80,83,84]. t-SNE uses a probabilistic strategy to map the high dimensional space to a low representation. At the same time, UMAP applies manifold learning and topological structure analysis to project the high dimensional data to a lower dimension [80,83,84].

The dimensionality reduction models were evaluated using cluster analysis and the Hopkins test[85,99]. The Hopkins statistic provides a framework to determine if a dataset has a clustering tendency, i.e., it is not randomly generated[85,99]. As shown in Figure 3. 5: **Results of non-linear dimensionality reduction techniques**. Several non-linear dimensionality reduction strategies were used to transform biological vectors obtained from the MIBiG database and BioVec module. A) A t-SNE model was built with perplexity 50 and used the Manhattan distance metric, with Hopkins's statistic 0.068. B) An

unsupervised UMAP (uUMAP) model was also built with input metric Chebyshev and output metric Chebyshev, with 15 neighbours and a minimum distance of 0.4 and a Hopkins statistic of 0.073. Finally, C) a supervised UMAP (sUMAP) model was built with input metric Chebyshev and output metric Euclidean, 30 neighbours, a minimum distance of 0.1, weight of 0.3 and Hopkins's statistic 0.027.

, all datasets were shown to have a clustering tendency, with the Hopkins statistics approaching zero[99]. However, visually it can be determined that t-SNE and unsupervised UMAP did not form optimal clusters as the former makes a single large cluster that does not distinguish between each BGC type while the latter forms multiple clusters but cannot quantitatively distinguish BGC types. Supervised UMAP (sUMAP), however, is shown to build clusters representative of BGC type (Figure 3. 5)[84,89].

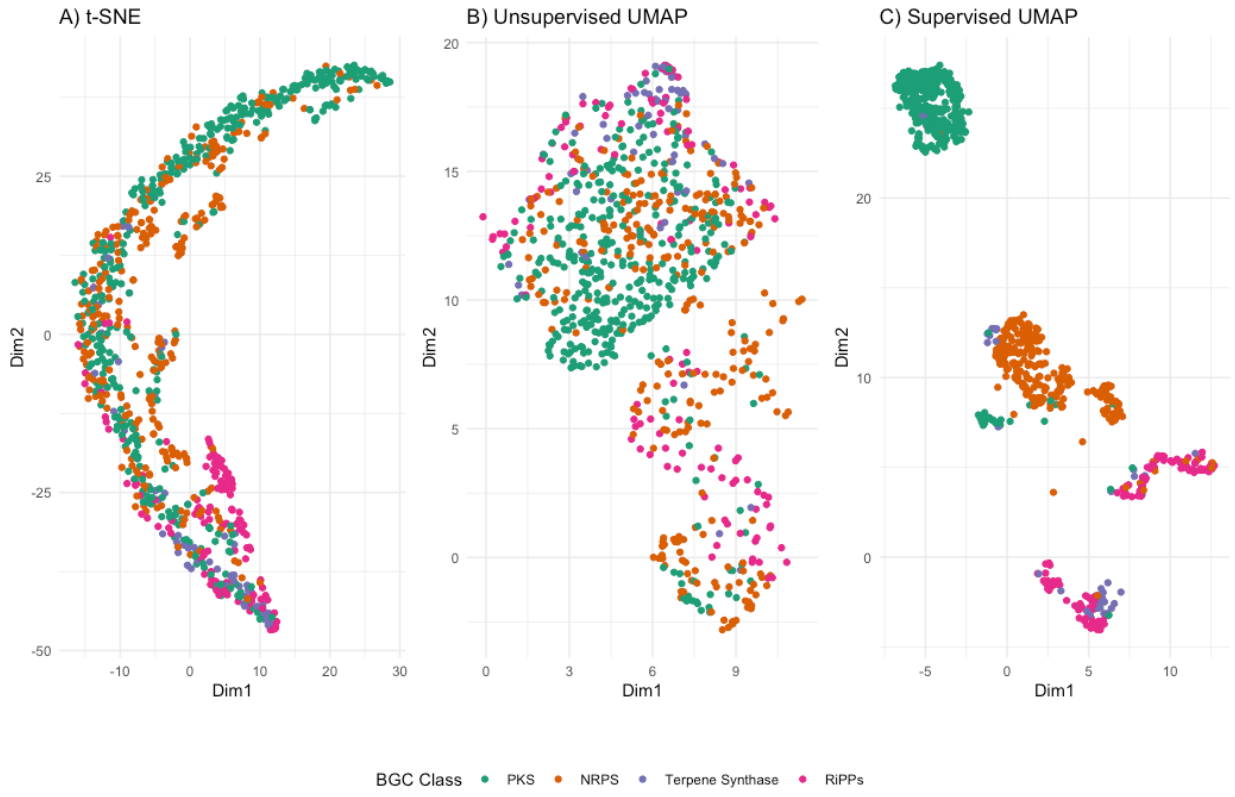


Figure 3. 5: ***Results of non-linear dimensionality reduction techniques****. Several non-linear dimensionality reduction strategies were used to transform biological vectors obtained from the MIBiG database and BioVec module. A) A t-SNE model was built with perplexity 50 and used the Manhattan distance metric, with Hopkins's statistic 0.068. B) An unsupervised UMAP (uUMAP) model was also built with input metric Chebyshev and output metric Chebyshev, with 15 neighbours and a minimum distance of 0.4 and a Hopkins statistic of 0.073. Finally, C) a supervised UMAP (sUMAP) model was built with input metric Chebyshev and output metric Euclidean, 30 neighbours, a minimum distance of 0.1, weight of 0.3 and Hopkins's statistic 0.027.*

Table 3. 2: ***Statistics used to evaluate non-linear dimensionality reduction algorithms. The resulting vectors obtained from BioVec were reduced to two dimensions****. The three-dimensional embedding was assessed using the Hopkins statistic and the Silhouette coefficient.*

Model	Hopkins' statistic	Silhouette (k-Means)	Silhouette (DBSCAN)
t-SNE	0.066	0.577	N/A
uUMAP	0.144	0.7	0.3
sUMAP	0.032	0.756	0.63

A cluster analysis was performed on each dataset produced by the dimensionality reduction algorithms. Using the Silhouette coefficient, parameters that allowed for the formation of optimal clusters were identified[86]. It was determined that t-SNE does not form clusters based on the Silhouette score calculated from k-means and DBSCAN. On the other hand, both UMAP models were shown to form clusters, with sUMAP shown to form the optimal clusters as the Silhouette score was equally high for both clustering algorithms (Figure 3. 6) [67,84,86]. The cluster analysis suggests that sUMAP formed effective clusters, which appeared to cluster based on individual BGC types, and thus indicates that classification algorithms may be practical on the lower dimensional vectors[4,67,84,86,100].

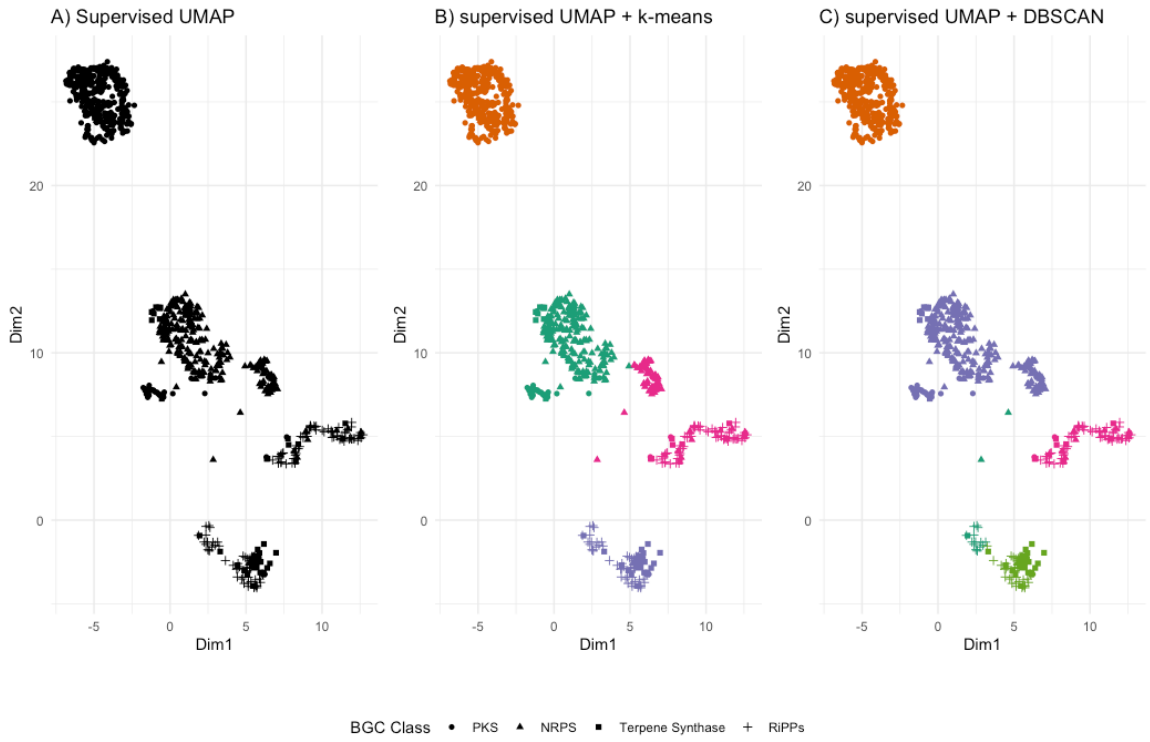


Figure 3. 6: ***Visualization of clustering analysis after dimensionality reduction****. A) A supervised UMAP model with three components and the following parameters (neighbours: 30, input metric: Chebyshev, output metric: Euclidean, minimum distance: 0.1 and weight: 0.3) was used to create a low dimensional dataset B) Using k-means, the number of optimal clusters to be four C) DBSCAN with parameters (eps: 1, minimum clusters: 19 and metric: Euclidean) was also employed to determine the number of optimal clusters. To determine if the supervised UMAP formed actual clusters, k-Means and DBSCAN were used for cluster analysis and evaluated using a Silhouette coefficient to optimal parameters.*

Using k-means and DBSCAN, the number of optimal clusters from the dimensionality reduction of the MIBiG dataset (Figure 3. 6) was determined[4,86,100]. It was observed that both k-Means and

DBSCAN formed clusters[86]. However, DBSCAN formed clusters that captured the BGC type better, more accurately concerning the MIBiG assigned labels[65,67]. This can mainly be observed by examining the clustering of the BGCs from terpene and RiPP systems (Figure 3. 6)[67,86]. These data suggest that sUMAP is a reasonable dimensional reduction strategy, generating low dimensional vectors that cluster based on BGC type.

3.4. A random forest classifier can effectively assign BGC type to the low dimensional vectors

To generate datasets for training and evaluating the classifier, the parameters of the sUMAP dimensionality reduction algorithm were varied[84,97]. The emphasis on the label weight was considered highly important for generating a dataset that could be effectively classified[84]. It was hypothesized that the less emphasis the model places on the label, the higher the priority would be to identify class-specific patterns when performing dimensionality reduction, allowing for the generation of more accurate vectors[84]. In addition to the label weight, hyperparameters, including n_neighbors, which controls how UMAP balances local and global structure in the data, and minimum distance, which controls how tightly UMAP packs points together, were varied as these are known to have a significant impact on model generation[84]. Finally, various distance metrics were used as input and output distances[84]. Thus several UMAP models were fitted with varying parameters using the MIBiG dataset.[65,84] A subset of these representing the best models are shown in Figure 3. 7.

