

Computational Design of Novel Non-Ribosomal Peptides

Sherif Farag

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Curriculum in Bioinformatics and Computational Biology.

Chapel Hill
2019

Approved by:

Alexander Tropsha

Albert A. Bowers

Elizabeth A. Shank

Shawn Gomez

Timothy Elston

©2019
Sherif Farag
ALL RIGHTS RESERVED

ABSTRACT

Sherif Farag: Computational Design of Novel Non-Ribosomal Peptides
(Under the direction of Alexander Tropsha)

Non-ribosomal peptide synthetases (NRPSs) are modular enzymatic machines that catalyze the ribosome-independent production of structurally complex small peptides, many of which have important clinical applications as antibiotics, antifungals, and anti-cancer agents. Several groups have tried to expand natural product diversity by intermixing different NRPS modules to create synthetic peptides. This approach has not been as successful as anticipated, suggesting that these modules are not fully interchangeable.

Here, we explored whether inter-modular linkers (IMLs) impact the ability of NRPS modules to communicate during the synthesis of NRPs. We developed a parser to extract 39,804 IMLs from both well annotated and putative NRPS biosynthetic gene clusters from 39,232 bacterial genomes and established the first IMLs database. We analyzed these IMLs and identified a striking relationship between IMLs and the amino acid substrates of their adjacent modules. More than 92% of the identified IMLs connect modules that activate a particular pair of substrates, suggesting that significant specificity is embedded within these sequences. We therefore propose that incorporating the correct IML is critical when attempting combinatorial biosynthesis of novel NRPS.

In addition to the IMLs database and IML-Parser we have developed the NRP Discovery Pipeline, which is a set of bioinformatics and cheminformatics tools that will help facilitating early discovery of novel NRPs. Our pipeline comprises of five modules: (1) NRP comprehensive combinatorial biosynthesis: A tool that helps generating virtual libraries of NRPs. (2) NRP sequence-based predictor: A classifier based only on peptide sequences to help triaging peptides with no anti-bacterial activity. (3) Pep2struc: A tool that helps converting peptide sequences to their 2D structures form both linear and constrained peptides. (4) NRP structure-based predictor: A second classifier based on peptide structures to filter out inactive predicted peptides. (5) NRPS Designer: A tool that

helps reprogramming of the bacterial genome by editing its NRP BGC to synthesize the peptide of interest.

The IMLs database as well as the NRPS-Parser have been made available on the web at <https://nrps-linker.unc.edu>. The entire source code of the projects discussed in this dissertation is hosted in GitHub repository (<https://github.com/SWFarag>).

This dissertation is dedicated to my family, especially my parents, whose unwavering love and support throughout my life have encouraged me to dream and made this research possible.

ACKNOWLEDGEMENTS

I would like to express my gratitude to all those who took part in the completion of this dissertation. First and foremost, I would like to thank my advisor Alexander Tropsha for giving me the great opportunity to pursue my PhD at the Molecular Modeling Lab at the UNC Eshelman School of Pharmacy. I highly appreciate our fruitful discussion, his ability to offer guidance and at the same time encourage independent thinking and the constant support to ensure that I take the most out of my graduate school experience. Similarly, I would also like to thank the other members of my committee Albert Bowers, Elizabeth Shank, Tim Elston, and Shawn Gomez, who have helped me with valuable insights and suggestions over the years. They have been a crucial factor towards the completion of this work.

The unique work environment at the Molecular Modeling Lab fosters great productivity and work motivation. I was happy to be part of it and for that I can only be thankful to all Tropsha's lab members, especially Stephen Capuzzi, Vinicius Alves and Olexandr Isayev with whom I collaborated on many research projects and also shared many wonderful memorable moments over the years. I would also like to thank all my fellow graduate students with whom I have shared great scientific discussions, research struggles and many laughs. Additionally, I would like to thank all my other friends for their constant support, great company and for believing in me and pushing me forward.

Finally, I would like to thank my family and my wife Iva for patiently sharing in my joys and pains and for their unwavering love and dedication, which have encouraged me to dream and made this research possible.