Because the Dimensionality reduction was supervised, the accuracy of the label associated with each vector is paramount. While multiple databases contain BGCs, only the MIBiG database contains BGCs that have been experimentally characterized. Thus, this database has the highest fidelity between the BGC and BGC types, which is essential for generating the most representative low dimensional vectors from the sUMAP models. In addition, the high accuracy of BGC type assignment from the MIBiG database will be imperative in the downstream evaluation of the performance of the classifier[65].

The models were evaluated with a trustworthiness analysis[84,89]. Trustworthiness ranges from 0 to 1 and indicates how well the low dimensional embedding keeps the global structure of the high dimensional dataset and how much information is kept[85]. A UMAP model with high trustworthiness conserves substantial information in the three-dimensional dataset as described within the 300 dimensional dataset[84,85]. All four of the models from Figure 3. 7 showed trustworthiness above our benchmark of 0.8 and thus were taken on to develop the last module, the classifier.

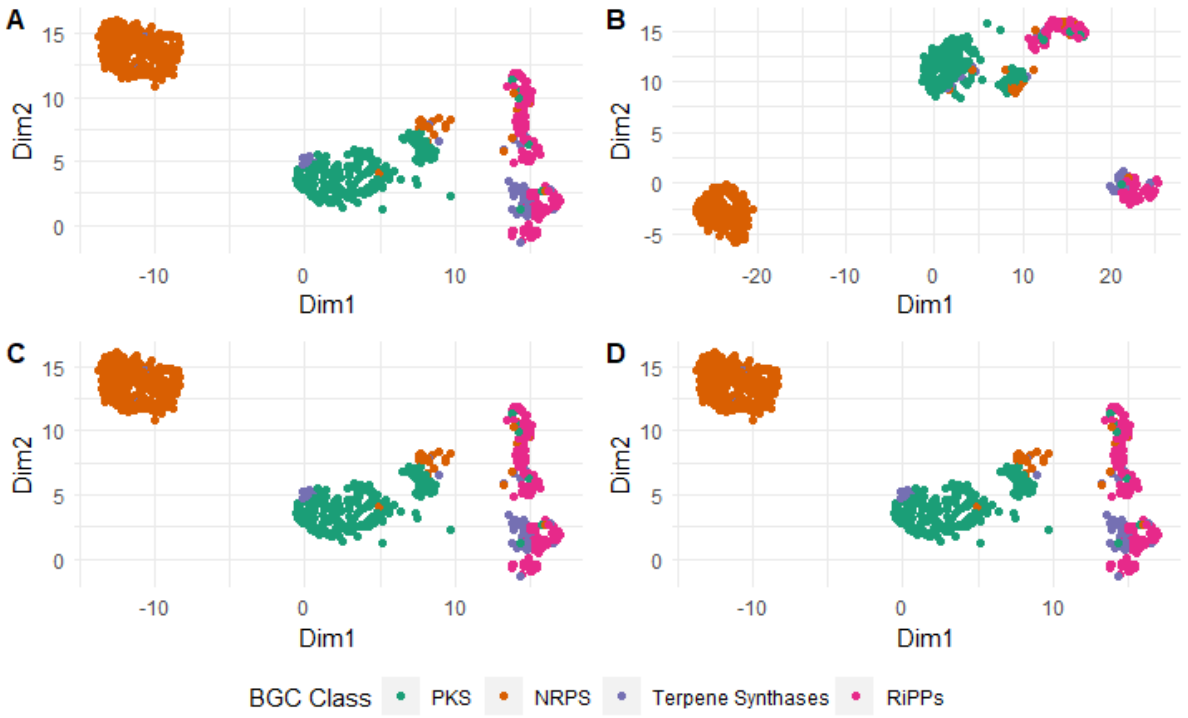


Figure 3. 7: ***Visualization of low dimensional vectors produced by Supervised UMAP from MIBiG BGCs. Protein sequences for each BGC were converted into vectors by concatenating each protein within the BGC****. UMAP was used for dimensional reduction. The resulting low dimensional space was visualized in two dimensions. Each panel uses different input metrics; A) Input metric: Chebyshev, Output metric: Euclidean, Neighbors: 30, Min. Distance: 0.1 and Weight: 0.3 B) Input metric: Chebyshev, Output metric: Chebyshev, Neighbors: 45, Min. Distance: 0.2 and Weight: 0.3 C) Input metric: Chebyshev, Output metric: Euclidean, Neighbors: 30, Min. Distance: 0.2 and Weight: 0.3 and D) Input metric: Euclidean, Output metric: Chebyshev, Neighbors: 15, Min. Distance: 0.1 and Weight: 0.3.*

Table 3. 3: ***Parameters and metrics used to obtain the best result from dimensionality reduction using UMAP****. UMAP models were fitted with varying parameters to transform data into three components. A trustworthiness analysis was performed to detect the low dimensional datasets fidelity to the high dimensional dataset obtained by transforming BGCs from the MIBiG database. The average trustworthiness was calculated by randomly selecting 80% of vectors, both transformed and untransformed, within the data set. The process was repeated ten times.*

UMAP model	Input Metric	Output Metric	Neighbors	Min. Distance	Weight	Trustworthiness (%)
Model A	Chebyshev	Euclidean	30	0.1	0.3	87.88 ± 0.34
Model B	Chebyshev	Chebyshev	45	0.2	0.3	86.31 ± 0.22
Model C	Chebyshev	Euclidean	30	0.2	0.3	84.54 ± 0.26
Model D	Euclidean	Chebyshev	15	0.1	0.3	90.46 ± 0.38

The output of these sUMAP models was used to train several classifiers. The dataset was partitioned into a training and testing set comprising 64% and 34% of the data. The data set was used for several sUMAP models with varying parameters, and the average trustworthiness score was calculated for each model. Table 3. 3 shows the sUMAP models with the best trustworthiness score. The output of these models was used to train a random forest classifier, and several classification metrics were obtained. The metrics chosen were balanced accuracy, Kappa statistic and MCC and were all above our benchmark of 80%, 0.7 and 0.7, respectively Table 3. 4[67,89–91]. Other classifiers such as a MLP and AdaBoost with a random forest produce similar results. This data suggests that dimensionality reduction produces datasets that can be effectively modelled with low-capacity models[98].

*Table 3. 4**: **A Random Forest was fitted using parameters from UMAP models as described in Table 3. 3.** Classification metrics such as accuracy, balanced accuracy, Kappa statistic and MCC were calculated. The classification metrics were calculated by performing a repeated stratified k-fold cross-validation with three splits and ten repetitions. The MIBiG database was split into the ratio of 2:1, with the more extensive set used to train the algorithm and the smaller set for validation.*

UMAP model	Accuracy (%)	Balanced Accuracy (%)	Kappa	MCC
Model A	91.65 ± 5.61	84.164 ± 4.67	0.8724 ± 0.0834	0.875 ± 0.067
Model B	90.4 ± 4.2	85.475 ± 5.32	0.867 ± 0.0782	0.87 ± 0.071
Model C	89.46 ± 4.82	82.124 ± 4.17	0.883 ± 0.0694	0.88 ± 0.076
Model D	88.67 ± 5.72	81.61 ± 4.23	0.8133 ± 0.0743	0.81 ± 0.065

A confusion matrix using sUMAP Model C (Table 3. 5) and a random forest with default parameters was built to determine how the model performed for each class. The results are shown in Table 3. 5. The model identified PKS, NRPS and RiPPs with high accuracy. Terpene synthases proved particularly difficult for the classifier, with many misclassified as RiPPs. This result can be rationalized based on observations from the dimensionality reduction step (Table 3. 7). Cluster analysis by k-means and DBSCAN differentially clustered RiPP and terpene synthase BGCs suggests that generating reliable clusters for these two BGC types may be challenging. Regardless, the classifier's general performance was acceptable, and thus we chose to evaluate the machine learning pipeline model with data it had not yet been exposed to.

*Table 3. 5: **Normalized confusion matrix obtained from a Random Forest Classifier. A UMAP model was fitted with input metric: Chebyshev, output metric: Euclidean, Neighbors: 30, min. distance: 0.1and weight: 0.3 to output vectors of three dimensions**. The transformed testing set was evaluated using a random forest and assessed. Sequences were obtained from the MIBiG database and converted to vectors using BioVec, split into the ratio 2:1 for training and testing respectively.*

	Polyketide Synthases	NRPS	Terpene Synthases	RiPPs
Polyketide Synthases	97.5	0	2.5	0
NRPS	2.3	94.7	3	0
Terpene Synthases	0	4	76	20
RiPPs	0	0	5	95

3.5. The machine learning-based pipeline model classifies BGC types with high accuracy

The antiSMASH database consists of BGCs that have been predicted computationally. However, the vast majority have not been experimentally validated[66]. Our pipeline, ML-Miner, was thus evaluated with an independent dataset comprising sequences uniquely present within the antiSMASH database[66]. ML-Miner processed the sequences, transforming them into vectors using BioVec, using sUMAP for dimensionality reduction, and classifying them using a random forest[74,84,89].

Table 3. 6: ***Normalized confusion matrix obtained from a Random Forest Classifier****. Sequences from the antiSMASH dataset were transformed into vectors using BioVec. Dimensional reduction using the supervised UMAP model A described in Table 3. 3 provided input for training (67% of the dataset) and testing (33the % of the dataset) the Random Forest classifier.*

	Polyketide Synthases	NRPS	Terpene Synthases	RiPPs
Polyketide Synthases	84	5	4	7
NRPS	3.4	85	4.1	7.5
Terpene Synthases	3.7	4.3	85	7
RiPPs	4	5	4.5	86.5

The normalized confusion matrix for this analysis is shown in Table 3. 6. The pipeline mapped more than 80% of each BGC sequence per class into their respective class. Interestingly a substantial increase in the accurate classification of terpenes synthase BGCs was observed, with terpene synthases BGCs now classified with 85% accuracy. These results suggest that the pipeline generalizes well, as antiSMASH contains more evolutionary divergent sequences than the MIBiG dataset[65]. It should be noted that the antiSMASH database is largely non-curated and contains sequences with assembly artifacts and BGCs that have been misassigned to a BGC type[66]. Thus, the error rate in the dataset is much higher than in the MIBiG dataset, which may impact the results in the confusion matrix, potentially artifactually reducing the accuracy.[22,65,66]

Table 3. 7: ***Statistics used to evaluate random forests within the pipeline. Statistics were calculated to assess the pipeline's performance and determine how the model was generalized.*** *The classification metrics were calculated by performing a repeated stratified k-fold cross-validation with three splits and ten repetitions.*

Accuracy (%)	Balanced Accuracy (%)	Kappa statistic	Matthews Correlation Coefficient
85.42 ± 0.79	85.05 ± 0.65	0.796 ± 0.011	0.796 ± 0.023

Evaluating ML-Miner using sequences present uniquely within the antiSMASH database provides a good test of how well our model generalizes[66]. With classification statistics above ~80% or more (Table 3. 7), our model identifies more than 80% of BGCs per class within the antiSMASH database, suggesting that it is robust when analyzing new sequences[66]. The high classification metrics can be attributed to BioVec's ability to capture biological and phylogenetic relationships, which are retained when performing dimensionality reduction using supervised UMAP[74,84]. The ability of the upstream modules to capture and retain biological relevant information assists in efficient downstream classification.

Hyperparameters are parameters whose values control the learning process of a model[67]. They control the behaviour of the training algorithm and have a significant impact on the performance of the model being trained[67]. Using a grid search and testing set generated from the MIBiG dataset, hyperparameters that have an essential role in the learning process of the random forest were identified.[65]

Table 3. 8: ***Hyperparameters used to train a random forest. The model was prepared using the transformed training set from protein vectors of biosynthetic gene clusters from the MIBiG database****. A grid search was performed on a random forest to identify hyperparameters that would provide the best results.*

Parameters	Values
bootstrap	False
n_estimators	10
max_features	auto
criterion	Gini
max_depth	60
min_samples_split	5
min_samples_leaf	2

Using a random forest with parameters as described in Table 3. 8, the pipeline was evaluated using a subset of the antiSMASH database. A confusion matrix was obtained to determine how well the model performed per class[66]. The results are as shown in Table 3. 9. The number of correctly classified BGCs increased to ~95%. These results suggest that our newly developed strategy, *ML-Miner* performed on par with the most used BGC mining tool, antiSMASH.

*Table 3. 9****: UMAP model was fitted with input metric: Chebyshev, output metric: Euclidean, Neighbors: 30, min. distance: 0.1 and weight: 0.3****. The transformed testing set was evaluated using a random forest with parameters described in* ***Error! Reference source not found.****. Sequences were obtained from the a ntiSMASH database and converted to vectors using BioVec.*

	Polyketide Synthases	NRPS	Terpene Synthases	RiPPs
Polyketide Synthases	95	4	04	1

NRPS	3	95	1	1
Terpene Synthases	1	1	96	4
RiPPs	1	0	3	94

Finally, classification metrics were further calculated to determine how the model generalized when exposed to novel information Table 3. 10. These results suggest that the model generalized well, with each metric higher than 90%, reinforcing the notion that the pipeline's performance is on par with the antiSMASH software[56].

Table 3. 10: ***Statistics used to evaluate the classifier within the pipeline's performance after hyperparameter optimization****. Statistics were calculated to assess the pipeline's performance and determine how the model was generalized. The classification metrics were calculated by performing a repeated stratified k-fold cross-validation with three splits and ten repetitions.*

Accuracy	Balanced Accuracy	Kappa statistic	Matthews Correlation Coefficient
0.9851 ± 0.67	0.9532 ± 0.57	0.9051 ± 0.033	0.9133 ± 0.013

CHAPTER FOUR: DISCUSSION AND CONCLUSIONS

In chapter two, we developed a machine learning-based pipeline, ML-Miner, to classify BGC types starting from GenBank files containing sequence data for BGCs. The pipeline consists of a Natural Language Processing model based on BioVec, a dimensionality reduction module relying on supervised UMAP, and a random forest classifier. Evaluation of the pipeline with new data from the antiSMASH database showed that the model provided high accuracy (~95%) in assigning BGCs to PKS, NRPS, RiPP or terpene BGC types generalized well.

4.1. Natural Language Processing extracts meaningful biological information from protein sequences

Most bioinformatics tools for the analysis of BGCs rely on HMMs. HMMs are probabilistic models of protein families or domains generated from multiple sequence alignments that allow for identifying protein families or classes[70,71]. antiSMASH, the leading tool in BGC identification and annotation, uses HMM and a rule-based algorithm to identify and classify BGCs[59,66,70,71]. While HMMs are more effective at identifying homologous sequences than pairwise comparisons, they suffer from a high rediscovery, i.e. they identify sequences that are very similar to those used to build the model[42,53]. An essential aspect of HMMs is that the alignment of the model occurs with text-based protein sequence data.

To implement a machine learning classification strategy that we hypothesized would be more generalizable than antiSMASH, we chose to embed the text-based protein sequence data into a lower-dimensional vector representation. NLP allows for efficient analysis of sequences through its ability to extract more abstract rules within the dataset[73]. Using NLP, we hypothesized that more abstract rules defining BGCs could be captured in the vector representations of the BGCs. Several embeddings have been developed to capture relevant information from biological sequences to build practical artificial intelligence algorithms[74,101]. DeepFRI uses an LSTM to model sequences within the Pfam database[102]. LSTM captures dependencies between amino acids and generates a model that captures the depth of variation within a protein family[42,102]. Combining these models with a graph convolutional neural network and gradient class activating maps, the authors were able to classify proteins into their respective classes and predict enzyme commissions and gene ontologies with high accuracy[102]. DeepBGC employs a similar strategy to model sequences from the Pfam database[42,59]. Here, LSTM is used to model sequences from the Pfam database and allows for identifying Pfam domains involved[42]. The resulting predictions are scored to determine which ones would encode for a domain involved in biosynthesis[42]. This strategy has been used to identify several putative BGCs, further analyzed *in vivo* to confirm their assignment as BGCs [42].

ML-Miner uses *BioVec* to transform the protein sequences from a BGC into a continuous distributed vector representing the BGC[74]. *BioVec* uses word2vec and employs both a one-hot encoding and skip-gram neural network architectures to extract from sequences meaningful biological information, such as biophysical and biochemical properties or phylogenetic information intrinsic to proteins[74]. The derived distributed representation of biological sequences was shown to classify protein sequences within their respective Pfam domains with high accuracy and identify the biophysical properties of proteins[59,74]. Because the original trained model was not available, we retrained *BioVec*. Further, we validated the retrained model by calculating the cosine similarity scores that generated the same relationship expected from sequence alignment between homologous sequences (Figure 3. 2).

4.2. Supervised UMAP retains biological and phylogenetic relationships stored in vectors generated using *BioVec*

Classification of the high dimensional dataset generated from the *BioVec* embedding was unsuccessful. Dimensionality reduction, however, generated a lower-dimensional representation that showed clustering consistent with the known BGC type. While dimensionality reduction reduces the complexity of the dataset, and as such, the information content of the dataset, thereby increasing the ability of a classifier to model the data accurately, it can, in some instances, improve the performance of classifiers. In particular, dimensionality reduction may make the dataset better suited to low-capacity algorithms such as the random forest algorithm. We thus chose to implement a dimensionality reduction step.

With the rapid growth of biological data and the rise in the application of machine learning to analyze biological data, dimensional reduction techniques have taken the forefront[83–85,103]. These algorithms typically rely on finding patterns within the dataset to create a low-dimensional representation, thereby reducing the model's complexity and the risk of overfitting[85]. PCA is commonly used in metabolomics studies to identify biomarkers that distinguish features between two populations. Moreover, transcriptomic analyses have used t-SNE to determine factors between cells or specify a group of cells with similar expression profiles[83]. The ability of t-SNE to distinguish cell populations or genes within an organism based on their expression profiles provides a framework to visualize the dataset but, more importantly, ask better and more informed questions[83]. A drawback with t-SNE is that it does not keep the global structure of the dataset; hence relationships between isolated clusters are lost[83,84,97,103]. UMAP has been proposed as the dimensional reduction algorithm that keeps the high dimensional dataset's global and local structure[84,97]. UMAP's ability to conserve the global structure of the dataset has gained popularity in recent years and, most importantly, in transcriptomics[97]. It has been used to identify proteins that cooperate within a biological pathway and have a similar function within a cellular context [97].

We thus implement these dimensionality reduction strategies. Ultimately, supervised UMAP showed superior performance in forming clusters within the dataset. A Hopkins test and Silhouette analysis showed that the dataset had a clustering tendency. Furthermore, DBSCAN showed that the clusters mapped to the BGC type are available from the MIBiG database as an accurate benchmark. A critical observation was the effective clustering of BGCs that mapped to terpene and RiPP BGC type, as these were challenging to cluster correctly in some of the models. This suggests that supervised UMAP retained the necessary information stored within the dataset. The supervised UMAP models also had a high trustworthiness score, implying that global information within the high dimensional dataset was effectively transferred to the low dimensional representation[85].

We were particularly surprised by the result of the PCA analysis. While BioVec generates highly correlated variables based on its embedding strategy, the data is not unidimensional, as demonstrated by the t-SNE and UMAP dimensionality reduction strategies. PCA relies on an orthogonal linear transformation, transforming the variables into a new coordinate where the most significant variance resides on the first coordinate. In our PCA analysis, more than 95% of the variance was explained by the first coordinate, suggesting a single variable could describe the dataset. The results obtained from the PCA analysis could be attributed to either technical or methodological error; however, the success of the non-linear strategies such as UMAP enabled us to move forward in developing *ML-Miner.*

4.3. *ML-Miner* identifies more than 80% of BGCs within the antiSMASH database

Identifying putative BGCs is one of the main challenges of modern NP research[42,56,60]. *ML-Miner* aims to identify putative BGCs that can then be experimentally validated to confirm the presence of a natural product. We used several classifiers to classify BGCs from the low dimensional embedding created by supervised UMAP. Models were trained with two-thirds of the MIBiG dataset and tested with the remaining third of the dataset. A random forest classifier identified each BGC type with a more than 80% balanced accuracy, as shown in Table 3. 6[65].

The pipeline was further evaluated with an independent set of BGCs obtained from the antiSMASH database[66]. This dataset contains BGC that the antiSMASH software has computationally predicted compared to the MIBiG dataset, which only contains BGC that have been experimentally validated[65]. It was observed that the generalized pipeline well as it was able to correctly classify ~80% of sequences within the antiSMASH database per class after hyperparameter optimization[66]. Our analysis suggests that the *ML-Miner* is as good as antiSMASH when used to determine the type of BGCs since the confusion matrix showed ~95% agreement with antiSMASH for all predicted BGC types[56]. *ML-Miner* generalizes well by its ability to effectively identify BGCs that it has ever been exposed to before, reiterating that biological information within protein sequences is retained within the model, making it a robust model for mining BGCs identifying novel natural products.

4.4. Comparison to other BGC discovery tools

antiSMASH was a pioneer in NP research and revolutionized BGC identification from genomic datasets. Its framework relied on HMMs to identify BGCs[58]. Using HMMs, antiSMASH models core domains involved in secondary metabolisms and capture the variance within the protein class[70,71]. As a result of this, however, antiSMASH suffers from a rediscovery problem[42,106]. It identifies very similar sequences to the sequences used to build the model and has difficulty when exposed to more divergent sequences[58]. In addition, due to the short length and hyper-variability of RiPP sequences, GLIMMER, which is used to identify coding regions, often does not detect the genes, and HMMs are not well suited to identify them even when they are called genes.[60]. To address the challenges of identifying RiPP BGCs, RODEO was developed. RODEO is dedicated to identifying RiPPs by combining HMMs and SVMs[13]. While it uses a modified approach to calling potential core RiPP genes, like antiSMASH, RODEO has the same limitation because it uses HMMs to characterize the RiPP core domain[13,58]. RODEO has now been implemented as part of the pipeline in antiSMASH, improving RiPP BGC annotation, though effective identification of these BGCs still lags behind other BGC types.

DeepBGC was the first tool to apply artificial intelligence to identify BGCs[42]. DeepBGC employs NLP to identify domains used in secondary metabolism using the Pfam database[42,59,72]. The BGC protein-vector space produced from NLP is analyzed using an LSTM as this RNN architecture allows for the identification of dependencies between amino acids with a BGC class[42]. DeepBGC allowed the Merck team to identify several candidate BGCs for further experimental validation, demonstrating its utility for natural product discovery. The reliance of DeepBGC on Pfam domains is potentially problematic. However, the discovery of new BGCs. New functional domains for which Pfam families do not yet exist will not be included in DeepBGC, and thus radically new BGCs may be missed using this approach.