TABLE OF CONTENTS

LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xviii
1 Introduction	1
1.1 Antibiotic resistant crisis	1
1.2 Managing the antibiotic resistant crisis	2
1.3 Non-Ribosomal Peptides	4
1.4 Combinatorial biosynthesis of novel NRPs	5
1.5 IDLs vs. IMLs	5
1.6 Objectives	6
2 Inter-Modular Linkers play a crucial role in governing the biosynthesis of non- ribosomal peptides	7
2.1 Introduction.....	7
2.2 Materials and Methods	9
2.2.1 Study design and dataset	9
2.2.2 IML NRPS-Parser	9
2.2.3 Web-Server	11
2.3 Results	11
2.3.1 IML extraction	11
2.3.2 Analysis of IMLs	11
2.3.2.1 Selectivity of unique IMLs toward pairs of modules	12

2.3.2.2	Phylogenetic conservation of module-specific IMLs within and across genera	13
2.3.2.3	IMLs as independent building blocks	16
2.4	Discussion	17
2.5	Conclusion	22
2.6	Supplementary	24
3	NRP Discovery Pipeline	36
3.1	Introduction	36
3.1.1	Pipeline Overview	37
3.2	Materials and Methods	38
3.2.1	NRP Comprehensive Combinatorial Biosynthesis (CCB)	38
3.2.1.1	Objective	38
3.2.1.2	CCB algorithm	39
3.2.1.3	CCB distinct runs	40
3.2.1.4	Code availability	40
3.2.1.5	Notes	40
3.2.2	Sequence Based Model	42
3.2.2.1	Objective	42
3.2.2.2	Modeling Strategy	42
3.2.2.3	Data set	43
3.2.2.4	End Point	43
3.2.2.5	Data curation	43
3.2.2.6	Descriptors	44
3.2.2.7	Modeling approaches	44
3.2.2.8	Code availability	46
3.2.2.9	Notes	46
3.2.3	Pep2Struc	47
3.2.3.1	Objective	47

3.2.3.2	Pep2Struc algorithm	52
3.2.3.3	Pep2Struc VS CycloPs	55
3.2.3.4	Code availability	56
3.2.3.5	Notes	57
3.2.4	Structure Based Model	58
3.2.4.1	Objective	58
3.2.4.2	Modeling Strategy	58
3.2.4.3	Data set	59
3.2.4.4	End point	59
3.2.4.5	Data curation	59
3.2.4.6	Descriptors	60
3.2.4.7	Modeling approaches	61
3.2.4.8	Code availability	62
3.2.4.9	Notes	62
3.2.5	NRPS Designer	64
3.2.5.1	Objective	64
3.2.5.2	NRPS Designer algorithm	64
3.2.5.3	Code availability	66
3.2.5.4	Notes	66
3.3	Results	68
3.3.1	Model evaluation	68
3.3.1.1	Five fold cross validation(5-FCV)	68
3.3.2	NRPS Comprehensive Combinatorial Biosynthesis (NRPS-CCB)	69
3.3.3	Sequence Based Model	69
3.3.3.1	Model evaluation	69
3.3.3.2	Virtual screening	71
3.3.3.3	Reducing the number of hit list	73
3.3.4	Pep2Struc: Peptide to Structure	74

3.3.5	Structure Based Model.....	74
3.3.5.1	Model evaluation	74
3.3.5.2	Virtual screening	75
3.3.5.3	Reducing the number of hit list	76
3.3.6	NRPS Designer	78
3.4	Discussion	80
3.5	Conclusion	81
3.6	Supplementary	82
4	Conclusion and Future Directions	90
	BIBLIOGRAPHY	93

LIST OF TABLES

S1	Mycobacterium Abscessus Module-Specific IMLs: Metadata of the community network, showing the size of each community, number of unique and redundant BGCs from which the IMLs were extracted.	29
S2	Data Summary: Overview of data analyzed in this study.	33
3.1	A Subset of Monomers	47
3.2	A subset of the final outcome of the CCB algorithm: A subset demonstrating two peptide examples from each of the four runs.	70