We sought to develop a machine learning approach for BGC type assignments that did not rely on HMMs or rule-based algorithms, like antiSMASH[58]. Our strategy relies on NLP to transform protein sequences to their vector representation using a biological embedding – *BioVec.* This fundamentally differentiates our method from both DeepBGC and antiSMASH. *BioVec* produces vectors of high dimension[73]. The analysis of the high dimensional vectors is computationally burdensome; hence we employed UMAP dimensionality reduction to produce a robust low dimensional representation of the dataset [83–85,99,104]. The dimensionality reduction model was evaluated using a trustworthiness analysis[85].

Lastly, a classifier was trained and evaluated using accuracy, balanced accuracy, and other classification metrics to determine the pipeline's ability to perform classification[67,89]. This generated a robust pipeline, *ML-Miner*.

It will be necessary to compare it to other AI-based tools, including DeepRipp and DeepBGC, to evaluate the robustness of *ML-Miner*. DeepRiPP combines a multi-omics and computational strategy to identify new RiPP gene clusters. This comparison would provide a further test with the benefit of combining both experimental and computational modes when identifying new BGCs. On the other hand, DeepBGC uses HMMs to identify domains, which are further processed using Pfam2vec, an NLP embedding that converts Pfam domains to vectors. This strategy is limited since not all domains involved in secondary metabolism have a well-defined Pfam domain, as seen in RiPPs. Since *ML-Miner* does not use HMMs to identify domains involved in secondary metabolism as done by DeepBGC, this test would evaluate the ability of NLP to define the rules of what defines a BGC implicitly.

ML-Miner thus provides a steppingstone for developing an unbiased tool for new BGC discovery. However, several additional steps must be taken before deploying and using it. Below are some recommendations to make *ML-Miner* a more robust tool.

4.5. Alternative solutions to dimensionality reduction

The incompatibility of the high dimensional dataset with the low-capacity classifier led to the implementation of a dimensionality reduction step. Several possible modifications to the pipeline to eliminate this intermediate step. One is to develop a new embedding that generates a lower dimension dataset than *BioVec*. This new embedding would replace the dimensionality reduction and classification modules of *ML-Miner*.

Advances in deep learning have allowed for the development of algorithms that derive vector representation of sequences and simultaneously perform dimensionality reduction and classification. Language models have been developed using LSTMs, that[74] allow for modelling the relationships between amino acids[42,74]. In this scenario, amino acids are treated as time-series datasets.
LSTM language models model the relationship between amino acids within a protein family and capture the wealth and depth of biological complexity[74]. Furthermore, with an abundance of protein structures within the PDB database involved in biosynthesis, graph convolutional neural networks (GCNN) can be used to model the 3D structures of proteins[74,109]. Combining LSTM language models and GCNNs, classification models can be developed to identify protein sequences involved in biosynthesis[74].

4.6. Challenges adapting BGCs to *BioVec* encoding

BGCs, responsible for constructing complex natural products, typically encode multiple proteins. *BioVec,* however, encodes a single protein into a vector representation. We concatenated all the protein sequences encoded in the BGC, starting with the first gene in the GenBank file.

It is important to note that while MIBiG contains highly curated and the best annotated BGCs, even in this database, the exact number of genes required for a biologically complete and functional BGC is poorly defined. Some genes at the 5' and 3' end of BGCs in the MIBiG database may not play a role in biosynthesis. Furthermore, the BGCs defined in the antiSMASH database are well known to have poorly defined starts and stops. antiSMASH relies on a simple distance metric from the last core gene detected at the 5' and 3' end of a BGC to define the start and stop of the gene cluster. Thus, evaluating the performance of ML-Miner to classify BGCs where some of the 5' and 3' encoded proteins are removed will be of particular interest.

4.7. An interface enabling *ML-Miner* to analyze genomic data

ML-Miner currently does not make predictions based on genomic sequences. It uses GenBank files of BGCs. Hence a preprocessing module must be added to the pipeline to convert genomes into potential BGCs Figure 4. 1. The envisioned preprocessing module would consist of three steps. The first step would identify open reading frames within the genomic sequences and triage overlapping open reading frames to identify actual genes. The second would remove core parts of the genome widely conserved across bacterial species and thus constitute the minimal bacterial genome [70,71,107]. As BGCs are not highly conserved across bacteria, they will not be found in this core genome. Finally, the third step would involve building putative clusters of many genes. As BGC has been shown to repurpose genes involved in primary metabolism, putative BGCs with at most three genes involved in primary metabolism would be prioritized for analysis. These are expected to give a better chance of identifying novel BGCs [4,102,108].

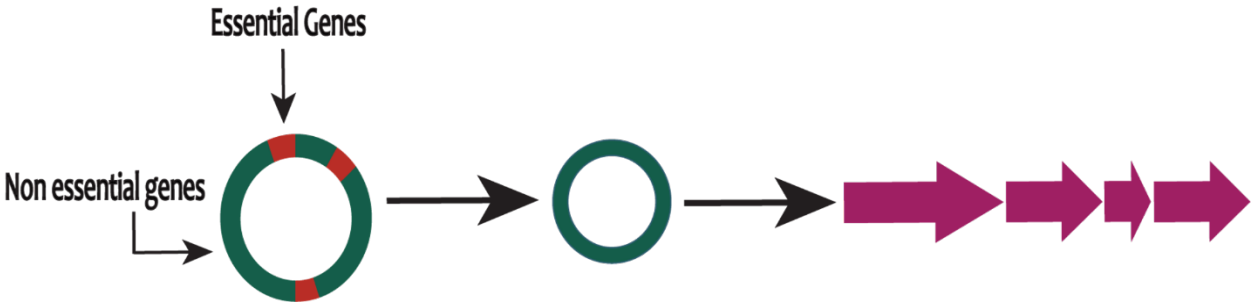


Figure 4. 1: **Illustration of a proposed strategy to create putative biosynthetic gene clusters from genomes**. HMMs of essential genes obtained from the minimal genome would be built and used to identify genes involved in secondary metabolism. The resulting partial genome would be used to create putative gene clusters containing at most three genes involved in primary metabolism.

4.8. Confidence scores

ML-Miner does not provide a method to determine how robust the classification results are. For example, it does not produce a p-value, an e-value, or a false discovery rate. There are several potential strategies to implement a confidence score. For example, a probabilistic random forest classifier could be implemented. Alternatively, a database of BGCs with random genes could be analyzed to determine the frequency with which random gene clusters get assigned to different BGC types. Alternatively, or potentially in combination, individual genes could be dropped from a BGC and reclassified. The number of dropped genes required to change classification could provide a metric for the strength of the assignment.

4.9. Limitation

To identify the most informative supervised UMAP models, the same data (MIBIG dataset) used to fit the UMAP models was also used to train and test several classifiers. Since the supervised dimensionality reduction had access to the labels, there is a strong possibility of data leakage. To mitigate this, we used the antiSMASH dataset, a dataset built of sequences present within the antiSMASH database but not the MIBiG database, to validate the results obtained using the MIBiG dataset. Even though sequences within the antiSMASH database have been hypothesized to have more evolutionary diverse sequences than the MIBiG database, this is because if the majority of the sequences within the antiSMASH dataset are 80% - 90% similar to those within the MIBiG dataset, the

results obtained may be overoptimistic. An analysis should be performed to validate this claim as a result of this analysis would assist in validating the classification metrics obtained.

ML-Miner currently identifies four biosynthetic gene cluster types, *i.e.* terpenes, PKSs, NRPSs and RiPPs. These are the most common types of BGCs observed in bacterial genomes. However, a negative or non-BGC type would be a valuable addition to the classifier since many groups of genes are not BGCs. Because the pipeline has not been exposed to non-BGCs, we cannot predict how it will respond to them. To address this, examples of actual sequences known not to be part of BGCs or randomly generated protein sequences of similar amino acid composition as those present within biosynthetic enzymes involved in secondary metabolism could be added to the machine learning pipeline to enable *ML-Miner* to label non-BGC sequences correctly.

When applying dimensionality reduction, the number of dimensions generally retained is usually high in the order of 50 dimensions. In our study, we used three dimensions to train the models. In future efforts, examining the impact of increasing dimensionality on the performance of the pipeline would be significant.

4.10. Conclusion

In this thesis, we develop a machine learning strategy for classifying BGCs. This pipeline relies on NLP embedding using *BioVec*, a skip-gram algorithm, to produce machine learning ready vectors from biological sequences. The high dimensional vectors produced from *BioVec* are transformed using supervised UMAP into a low dimensional representation. These low-dimensional vectors were used to train a random forest classifier, which was able to classify BGCs into their respective classes with high accuracy. This represents the first machine learning strategy for BGC discovery that does not rely upon any point on HMM to identify protein domains and is thus fully independent of multiple sequence alignments based on known protein-coding sequences.

Appendix

List of parameters

Table A. 1: Summary of parameters used in building the t-SNE model as described by the sklearn.manifold API

Parameters	Description	Values
n_components	Dimension of the embedded space.	2
perplexity	The number of nearest neighbors that are used during manifold learning.	5, 15, 30, 45, 60
learning_rate	Tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.	auto
metric	Metric to use when calculating distance between instances in a feature array.	Euclidean, Manhattan and Chebyshev
random_state	Determines the random number generator.	1042
n_jobs	Number of parallel jobs to run for neighbor search.	-1

Table A. 2: Summary of parameters used in building the UMAP model as described by the API

Parameters	Description	Values
n_neighbors	The size of the local neighborhood used for manifold approximation.	15, 30, 45, 60s
n_components	The dimension of the space to embed into.	2, 3
metric	The metric to use to compute distances in high dimensional space.	Euclidean, Manhattan and Chebyshev
min_dist	The effective minimum distance between embedded points.	0.1, 0.2, 0.3, 0.4
random_state	A state capable being used as a numpy random space	1042

output_metric	Function returning the distance between the two points in the embedig space and the gradient of the distance with respect to the first argument.	Euclidean, Manhattan and Chebyshev
target_weight	Weight factor between data topology and target topology	0.1, 0.2, 0.3, 0.4

References

1. Katz, L. & Baltz, R. H. Natural product discovery: past, present, and future. *Journal of Industrial Microbiology and Biotechnology* **43**, 155–176 (2016).
2. Zhang, M. M., Qiao, Y., Ang, E. L. & Zhao, H. Using natural products for drug discovery: the impact of the genomics era. *Expert Opinion on Drug Discovery* **12**, 475–487 (2017).
3. Clardy, J. & Walsh, C. Lessons from natural molecules. *Nature* **432**, 829–837 (2004).
4. Adamek, M., Alanjary, M. & Ziemert, N. Applied evolution: phylogeny-based approaches in natural products research. *Natural Product Reports* **36**, 1295–1312 (2019).
5. Rutledge, P. J. & Challis, G. L. Discovery of microbial natural products by activation of silent biosynthetic gene clusters. *Nature Reviews Microbiology* **13**, 509–523 (2015).
6. Keller, L. & Surette, M. G. Communication in bacteria: an ecological and evolutionary perspective. *Nature Reviews Microbiology* **4**, 249–258 (2006).
7. Boto, L. Horizontal gene transfer in evolution: facts and challenges. *Proceedings of the Royal Society B: Biological Sciences* **277**, 819–827 (2010).
8. Scherlach, Kirstin; Hertweck, C. Chemical Mediators at the Bacterial-Fungal Interface. *Annual Reviews Microbiology* **74**, 267–290 (2020).
9. Tyurin, A. P., Alferova, V. A. & Korshun, V. A. Chemical Elicitors of Antibiotic Biosynthesis in Actinomycetes. *Microorganisms* **6**, (2018).
10. Zarins-Tutt, J. S. *et al.* Prospecting for new bacterial metabolites: a glossary of approaches for inducing, activating and upregulating the biosynthesis of bacterial cryptic or silent natural products. *Natural Product Reports* **33**, 54–72 (2016).
11. Aggarwal, E., Chauhan, S. & Sareen, D. Thiopeptides encoding biosynthetic gene clusters mined from bacterial genomes. *Journal of Biosciences* **46**, 36–48 (2021).
12. Georgiou, M. A., Dommaraju, S. R., Guo, X., Mast, D. H. & Mitchell, D. A. Bioinformatic and Reactivity-Based Discovery of Linaridins. *ACS Chemical Biology* **15**, 2976–2985 (2020).
13. Georgiou, M. A., Dommaraju, S. R., Guo, X., Mast, D. H. & Mitchell, D. A. Bioinformatic and Reactivity-Based Discovery of Linaridins. *ACS Chemical Biology* **15**, 2976–2985 (2020).
14. Cox, C. L. *et al.* Nucleophilic 1,4-Additions for Natural Product Discovery. *ACS Chemical Biology* **9**, 2014–2022 (2014).
15. Gulder, T. A. M. & Moore, B. S. Salinosporamide Natural Products: Potent 20 S Proteasome Inhibitors as Promising Cancer Chemotherapeutics. *Angewandte Chemie International Edition* **49**, 9346–9367 (2010).
16. Kale, A. J., McGlinchey, R. P., Lechner, A. & Moore, B. S. Bacterial Self-Resistance to the Natural Proteasome Inhibitor Salinosporamide A. *ACS Chemical Biology* **6**, 1257–1264 (2011).
17. Donia, M. S. & Fischbach, M. A. Small molecules from the human microbiota. *Science (1979)* **349**, (2015).
18. Tran, P. N., Yen, M.-R., Chiang, C.-Y., Lin, H.-C. & Chen, P.-Y. Detecting and prioritizing biosynthetic gene clusters for bioactive compounds in bacteria and fungi. *Applied Microbiology and Biotechnology* **103**, 3277–3287 (2019).

19. Feng, L., Gordon, M. T., Liu, Y., Basso, K. B. & Butcher, R. A. Mapping the biosynthetic pathway of a hybrid polyketide-nonribosomal peptide in a metazoan. *Nature Communications* **12**, 4912 (2021).
20. Ren, H., Shi, C. & Zhao, H. Computational Tools for Discovering and Engineering Natural Product Biosynthetic Pathways. *iScience* **23**, 100795–100810 (2020).
21. Oluwatayo, Makinde A.; Bezuidt, Oliver K.I.;makhalanyane, T. P. ; Metagenomic tools for unravelling novel polyketides. *Computational and Structural Biotechnology Journal* (2020).
22. Medema, M. H. *et al.* antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences. *Nucleic Acids Research* **39**, W339–W346 (2011).
23. Hetrick, K. J. & van der Donk, W. A. Ribosomally synthesized and post-translationally modified peptide natural product discovery in the genomic era. *Current Opinion in Chemical Biology* **38**, 36–44 (2017).
24. Russell, A. H. & Truman, A. W. Genome mining strategies for ribosomally synthesised and post-translationally modified peptides. *Computational and Structural Biotechnology Journal* **18**, 1838–1851 (2020).
25. Valverde, J. P. & Hanson, P. Parenchyma: a neglected plant tissue in the Cecropia/ant mutualism. *Symbiosis* **55**, 47–51 (2011).
26. Rickson, F. R. Anatomical Development of the Leaf Trichilium and Mullerian Bodies of Cecropia peltata L. *American Journal of Botany* **63**, 1266 (1976).
27. Dejean, A. *et al.* Arboreal Ants Use the "Velcro® Principle" to Capture Very Large Prey. *PLoS ONE* **5**, e11331 (2010).
28. Agrawal, A.A; Dublin-Thaler, B. J. Induced responses to herbivory in the Neotropical ant-plant association between Azteca ants and Cecropia trees: response of ants to potential inducing cues. *Behav. Ecol. Sociobiol.* **45**, 47–54 (1999).
29. George, K. M. *et al.* Mycolactone: A Polyketide Toxin from *Mycobacterium ulcerans* Required for Virulence. *Science (1979)* **283**, 854–857 (1999).
30. Newman, D. J. & Cragg, G. M. Natural Products as Sources of New Drugs over the Nearly Four Decades from 01/1981 to 09/2019. *Journal of Natural Products* **83**, 770–803 (2020).
31. Wright, P. M., Seiple, I. B. & Myers, A. G. The Evolving Role of Chemical Synthesis in Antibacterial Drug Discovery. *Angewandte Chemie International Edition* **53**, 8840–8869 (2014).
32. Faron, M. L., Ledeboer, N. A. & Buchan, B. W. Resistance Mechanisms, Epidemiology, and Approaches to Screening for Vancomycin-Resistant Enterococcus in the Health Care Setting. *Journal of Clinical Microbiology* **54**, 2436–2447 (2016).
33. Blair, J. M. A., Webber, M. A., Baylay, A. J., Ogbolu, D. O. & Piddock, L. J. v. Molecular mechanisms of antibiotic resistance. *Nature Reviews Microbiology* **13**, 42–51 (2015).
34. Hecht, S. M. Bleomycin: New Perspectives on the Mechanism of Action. *Journal of Natural Products* **63**, 158–168 (2000).
35. Tacar, O., Sriamornsak, P. & Dass, C. R. Doxorubicin: an update on anticancer molecular action, toxicity and novel drug delivery systems. *Journal of Pharmacy and Pharmacology* **65**, 157–170 (2012).

36. Meng, L. *et al.* Epoxomicin, a potent and selective proteasome inhibitor, exhibits *in vivo* antiinflammatory activity. *Proceedings of the National Academy of Sciences* **96**, 10403–10408 (1999).
37. Montalbán-López, M. *et al.* New developments in RiPP discovery, enzymology and engineering. *Natural Product Reports* **38**, 130–239 (2021).
38. Winn, M., Fyans, J. K., Zhuo, Y. & Micklefield, J. Recent advances in engineering nonribosomal peptide assembly lines. *Natural Product Reports* **33**, 317–347 (2016).
39. Rudolf, J. D. & Chang, C.-Y. Terpene synthases in disguise: enzymology, structure, and opportunities of non-canonical terpene synthases. *Natural Product Reports* **37**, 425–463 (2020).
40. Gershenzon, J. & Dudareva, N. The function of terpene natural products in the natural world. *Nature Chemical Biology* **3**, 408–414 (2007).
41. Medema, M. H. *et al.* Minimum Information about a Biosynthetic Gene cluster. *Nature Chemical Biology* **11**, 625–631 (2015).
42. Hannigan, G. D. *et al.* A deep learning genome-mining strategy for biosynthetic gene cluster prediction. *Nucleic Acids Research* **47**, e110–e110 (2019).
43. Baunach, M., Chowdhury, S., Stallforth, P. & Dittmann, E. The Landscape of Recombination Events That Create Nonribosomal Peptide Diversity. *Molecular Biology and Evolution* **38**, 2116–2130 (2021).
44. Kondrashov, F. A. Gene duplication as a mechanism of genomic adaptation to a changing environment. *Proceedings of the Royal Society B: Biological Sciences* **279**, 5048–5057 (2012).
45. Demay, J., Bernard, C., Reinhardt, A. & Marie, B. Natural Products from Cyanobacteria: Focus on Beneficial Activities. *Marine Drugs* **17**, 320 (2019).
46. McDonald, B. R. & Currie, C. R. Lateral Gene Transfer Dynamics in the Ancient Bacterial Genus *Streptomyces*. *mBio* **8**, (2017).
47. Doroghazi, J. R. *et al.* A roadmap for natural product discovery based on large-scale genomics and metabolomics. *Nature Chemical Biology* **10**, 963–968 (2014).
48. Pye, C. R., Bertin, M. J., Lokey, R. S., Gerwick, W. H. & Linington, R. G. Retrospective analysis of natural products provides insights for future discovery trends. *Proceedings of the National Academy of Sciences* **114**, 5601–5606 (2017).
49. Pye, C. R., Bertin, M. J., Lokey, R. S., Gerwick, W. H. & Linington, R. G. Reply to Skinnider and Magarvey: Rates of novel natural product discovery remain high. *Proceedings of the National Academy of Sciences* **114**, (2017).
50. Bassler, B. Qurorum Sensing: How Bacteria Communicate. *Explore Biology* (2019).
51. Xu, F., Nazari, B., Moon, K., Bushin, L. B. & Seyedsayamdost, M. R. Discovery of a Cryptic Antifungal Compound from *Streptomyces albus* J1074 Using High-Throughput Elicitor Screens. *J Am Chem Soc* **139**, 9203–9212 (2017).
52. Blin, K. *et al.* antiSMASH 2.0—a versatile platform for genome mining of secondary metabolite producers. *Nucleic Acids Research* **41**, W204–W212 (2013).
53. Blin, K. *et al.* antiSMASH 5.0: updates to the secondary metabolite genome mining pipeline. *Nucleic Acids Research* **47**, W81–W87 (2019).
54. Vandova, Gergana; Nivina, Aleksandra; Khosla, Chaitan; Davis, Ronald; Fisher, Curt; Hillenmeyer, M. Identification of polyketide biosynthetic gene clusters that harbor self-resistance target genes. *BioRxiv* (2020) doi:10.1101/2020.06.01.128595.

55. Alanjary, M. *et al.* The Antibiotic Resistant Target Seeker (ARTS), an exploration engine for antibiotic cluster prioritization and novel drug target discovery. *Nucleic Acids Research* **45**, W42–W48 (2017).
56. Blin, K. *et al.* antiSMASH 6.0: improving cluster detection and comparison capabilities. *Nucleic Acids Research* **49**, W29–W35 (2021).
57. Weber, T. *et al.* antiSMASH 3.0—a comprehensive resource for the genome mining of biosynthetic gene clusters. *Nucleic Acids Research* **43**, W237–W243 (2015).
58. Blin, K. *et al.* antiSMASH 4.0—improvements in chemistry prediction and gene cluster boundary identification. *Nucleic Acids Research* **45**, W36–W41 (2017).
59. Mistry, J. *et al.* Pfam: The protein families database in 2021. *Nucleic Acids Research* **49**, D412–D419 (2021).
60. Merwin, N. J. *et al.* DeepRiPP integrates multiomics data to automate discovery of novel ribosomally synthesized natural products. *Proceedings of the National Academy of Sciences* **117**, 371–380 (2020).
61. Sorokina, M. & Steinbeck, C. Review on natural products databases: where to find data in 2020. *Journal of Cheminformatics* **12**, 20 (2020).
62. Xie, T. *et al.* Review of natural product databases. *Cell Proliferation* **48**, 398–404 (2015).
63. van Santen, J. A. *et al.* The Natural Products Atlas 2.0: a database of microbially-derived natural products. *Nucleic Acids Research* **50**, D1317–D1323 (2022).
64. Flissi, A. *et al.* Norine: update of the nonribosomal peptide resource. *Nucleic Acids Research* **48**, (2019).
65. Kautsar, S. A. *et al.* MIBiG 2.0: a repository for biosynthetic gene clusters of known function. *Nucleic Acids Research* **48**, D454–D458 (2019).
66. Blin, K., Shaw, S., Kautsar, S. A., Medema, M. H. & Weber, T. The antiSMASH database version 3: increased taxonomic coverage and new query features for modular enzymes. *Nucleic Acids Research* **49**, D639–D643 (2021).
67. Andriy Burkov. *The Hundred-Page Machine Learning Book*. vol. 1 (2019).
68. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
69. Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J. & Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences* **116**, 8852–8858 (2019).
70. Eddy, S. R. What is a hidden Markov model? *Nature Biotechnology* **22**, 1315–1316 (2004).
71. Yoon, B.-J. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Current Genomics* **10**, 402–415 (2009).
72. Allen, J. F. Natural Language Processing. *John Wiley and Sons Ltd.* 1218–1222 (2003) doi:10.5555/1074100.1074630.
73. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* (2013).
74. Asgari, E. & Mofrad, M. R. K. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLOS ONE* **10**, e0141287 (2015).
75. Haibo He, Yang Bai, Garcia, E. A. & Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* 1322–1328 (IEEE, 2008). doi:10.1109/IJCNN.2008.4633969.