LIST OF FIGURES

- 2.1 **Study design:** (A) Two bacterial genome databases and a BCG repository were processed and integrated: The NCBI prokaryotic RefSeq genomes, ENA Ensembl Bacteria and MIBiG, respectively. (B) In addition to NRPS clusters from the MIBiG repository, antiSMASH 3.0 was run on downloaded genomes to identify all potential NRP BGCs. (C) Our NRPS-Parser tool was applied to extract all possible IMLs. (D) A database of IMLs was established. All identified IMLs were then analyzed for (i) Selectivity and specificity, (ii) Phylogenetic conservation, and (iii) Properties. 10
- 2.2 **Mycobacterium Abscessus Module-Specific IMLs:** Community network, where nodes refer to linkers, and edges are constructed between two linkers, if they share 80% or more sequence similarity. The graph depicts nine distinct communities. Each community represents all the linkers that bind a specific pair of modules. For instance, the orange community refers to all linkers that bind the pair of modules that activate phenylalanine and tyrosine. 14
- 2.3 **(Thr-Val) IMLs community network:** A network of all the linkers that bridge the Thr-Val module pair. These linkers belong to various distinct communities, despite the fact that they are all linking the same pair. The coloring of the graph refers to the species from which linkers were extracted. 17
- 2.4 **Linkers length distribution:** There are three clusters of IMLs based on their lengths: (red) Linkers with lengths ranging between 9-120 amino acids. (green) Linkers with lengths ranging between 160-280 amino acids. (black) These are linkers with more than 300 amino acids in length (outliers). 19
- 2.5 **Retrospective analysis of daptomycin analogues biosynthesis:** (A) Daptomycin BGC from *Streptomyces roseosporus*. (B) NRPS organization of the daptomycin cluster and schematic showing module exchange strategy. Modules 8 and 11 were swapped for each other, or for the Asn 11 module from A54145 biosynthesis from *Streptomyces fradiae*. The swapping resulted in the synthesis of four daptomycin analogues I, II, III, and IV each with 15%, 45%, 19% and 6% yield relative to the wild-type. The lightning bolts signify IML incompatibility. 20
- S1 **Inter-Modular Linker:** The NRPS-Parser extracts linkers in the following pattern “A1-linker-A2”. A1 and A2 refer to the activated substrates of the Adenylation domains from module 1 and module 2, respectively. The linker is the segment of amino acids linking these two successive NRPSs modules. 24
- S2 **GC Context and IML length:** (Left) Distribution of GC content across all species. (Right) Distribution of linker lengths across all species. 25

S3	IMLs Hydrophobicity and Secondary Structure Profiles: (Left) On average linkers were found to be composed of 44% neutral amino acids, 33% polar amino acids and 23% hydrophobic amino acids. (Right) 49% of all secondary structures were α -helices, while strands and coils comprised 22% and 29%, respectively.	26
S4	Linkers pairwise distribution: Based on sequence similarity (green), based on sequence identity (blue) and based on gap ratio (red).	27
S5	IMLs Selectivity: (A) Clustering the 39,804 extracted IMLs using the cluster-fast algorithm from UCLUST lead to 12,174 clusters. The removal of singletons resulted in 3,916 unique IMLs centroids (clusters). Only 8% of all unique IMLs tend to bridge multiple pairs of modules, while the remaining 92% link specifically just to a single pair of modules (B) An example of an IML, extracted from <i>Burkholderia pseudomallei</i> 406e linking five distinct pairs of modules.	28
S6	<i>Burkholderia pseudomallei</i> module-specific IMLs: Community network, where nodes refer to linkers, and edges are constructed between two linkers, if they share 80% or more sequence similarity. The graph depicts seventeen distinct communities. Each community represents all the linkers that bind a specific pair of modules. For instance, the red community refers to all linkers that bind the pair of modules that activate glycine and valine, respectively.	30
S7	Phylogenetic conservation of module-specific IMLs within and across genera: (A) Analysis of all pairwise sequence similarity comparisons of IMLs that link the same pair of modules retrieved from the MIBiG NRP clusters. Only 10% of all comparisons showed sequence similarity of over 80%. (B) Multiple sequence alignment of linkers of the same genus linking the pair of modules (Val-Leu), showing high degree of conservation. (C) Multiple sequence alignment of linkers across different genera linking the same pair of modules (Hpg-Hpg), showing a relatively high degree of conservation. (D) Multiple sequence alignment of linkers of the same genus but different species linking the pair of modules (Leu-Ser), showing low degree of conservation.	31
S8	Pairwise similarities of IMLs linking the same pair of modules: (A) Total pairwise comparisons of linkers (5,827,650) that link the same module pairs retrieved from the 51,810 potential NRP clusters. Only 24% of all comparisons showed sequence similarity of over 80%, while the remaining 76% showed a lesser degree of conservation. (B) A sequence logo of 27 IMLs all linking the same pair of module VAL-ASP, extracted from 3 distinct genera: <i>Bacillus</i> , <i>Brevibacterium</i> and <i>Jeotgalibacillus</i> . (C) A sequence logo of 427 IMLs all linking the same pair of modules GLY-CYS, extracted from 4 distinct genera: <i>Citrobacter</i> , <i>Escherichia</i> , <i>Klebsiella</i> and <i>Enterobacter</i>	32