76. Chawla, N. v., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **16**, 321–357 (2002).
77. Quinlan, J. R. Induction of decision trees. *Machine Learning* **1**, 81–106 (1986).
78. Wyner, A. J., Olson, M., Bleich, J. & Mease, D. Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers. *Journal of Machine Learning Research* **18**, 1–33 (2017).
79. Breiman, L. Random Forest. *Machine Learning* **45**, 5–32 (2001).
80. Gracia, A., González, S., Robles, V. & Menasalvas, E. A methodology to compare Dimensionality Reduction algorithms in terms of loss of quality. *Information Sciences* **270**, 1–27 (2014).
81. Duan, M., Fan, J., Li, M., Han, L. & Huo, S. Evaluation of Dimensionality-Reduction Methods from Peptide Folding–Unfolding Simulations. *Journal of Chemical Theory and Computation* **9**, 2490–2497 (2013).
82. Sokolova, M. & Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* **45**, 427–437 (2009).
83. van der Maaten, Laurens; Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9**, 2579–2605 (2008).
84. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* **3**, 861 (2018).
85. van der Maaten, L., Postma, E. & Jaap, V. den H. Dimensionality Reduction: A Comparative Review. *Tilburg centre for Creative Computing* **2009**, 1–36 (2009).
86. Palacio-Niño, J.-O. & Berzal, F. Evaluation Metrics for Unsupervised Learning Algorithms. *arxiv* (2019).
87. Cross, G. R. & Jain, A. K. MEASUREMENT OF CLUSTERING TENDENCY. in *Theory and Application of Digital Control* vol. 15 315–320 (Elsevier, 1982).
88. Sokolova, M. & Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* **45**, 427–437 (2009).
89. Gupta, A. *et al.* Class-Weighted Evaluation Metrics for Imbalanced Data Classification. *arXiv* (2020).
90. Chicco, D. & Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* **21**, 6 (2020).
91. Sim, J. & Wright, C. C. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy* **85**, 257–268 (2005).
92. Bateman, A. *et al.* UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research* **49**, D480–D489 (2021).
93. Benson, D. A. GenBank. *Nucleic Acids Research* **33**, D34–D38 (2004).
94. Pedregosa F.; Varoquaux, G. ; G. A. ; M. V. ; T. B. ; G. O. ; B. M. ; P. P. ; W. R. ; D. V. ; V. J. ; P. A. ; C. D. ; B. M. ; P. M. D. E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).
95. Meyer, J. G., Liu, S., Miller, I. J., Coon, J. J. & Gitter, A. Learning Drug Functions from Chemical Structures with Convolutional Neural Networks and Random Forests. *Journal of Chemical Information and Modeling* **59**, 4438–4449 (2019).
96. Sayers, E. W. *et al.* Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research* **44**, D7–D19 (2016).

97. Dorrity, M. W., Saunders, L. M., Queitsch, C., Fields, S. & Trapnell, C. Dimensionality reduction by UMAP to visualize physical and genetic interactions. *Nature Communications* **11**, 1537 (2020).
98. Wang, W. & Carreira-Perpiñán, M. Á. The role of dimensionality reduction in linear classification. *arXiv* (2014).
99. Banerjee, A. & Dave, R. N. Validating clusters using the Hopkins statistic. in *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)* 149–153 (IEEE, 2004). doi:10.1109/FUZZY.2004.1375706.
100. Rantala, A. *et al.* Phylogenetic evidence for the early evolution of microcystin synthesis. *Proceedings of the National Academy of Sciences* **101**, 568–573 (2004).
101. Du, J. *et al.* Gene2vec: distributed representation of genes based on co-expression. *BMC Genomics* **20**, 82 (2019).
102. Gligorijević, V. *et al.* Structure-based protein function prediction using graph convolutional networks. *Nature Communications* **12**, 3168 (2021).
103. Kobak, D. & Berens, P. The art of using t-SNE for single-cell transcriptomics. *Nature Communications* **10**, 5416 (2019).

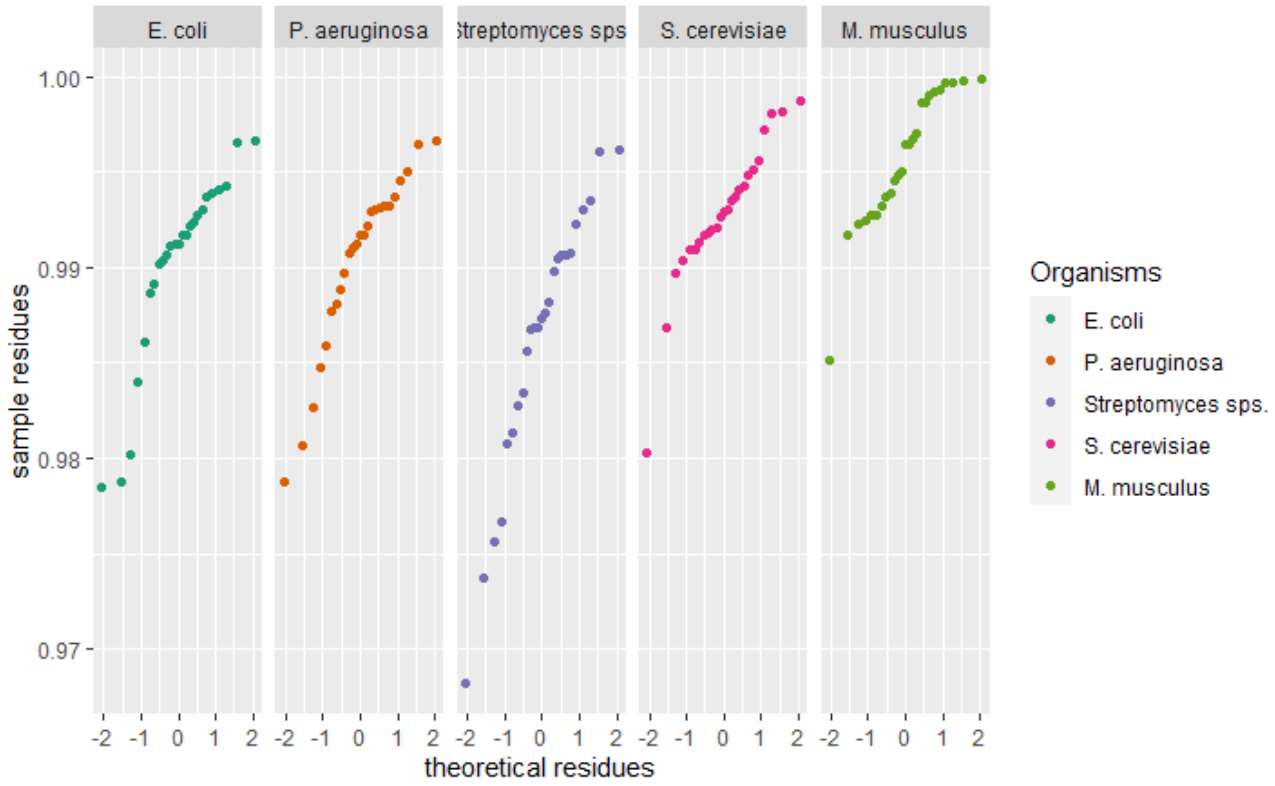
Supplementary Information

Supplementary Table 1: List of housekeeping proteins used to calculate cosine similarity scores with the human homologue. Cosine similarity was calculated between human housekeeping proteins and their homologous sequences from both prokaryotes and eukaryotes.

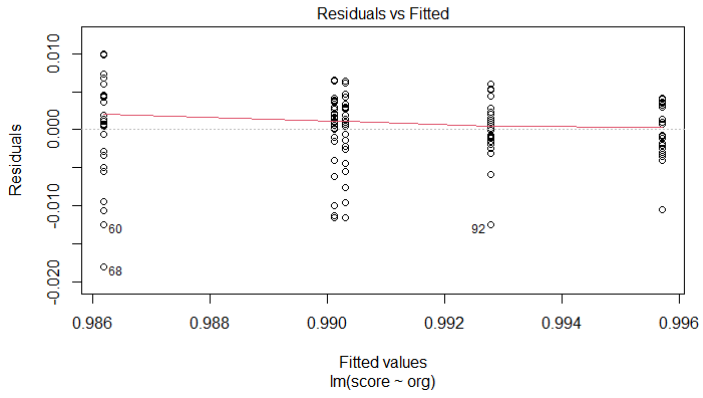
Hexokinase/Glucokinase, Pyruvate kinase, Glycerol Kinase, Glycerol phosphatase, Glutamate dehydrogenase, Aspartate transaminase, Lipoyl synthase, Fumarate dehydrogenase, Threonine synthase, Fumarate hydratase, DNA Polymerase, RNA Polymerase, Ise-tRNA ligase, Trp-tRNA ligase, Ala-tRNA ligase, Glu-tRNA ligase, Gly-tRNA ligase, Tryptophanase, Ser-tRNA ligase, Thr-tRNA ligase, Lysozyme, Fatty acid synthase, Glyceraldehyde-3-phosphate dehydrogenase

Supplementary Table 2: Normality test using a One-way ANOVA. An ANOVA model was fitted using cosine similarity values calculated between housekeeping proteins between each organism and their human homologues. A Shapiro-Wilk test was performed to determine if the residues (cosine-similarity) within each category were normally distributed at a 95% confidence interval

Organism	**Statistic (W)**	**p-value**
E. coli	0.853	0.00204
P. aeruginosa	0.912	0.0330
Streptomyces sps.	0.936	0.118
S. cerevisiae	0.901	0.0190
M. musculus	0.896	0.0152



Supplementary Figure 2: Residual Quantile-Quantile plot of cosine-similarity values of a One-way ANOVA. Cosine similarity was calculated using the vector representation of housekeeping proteins and their homologues present within prokaryotic and eukaryotic organisms. The analysis revealed that several categories violated the normality assumption except for the *Streptomyces sps.* at a 95% confidence interval as described in Table 1.



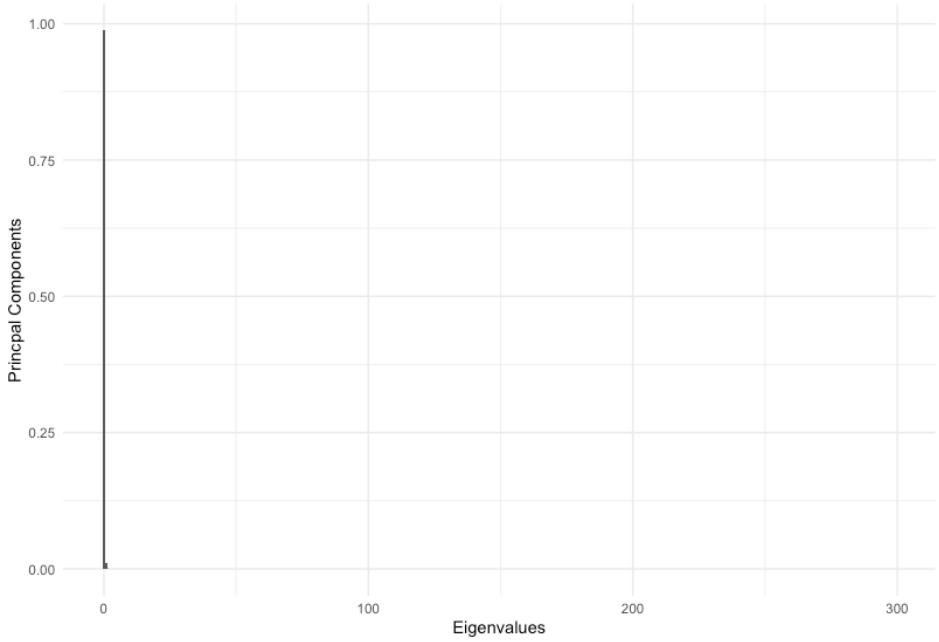
Supplementary Figure 3: Illustration of Homogeneity of Variance of a One-way ANOVA**.** Using the cosine-similarity values as described in Table 1, an ANOVA model was fitted, and its residues were extracted for visualization. Using Levene's test with a 95% confidence interval, it was determined there was no violation of homoscedasticity, with a p-value > 0.05.

Group 1	Group 2	Null Value	Estimate	Low CI	High CI	P-value adjusted	p-value significance
E. coli	*P. aeruginosa*	0	0.000193	-3.70e-3	4.09e-3	1	NS
E. coli	*Streptomyces sps.*	0	-0.00392	-7.81e-3	-2.18e-5	4.8 e-2	*
E. coli	*S. cerevisiae*	0	0.00268	-1.22e-3	6.57e-3	3.22e-1	NS
E. coli	*M. musculus*	0	0.00560	1.70e-3	9.49e-3	1.1 e-3	**
P. aeruginosa	*Streptomyces sps.*	0	-0.00411	-8.01e-3	-2.14e-4	3.32e-2	*
P. aeruginosa	*S. cerevisiae*	0	0.00248	-1.41e-3	6.38e-3	3.99e-1	NS
P. aeruginosa	*M. musculus*	0	0.00541	1.51e-3	9.30e-3	1.8 e-3	**
Streptomyces sps.	*S. cerevisiae*	0	0.00659	2.70e-3	1.05e-2	7.16e-5	****
Streptomyces sps.	*M. musculus*	0	0.00952	5.62e-3	1.34e-2	5.14e-9	****
S. cerevisiae	*M. musculus*	0	0.00292	-9.72e-4	6.82e-3	2.36e-1	NS

Supplementary Table 3: Multiple-comparison adjustment using Tukey Honest Significant Difference (HSD) shows statistical differences between proteins of different life domains. Multiple comparisons were corrected following an ANOVA, by using a Tukey HSD test was performed with a confidence level of 95% to determine significant differences.

Supplementary Table 5: List of Biosynthetic Gene Clusters used for phylogenetic analysis. A subset of BGCs was extracted from the MIBiG database to determine how the BGCs form clades.

Polyketide Synthase	**Non-ribosomal peptide synthase**	**RiPPs**	**Terpene Synthase**
BGC0000038	BGC0000303	BGC0000479	BGC0000636
BGC0000002	BGC0000295	BGC0000288	BGC0000646
BGC0000029	BGC0000299	BGC0000496	BGC0000633
BGC0000028	BGC0000294	BGC0000513	BGC0000649
BGC0000034	BGC0000324	BGC0000501	BGC0000648
BGC0000014	BGC0000495	BGC0000290	BGC0000634
BGC0000023	BGC0000316	BGC0000505	BGC0000639
BGC0000017	BGC0000291	BGC0000502	BGC0000641
BGC0000032	BGC0000311	BGC0000507	BGC0000640
BGC0000024	BGC0000296	BGC0000509	BGC0000638
BGC0000031	BGC0000310	BGC0000510	BGC0000642
BGC0000019	BGC0000301	BGC0000499	BGC0000630
BGC0000020	BGC0000302	BGC0000494	BGC0000635
BGC0000025	BGC0000298	BGC0000498	BGC0000643
BGC0000033	BGC0000308	BGC0000506	BGC0000637
BGC0000036	BGC0000300	BGC0000511	BGC0000647
BGC0000018	BGC0000309	BGC0000497	BGC0000644
BGC0000021	BGC0000287	BGC0000500	BGC0000651
BGC0000035	BGC0000319	BGC0000504	BGC0000645
BGC0000001		BGC0000508	BGC0000650
		BGC0000503	



Supplementary Figure 4 : Scree plot displaying the number of principal components which describe the majority of variance within the MIBiG database protein-vector space.

Supplementary Table 4: Classification metrics used to evaluate Random Forest when evaluating using the MIBiG database.

Predict	**0.0**	**1.0**	**3.0**	**4.0**
Actual				
0.0	**79**	**0**	**2**	**0**
1.0	**3**	**125**	**4**	**0**
3.0	**0**	**1**	**19**	**5**
4.0	**0**	**0**	**2**	**39**
Overall Statistics:				
95% CI	**(0.911,0.96714)**			
ACC Macro	**0.96953**			
ARI	**0.88487**			
AUNP	**0.96277**			
AUNU	**0.94488**			
Bangdiwala B	**0.92214**			
Bennett S	**0.91876**			
CBA	**0.87511**			
CSI	**0.79476**			
Chi-Squared	**639.14151**			
Chi-Squared DF	**9**			
Conditional Entropy	**0.34128**			
Cramer V	**0.87385**			
Cross-Entropy	**1.74895**			
F1 Macro	**0.89668**			
F1 Micro	**0.93907**			
FNR Macro	**0.09163**			
FNR Micro	**0.06093**			
FPR Macro	**0.01861**			
FPR Micro	**0.02031**			
Gwet AC1	**0.92158**			
Hamming Loss	**0.06093**			
Joint Entropy	**2.08854**			
KL Divergence	**0.00169**			

Kappa	**0.90894**
Kappa 95% CI	**(0.867,0.95089)**
Kappa No Prevalence	**0.87814**
Kappa Standard Error	**0.0214**
Kappa Unbiased	**0.90892**
Krippendorff Alpha	**0.90908**
Lambda A	**0.88435**
Lambda B	**0.88889**
Mutual Information	**1.44215**
NIR	**0.47312**
Overall ACC	**0.93907**
Overall CEN	**0.11701**
Overall J	**(3.30391,0.82598)**
Overall MCC	**0.90942**
Overall MCEN	**0.18423**
Overall RACC	**0.33084**
Overall RACCU	**0.331**
P-Value	**0.0**
PPV Macro	**0.88639**
PPV Micro	**0.93907**
Pearson C	**0.83434**
Phi-Squared	**2.29083**
RCI	**0.82538**
RR	**69.75**
Reference Entropy	**1.74726**
Response Entropy	**1.78343**
SOA1(Landis & Koch)	**Almost Perfect**
SOA2(Fleiss)	**Excellent**
SOA3(Altman)	**Very Good**
SOA4(Cicchetti)	**Excellent**
SOA5(Cramer)	**Very Strong**
SOA6(Matthews)	**Very Strong**
Scott PI	**0.90892**
Standard Error	**0.01432**
TNR Macro	**0.98139**
TNR Micro	**0.97969**
TPR Macro	**0.90837**
TPR Micro	**0.93907**
Zero-one Loss	**17**
Class Statistics:	

Classes	**0.0**	**1.0**	**3.0**	**4.0**
ACC (Accuracy)	**0.98208**	**0.97133**	**0.94982**	**0.97491**
AGF Adjusted F-score)	**0.98084**	**0.95872**	**0.85385**	**0.96289**
AGM Adjusted geometric mean)	**0.98205**	**0.97788**	**0.91063**	**0.97144**
AM (Difference between automatic and manual classification)	**1**	**-6**	**2**	**3**
AUC (Area under the ROC curve)	**0.98008**	**0.97008**	**0.86425**	**0.96511**
AUCI AUC value interpretation)	**Excellent**	**Excellent**	**Very Good**	**Excellent**
AUPR Area under the PR curve)	**0.96936**	**0.96952**	**0.73185**	**0.91879**
BCD (Bray-Curtis dissimilarity)	**0.00179**	**0.01075**	**0.00358**	**0.00538**
BM (Informedness or bookmaker informedness)	**0.96016**	**0.94017**	**0.7285**	**0.93021**
CEN (Confusion entropy)	**0.07117**	**0.07697**	**0.41807**	**0.14225**
DOR (Diagnostic odds ratio)	**2567.5**	**2607.14286**	**97.375**	**908.7**
DP (Discriminant power)	**1.87976**	**1.88343**	**1.09629**	**1.63106**
DPI Discriminant power interpretation)	**Limited**	**Limited**	**Limited**	**Limited**
ERR (Error rate)	**0.01792**	**0.02867**	**0.05018**	**0.02509**
F0.5 (F0.5 score)	**0.96577**	**0.9827**	**0.71429**	**0.89862**
F1(F1 score - harmonic mean of precision and sensitivity)	**0.96933**	**0.96899**	**0.73077**	**0.91765**
F2 (F2 score)	**0.97291**	**0.95566**	**0.74803**	**0.9375**
FDR (False discovery rate)	**0.03659**	**0.00794**	**0.2963**	**0.11364**
FN (False negative/miss/type 2 error)	**2**	**7**	**6**	**2**
FNR (Miss rate or false negative rate)	**0.02469**	**0.05303**	**0.24**	**0.04878**
FOR False omission rate)	**0.01015**	**0.04575**	**0.02381**	**0.00851**
FP False positive/type 1 error/false alarm)	**3**	**1**	**8**	**5**
FPR (Fall-out or false positive rate)	**0.01515**	**0.0068**	**0.0315**	**0.02101**
G (G-measure geometric mean of precision and sensitivity)	**0.96934**	**0.96925**	**0.73131**	**0.91822**
GI (Gini index)	**0.96016**	**0.94017**	**0.7285**	**0.93021**
GM(G-mean geometric mean of specificity and sensitivity)	**0.98007**	**0.96981**	**0.85794**	**0.96501**
IBA(Index of balanced accuracy)	**0.95137**	**0.89705**	**0.58259**	**0.90537**
ICSI(Individual classification success index)	**0.93872**	**0.93903**	**0.4637**	**0.83758**
IS(Information score)	**1.7305**	**1.06823**	**2.97331**	**2.59254**
J(Jaccard index)	**0.94048**	**0.93985**	**0.57576**	**0.84783**
LS(Lift score)	**3.31843**	**2.09686**	**7.85333**	**6.0316**
MCC(Matthews correlation coefficient)	**0.9567**	**0.94323**	**0.70378**	**0.90365**
MCCI(Matthews correlation coefficient interpretation)	**Very Strong**	**Very Strong**	**Strong**	**Very Strong**
MCEN(Modified confusion entropy)	**0.11609**	**0.12707**	**0.55111**	**0.21071**
MK(Markedness)	**0.95326**	**0.94631**	**0.67989**	**0.87785**
N(Condition negative)	**198**	**147**	**254**	**238**
NLR(Negative likelihood ratio)	**0.02507**	**0.05339**	**0.2478**	**0.04983**
NLRI(Negative likelihood ratio interpretation)	**Good**	**Good**	**Poor**	**Good**
NPV(Negative predictive value)	**0.98985**	**0.95425**	**0.97619**	**0.99149**