S9	Retrospective analysis of Ambactin analogues biosynthesis: (A) Ambactin BGC from <i>Xenorhabdus miraniensis</i> . (B) NRPS organization of the Ambactin cluster and schematic showing XU units exchange strategy. I) Phe-specific XU3 was exchanged against an Alaspecific XU from the Kolossin NRPS BGC from <i>Photorhabdus luminescens</i> , resulted in no product. II) Phe-specific XU3 against a Phe-specific XU from the GxpS NRPS, resulted in approximately 88% yield relative to wild-type. III) Phe-specific XU3 and Lys-specific XU4 was exchanged against the two building blocks XU3 and XU2 from GxpS, respectively. This resulted in approximately 57% yield relative to wild-type.	34
S10	Our Dataset vs. MIBiG: (A) Venn diagram showing number of unique bacteria involved in our study (31,338) as compared to number of unique bacteria biosynthesizing NRPs in the MIBiG repository (243), with only 111 bacteria shared between them. (B) Venn diagram showing number of unique NRP cluster-prints identified in this study (7,365) as compared to 266 from the MIBiG repository, with only 132 NRP cluster-prints shared between them.	35
3.1	CCB-peptide validity: (A) A subset of valid IMLs from the IMLs database. (B) An example of two virtually generated peptides, the first one would be considered valid (green) by the CCB algorithm while the second one would be considered invalid (red) due to the lack of an IML between serine and phenylalanine in the IMLs database.	38
3.2	Predictive statistical modeling workflow	42
3.3	Condensation reaction: A chemical reaction between alanine and phenylealanine amino acids, that will result in the expulsion a of a water molecule and formation of a peptide bond (amide)	48
3.4	Esterification reaction: A chemical reaction between 7-methyloctanoic acid and alanine amino acid, that will result in the expulsion a of a water molecule and formation of an ester bond.	48
3.5	Examples of NRPs with different cyclizations: (A) Tyrocidin represents a head to tail cyclization. (B) Daptomycin represents a partial cyclization. (C) Actinomycin represents a double cyclization.	50
3.6	Protection: (A) A condensation reaction between an acid and a base, where all their atoms are unprotected, leading to the creation of two products instead of one [a desired one and an undesired one]. (B) A condensation reaction between an acid and a base, however this time the secondary amine group in the base molecule is protected, which will then lead to the creation of just a single product [desired one].	51

3.7	Pep2struc vs. CycloPs: (A) Two condensation reactions conducted by cyclop, using two different smile annotations for the alanine molecule, that leads to the creation of two distinct products. (B) Two condensation reactions conducted by Pep2struc, using two different smile annotations for the alanine molecule, that leads to the creation of just a single product.	56
3.8	Predictive QSAR modeling workflow.	58
3.9	NRPS-designer: A scheme showing the role and impact of the NRPS-designer tool.	64
3.10	NRPS-designer template based: A scheme showing the usage of tyrocidin BGC as a template and applying couple of edits to its Tyc NRPS enzyme, namely exchanging module 5 on TycC that activates alanine to a module that activate lysine. Moreover, exchanging module 8 which activates valine to a module that activates glycine. In addition to exchanging modules also the corresponding inter-modular linkers are also add to the mix, denoted in red lines.	65
3.11	AUC performance: This is a model performance summary based on the AUC levels achieved by three different classifiers, namely, Random Forest (RF), Support Vector Machine (SVM) and Logistic Regression (LR)	71
3.12	Regular vs. randomized: The AUC levels of the five fold cross validation in a regular vs. a randomized random forest based models.	71
3.13	CCR performance: This is a model performance summary based on the CCR levels achieved by four different classifiers, namely, Deep Neural Network (DNN), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR)	72
3.14	DNN Training and Validation: (Left) The training and validation accuracy of the DNN model DNN model throughout all five epochs. (Right) The training and validation loss of the DNN model throughout all five epochs.	72
3.15	Virtual screening of the Prediction set: 6006168 virtually generated peptides have been screened against our four classifiers namely, DNN, RF, LG and SVM.	73
3.16	AUC performance: This is a model performance summary based on the AUC levels achieved by twelve classifiers, namely, Random Forest (RF), Support Vector Machine (SVM) and Logistic Regression (LR). For each machine learning algorithm four different descriptors were computed: (1) Dragon, (2) Atom pair fingerprints, (3) Topological torsion fingerprints, (4) MACCS keys fingerprints.	75