OC(Overlap coefficient)	**0.97531**	**0.99206**	**0.76**	**0.95122**
OOC(Otsuka-Ochiai coefficient)	**0.96934**	**0.96925**	**0.73131**	**0.91822**
OP(Optimized precision)	**0.97721**	**0.9475**	**0.82919**	**0.96052**
P(Condition positive or support)	**81**	**132**	**25**	**41**
PLR(Positive likelihood ratio)	**64.37037**	**139.20455**	**24.13**	**45.27805**
PLRI(Positive likelihood ratio interpretation)	**Good**	**Good**	**Good**	**Good**
POP(Population)	**279**	**279**	**279**	**279**
PPV(Precision or positive predictive value)	**0.96341**	**0.99206**	**0.7037**	**0.88636**
PRE(Prevalence)	**0.29032**	**0.47312**	**0.08961**	**0.14695**
Q(Yule Q - coefficient of colligation)	**0.99922**	**0.99923**	**0.97967**	**0.9978**
QI(Yule Q interpretation)	**Strong**	**Strong**	**Strong**	**Strong**
RACC(Random accuracy)	**0.08533**	**0.21367**	**0.00867**	**0.02318**
RACCU(Random accuracy unbiased)	**0.08533**	**0.21378**	**0.00868**	**0.0232**
TN(True negative/correct rejection)	**195**	**146**	**246**	**233**
TNR(Specificity or true negative rate)	**0.98485**	**0.9932**	**0.9685**	**0.97899**
TON(Test outcome negative)	**197**	**153**	**252**	**235**
TOP(Test outcome positive)	**82**	**126**	**27**	**44**
TP(True positive/hit)	**79**	**125**	**19**	**39**
TPR(Sensitivity, recall, hit rate, or true positive rate)	**0.97531**	**0.94697**	**0.76**	**0.95122**
Y(Youden index)	**0.96016**	**0.94017**	**0.7285**	**0.93021**
dInd(Distance index)	**0.02897**	**0.05346**	**0.24206**	**0.05311**
sInd(Similarity index)				

Supplementary Table 5: Classification statistics extracted when evaluating the fitted Random Forest with the antiSMASH database.

Actual				
0.0	**3788**	**228**	**198**	**307**
1.0	**228**	**5638**	**278**	**497**
3.0	**256**	**295**	**5852**	**481**
4.0	**516**	**618**	**597**	**11078**
Overall Statistics :				
95% CI				**(0.85025,0.85813)**
ACC Macro				**0.92709**
ARI				**0.65174**
AUNP				**0.89947**
AUNU				**0.90009**
Bangdiwala B				**0.7448**
Bennett S				**0.80559**
CBA				**0.83319**
CSI				**0.69143**
Chi-Squared				**58395.40725**
Chi-Squared DF				**9**
Conditional Entropy				**0.82436**
Cramer V				**0.79427**
Cross-Entropy				**1.89313**
F1 Macro				**0.84546**
F1 Micro				**0.85419**
FNR Macro				**0.14955**
FNR Micro				**0.14581**
FPR Macro				**0.05027**
FPR Micro				**0.0486**
Gwet AC1				**0.80869**
Hamming Loss				**0.14581**
Joint Entropy				**2.71669**
KL Divergence				**0.0008**
Kappa				**0.79566**

Kappa 95% CI	**(0.79014,0.80118)**
Kappa No Prevalence	**0.70838**
Kappa Standard Error	**0.00282**
Kappa Unbiased	**0.79564**
Krippendorff Alpha	**0.79564**
Lambda A	**0.75069**
Lambda B	**0.75671**
Mutual Information	**1.0856**
NIR	**0.41514**
Overall ACC	**0.85419**
Overall CEN	**0.29688**
Overall J	**(2.93206,0.73302)**
Overall MCC	**0.79584**
Overall MCEN	**0.43953**
Overall RACC	**0.28644**
Overall RACCU	**0.28651**
P-Value	**None**
PPV Macro	**0.84099**
PPV Micro	**0.85419**
Pearson C	**0.80888**
Phi-Squared	**1.89258**
RCI	**0.57368**
RR	**7713.75**
Reference Entropy	**1.89233**
Response Entropy	**1.90996**
SOA1(Landis & Koch)	**Substantial**
SOA2(Fleiss)	**Excellent**
SOA3(Altman)	**Good**
SOA4(Cicchetti)	**Excellent**
SOA5(Cramer)	**Strong**
SOA6(Matthews)	**Strong**
Scott PI	**0.79564**
Standard Error	**0.00201**
TNR Macro	**0.94973**
TNR Micro	**0.9514**
TPR Macro	**0.85045**
TPR Micro	**0.85419**
Zero-one Loss	**4499**
Class Statistics :	

Classes	**0.0**	**1.0**	**3.0**	**4.0**
ACC(Accuracy)	**0.94383**	**0.93051**	**0.93178**	**0.90225**
AGF(Adjusted F-score)	**0.89619**	**0.89961**	**0.90121**	**0.89063**
AGM(Adjusted geometric mean)	**0.92738**	**0.92293**	**0.92479**	**0.90826**
AM(Difference between automatic and manual classification)	**267**	**138**	**41**	**-446**
AUC(Area under the ROC curve)	**0.89995**	**0.90092**	**0.90266**	**0.89683**
AUCI(AUC value interpretation)	**Very Good**	**Excellent**	**Excellent**	**Very Good**
AUPR(Area under the PR curve)	**0.81451**	**0.84033**	**0.84757**	**0.88046**
BCD(Bray-Curtis dissimilarity)	**0.00433**	**0.00224**	**0.00066**	**0.00723**
BM(Informedness or bookmaker informedness)	**0.79989**	**0.80185**	**0.80532**	**0.79365**
CEN(Confusion entropy)	**0.35456**	**0.31617**	**0.30602**	**0.26025**
DOR(Diagnostic odds ratio)	**130.92114**	**113.66914**	**121.01033**	**83.4759**
DP(Discriminant power)	**1.16717**	**1.13334**	**1.14832**	**1.05941**
DPI(Discriminant power interpretation)	**Limited**	**Limited**	**Limited**	**Limited**
ERR(Error rate)	**0.05617**	**0.06949**	**0.06822**	**0.09775**
F0.5(F0.5 score)	**0.80007**	**0.83509**	**0.84606**	**0.88964**
F1(F1 score - harmonic mean of precision and sensitivity)	**0.81384**	**0.84024**	**0.84756**	**0.88018**
F2(F2 score)	**0.82809**	**0.84545**	**0.84908**	**0.87093**
FDR(False discovery rate)	**0.20886**	**0.16831**	**0.15495**	**0.10394**
FN(False negative/miss/type 2 error)	**733**	**1003**	**1032**	**1731**
FNR(Miss rate or false negative rate)	**0.16213**	**0.15103**	**0.14991**	**0.13514**
FOR(False omission rate)	**0.02812**	**0.04166**	**0.04313**	**0.09361**
FP(False positive/type 1 error/false alarm)	**1000**	**1141**	**1073**	**1285**
FPR(Fall-out or false positive rate)	**0.03797**	**0.04712**	**0.04476**	**0.07121**
G(G-measure geometric mean of precision and sensitivity)	**0.81417**	**0.84028**	**0.84757**	**0.88032**
GI(Gini index)	**0.79989**	**0.80185**	**0.80532**	**0.79365**
GM(G-mean geometric mean of specificity and sensitivity)	**0.8978**	**0.89942**	**0.90113**	**0.89626**
IBA(Index of balanced accuracy)	**0.70597**	**0.7249**	**0.72665**	**0.75192**
ICSI(Individual classification success index)	**0.62901**	**0.68065**	**0.69514**	**0.76092**
IS(Information score)	**2.4328**	**1.95014**	**1.9213**	**1.11002**
J(Jaccard index)	**0.68611**	**0.72449**	**0.73545**	**0.78601**
LS(Lift score)	**5.39942**	**3.86413**	**3.78764**	**2.15848**
MCC(Matthews correlation coefficient)	**0.78124**	**0.79591**	**0.80362**	**0.79804**
MCCI(Matthews correlation coefficient interpretation)	**Strong**	**Strong**	**Strong**	**Strong**
MCEN(Modified confusion entropy)	**0.5063**	**0.46145**	**0.4497**	**0.39553**
MK(Markedness)	**0.76302**	**0.79003**	**0.80193**	**0.80245**
N(Condition negative)	**26334**	**24214**	**23971**	**18046**
NLR(Negative likelihood ratio)	**0.16853**	**0.1585**	**0.15694**	**0.1455**
NLRI(Negative likelihood ratio interpretation)	**Fair**	**Fair**	**Fair**	**Fair**
NPV(Negative predictive value)	**0.97188**	**0.95834**	**0.95687**	**0.90639**
OC(Overlap coefficient)	**0.83787**	**0.84897**	**0.85009**	**0.89606**

OOC(Otsuka-Ochiai coefficient)	**0.81417**	**0.84028**	**0.84757**	**0.88032**
OP(Optimized precision)	**0.87485**	**0.87285**	**0.87353**	**0.86661**
P(Condition positive or support)	**4521**	**6641**	**6884**	**12809**
PLR(Positive likelihood ratio)	**22.06441**	**18.01659**	**18.99109**	**12.14574**
PLRI(Positive likelihood ratio interpretation)	**Good**	**Good**	**Good**	**Good**
POP(Population)	**30855**	**30855**	**30855**	**30855**
PPV(Precision or positive predictive value)	**0.79114**	**0.83169**	**0.84505**	**0.89606**
PRE(Prevalence)	**0.14652**	**0.21523**	**0.22311**	**0.41514**
Q(Yule Q - coefficient of colligation)	**0.98484**	**0.98256**	**0.98361**	**0.97632**
QI(Yule Q interpretation)	**Strong**	**Strong**	**Strong**	**Strong**
RACC(Random accuracy)	**0.02274**	**0.04729**	**0.05007**	**0.16634**
RACCU(Random accuracy unbiased)	**0.02276**	**0.04729**	**0.05007**	**0.16639**
TN(True negative/correct rejection)	**25334**	**23073**	**22898**	**16761**
TNR(Specificity or true negative rate)	**0.96203**	**0.95288**	**0.95524**	**0.92879**
TON(Test outcome negative)	**26067**	**24076**	**23930**	**18492**
TOP(Test outcome positive)	**4788**	**6779**	**6925**	**12363**
TP(True positive/hit)	**3788**	**5638**	**5852**	**11078**
TPR(Sensitivity, recall, hit rate, or true positive rate)	**0.83787**	**0.84897**	**0.85009**	**0.86486**
Y(Youden index)	**0.79989**	**0.80185**	**0.80532**	**0.79365**
dInd(Distance index)	**0.16652**	**0.15821**	**0.15645**	**0.15275**
sInd(Similarity index)	**0.88225**	**0.88813**	**0.88937**	**0.89199**