3.17	CCR performance: This is a model performance summary based on the CCR levels achieved by four different classifiers, namely, Deep Neural Network (DNN), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR). For each machine learning algorithm four different descriptors were computed: (1) Dragon, (2) Atom pair fingerprints, (3) Topological torsion fingerprints, (4) MACCS keys fingerprints.	76
3.18	Regular vs. randomized: The AUC levels of the five fold cross validation in a regular vs. a randomized random forest based models.	76
3.19	Virtual screening of the Prediction set: 30869 peptides have been screened against nine classifiers	77
3.20	Candidate 1 (62%)	79
3.21	Tyrocidin A	79
3.22	Candidate 2 (98%)	79
3.23	Ile-Polymyxin B1	79
3.24	Candidate 3 (72%)	79
3.25	Bacitracin A1	79
S1	Sequence based model - Logistic regression: (a) The AUC levels of the five fold cross validation in a regular model. (b) The AUC levels of the five fold cross validation in a randomized model. (c) Correct classification rate of test set	82
S2	Sequence based model - Support vector machine: (a) The AUC levels of the five fold cross validation in a regular model. (b) The AUC levels of the five fold cross validation in a randomized model. (c) Correct classification rate of test set	83
S3	Structure based models with Atom pair fingerprint: The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.	84
S4	Structure based models with Topological torsion fingerprint: The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.	85
S5	Structure based models with Morgan circular fingerprint: The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.	86

S6	Structure based models with RDKit fingerprint: The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.	87
S7	Structure based models with RDKit MACCSkeys: The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.	88
S8	Structure based models with Dragon descriptors: The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.	89

LIST OF ABBREVIATIONS

A	Adenylation
ASPs	antibiotic-sensitive pathogens
ARPs	antibiotic-resistant pathogens
AUC	Area Under The Curve
C	Condensation
CAS	Chemical Abstracts Service
CCB	Comprehensive Combinatorial Biosynthesis
CCR	Correct Classification Rate
CDC	Centers for Disease Control and Prevention
CRIStAL	Centre de Recherche en Informatique, Signal et Automatique de Lille
DNN	Deep Neural Network
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FCV	folds cross-validation
HAIs	Hospital-Acquired Infections
HCPs	Health Care Practitioners
HGT	Horizontal Gene Transfer
iChip	Isolation chip
IDLs	Inter-Domain Linkers
IMLs	Inter-Modular Linkers
LR	Logistic Regression
LIFL	Laboratoire d'Informatique Fondamentale de Lille
MIBiG	Minimum Information about a Biosynthetic Gene Cluster
MRSA	Methicillin-Resistant Staphylococcus Aureus
NHSN	National Health-care Safety Network
NRP	Non-ribosomal peptide
NRPS	Non-Ribosomal Peptide Synthetases

PCP	Peptidyl Carrier Protein
PCR	Polymerase Chain Reaction
Pep2Struc	Peptide to Structure
QSAR	Quantitative Structure Activity Relationship
RF	Random Forest
ROC	Receiver Operating Characteristic Curve
SMILE	Simplified Molecular Input Line Entry System
SVM	Support Vector Machine
TC	Tanimoto coefficient
TE	Thioesterase
TN	True Negative
TP	True Positive
TPR	True Positive Rate
XUs	Exchange units

CHAPTER 1

Introduction

1.1 Antibiotic resistant crisis

Antimicrobial resistance is recognized as one of the greatest threats to human health worldwide (Viswanathan, 2014; Ventola, 2015a; Martens and Demain, 2017). In the U.S. alone, it causes more than 2 million hospital-acquired infections (HAIs), resulting in 99,000 deaths. For instance, just one organism, methicillin-resistant *Staphylococcus aureus* (MRSA), kills more Americans every year (~19,000) than HIV/AIDS, Parkinson's disease, and homicide combined (Klevens et al., 2007). The rapid emergence of resistant bacteria is occurring worldwide, endangering our progress in healthcare, food production, and ultimately life expectancy (Golkar et al., 2014). Studies comparing the costs of infections caused by antibiotic-resistant pathogens (ARPs) versus antibiotic-sensitive pathogens (ASPs), have shown that the former has led to an annual cost of \$21 billion to \$34 billion to the US health care system and an additional 8 million hospital days. (Roberts et al., 2009; Mauldin et al., 2010; Filice et al., 2010; Spellberg et al., 2011).

The main culprits behind such crisis are the overuse, misuse and inappropriate prescription of antibiotics (Michael et al., 2014). Moreover, their extensive use in agriculture (Kennedy, 2013; Chang et al., 2015) and most importantly the lack of new drug development by the pharmaceutical industry due to reduced economic incentives and challenging regulatory requirements (Piddock, 2012; Bartlett et al., 2013; Viswanathan, 2014). Furthermore, mergers between pharmaceutical companies have also substantially reduced the number and diversity of research teams (Piddock, 2012; Bartlett et al., 2013).

1.2 Managing the antibiotic resistant crisis

The management of antibiotic resistant could be put under three categories:

1. Governmental initiatives and health care policy changes:

(i) *Optimizing Therapeutic Regimens:* Antibiotics are generally prescribed according to a fixed regimen that involves a specific dose, dosage frequency, and length of treatment. Thus, optimizing those parameters according to distinct types of infections would certainly help reducing the development of resistance. Previously, in order to ensure the complete eradication of the infecting pathogen from the body of a patient, extended regimens (patient administered a high dosage over a longer period) were usually recommended (Michael et al., 2014). However, prolonged antibiotic therapy may be pernicious as it facilitates colonization with antibiotic-resistant bacteria, which could cause recurrent episodes of infection (Luyt et al., 2014). Thus, by lowering the antibiotic dose and shortening the course of treatment, the selective pressure on bacterial organisms and the development of resistance may be reduced (Michael et al., 2014).

(ii) *Improving Diagnosis and Diagnostic Tools:* Accurate diagnosis of infectious diseases and prescribing the most proper and efficient antibiotic against it, is a highly desired goal. Doing such not only it protects the patients from being administered multiple antimicrobials simultaneously in the hope that one will be effective in controlling an unidentified pathogen but it would also help combating bacterial resistance by protecting the patient's microbiota from being subjected to intense and repeated selective pressure, which encourages the development of antibiotic resistance. Accurate diagnosis of infectious diseases using traditional methods is a tedious process that involves multiple laboratory-based tests which may take days and sometimes even weeks to complete. Fortunately, newer diagnostic techniques have emerged recently, such as real-time multiplex polymerase chain reaction (PCR) and matrix-assisted laser desorption/ionization time-of-flight mass spectrometry (Bartlett et al., 2013). The new techniques are capable of detecting the unique nucleic acid or biochemical composition of the microbe at the point of care, enabling rapid pathogen-specific identification and treatment (Luyt et al., 2014).

(iii) *Improving Tracking Methodologies*: The capabilities of federal and state governments to detect and respond to urgent or emerging antibiotic-resistant threats is currently limited. However, the Centers for Disease Control and Prevention (CDC) has recently implemented the National Health-care Safety Network (NHSN), which is the nation's most widely used healthcare-associated infection tracking system. The system is meant for use by health care facilities to electronically report infections, bacterial resistance and misuse of antibiotics. As NHSN database grows, this will enable the system to track antibiotic usage and bacterial resistance and most importantly enabling areas of concern to be addressed effectively.

(iv) *Preventing Transmission of Bacterial Infections*: "Prevention Is Better Than Cure". Modern medicine is overwhelmingly reactive rather than proactive. As it is always better to prevent an infection rather than to try to find a cure for it. Patients in hospitals are usually at a great risk for antibiotic-resistant infections when pathogens are transferred from one patient to another via the hands of careless health care practitioners (HCPs) or objects used in health care (Luyt et al., 2014). Therefore, HCPs must firmly comply with the infection-control guidelines established by the health care facility to prevent transmission of bacterial infections (Ventola, 2015b).

(v) *Governmental Legislations*: New legislations and incentives have been proposed to encourage pharmaceutical companies to re-enter the field of antibiotic drug development; these include measures to alleviate economic and regulatory obstacles, improve economic viability and provide supplemental funding for efforts in this area (Piddock, 2012; Gould and Bal, 2013; Ventola, 2015b).

2. **Developing novel antibiotics**: Most of our antibiotics are natural products extracted from bacteria and fungi. There are two main strategies to develop novel antibiotics. (1) Identifying new species of bacteria and scanning them for novel antimicrobial agents. This is the most sought strategy. For many years, this strategy was limited due to the fact that ~99% of the microorganisms that are a potential source of new antibiotics cannot be grown in a laboratory environment and therefore remain uncultured (Ling et al., 2015). However, new culturing techniques such as the isolation chip (iChip) (Kaeberlein et al., 2002; Nichols et al., 2010), that allows the growth of uncultured organisms by cultivation in their natural environment, have

paved the way to gain access to yet untapped source of new antibiotics. (2) Re-programming of bacterial genomes. This is the process of manipulating and editing bacterial biosynthetic gene clusters (BGCs) in a way to produce novel peptides with anti-microbial activity. The combinatorial biosynthesis of those BGCs will help speeding up the process of evolution by many orders of magnitude to compete with the natural evolution of new antibiotics (Nguyen et al., 2006; Bozhüyük et al., 2017; Farag et al., 2019).

3. **Making bacteria sensitive to current antibiotics:** This the process of reversing resistant bacteria to become sensitive and responsive to current known treatments. One way of accomplishing that is the use of asRNA which binds to the mRNA of the acquired resistant genes to inhibit their translation, hence rendering the bacteria to become susceptible again to antibiotics. (Good and Stach, 2011).

1.3 Non-Ribosomal Peptides

Non-ribosomal peptides (NRPs) are specialized metabolites produced by bacteria and fungi, many of which have clinical applications as antibiotics (e.g. daptomycin, vancomycin), anticancer agents (e.g. bleomycin), and immunosuppressants (e.g. cyclosporin). NRPs are synthesized by non-ribosomal peptide synthetases (NRPSs), which are exceptional mega-enzymes. Each NRPS subunit consists of multiple modules joined by linkers and each module consists of multiple catalytic domains. The four catalytic domains that are found on most NRPSs are Adenylation (A) domain, Thiolation or peptidyl carrier protein (PCP) domain, Condensation (C) domain and the Thioesterase (TE) domain. The A domain is responsible for selecting the substrate and activating it, a small peptidyl carrier protein (PCP) domain carries the activated amino acid and propagates the growing peptide chain, and a condensation (C) domain links amino acids of two adjacent modules via a condensation reaction. Therefore, all NRPS elongation modules are composed of at least three essential domains in the order [C-A-PCP]. A termination module contains an additional thioesterase (TE) domain, responsible for product release, either by hydrolysis or by macro-cyclization. Hence, all the modules work together to assemble highly complex, bioactive secondary metabolites. (Mootz et al., 2000; Baltz, 2006; Felnagle et al., 2008).

1.4 Combinatorial biosynthesis of novel NRPs

Exploiting the modularity of enzymes involved in secondary metabolism, has been proven feasible to produce novel biosynthetic compounds. Various strategies have been employed so far: a) exchanging entire NRPS subunits across different biosynthetic gene clusters (BGCs) (Nguyen et al., 2006; Coëffet-Le Gal et al., 2006; Baltz et al., 2006); b) exchanging modules (Nguyen et al., 2006); c) exchanging domains (Calcott et al., 2014); d) exchanging sub-domains (Crüseemann et al., 2013); e) using well-defined exchange units (XUs) and not modules as functional units (Bozhüyük et al., 2017). Common across all of these strategies is that the adenylation domain (A-domain) is either swapped or edited in place. Since the A-domain is the one responsible for activating the substrate that will be incorporated into the final peptide product, swapping or modifying it will potentially lead to the synthesis of a different peptide. Unfortunately, most of the NRP analogues derived using these strategies have resulted in either lower yield or no yield relative to the wild type (Stevens et al., 2005; Calcott et al., 2014; Winn et al., 2016).

1.5 IDLs vs. IMLs

It remained unclear why these strategies were not as successful as anticipated. One possible reason for the generally poor performance could be due to an incomplete understanding of the linkers role within NRPS subunits. There are two types of linkers (1) Inter-Domain Linkers (IDLs), which denotes to regions between domains (Bhaskara et al., 2013) and Inter-Modular Linkers (IMLs), which refer to the regions between modules. Studies have shown that IDLs can play an essential role in maintaining cooperative inter-domain interactions, as the composition and length of linkers affect protein stability, folding, and domain-domain orientation (Robinson and Sauer, 1998; Gokhale and Khosla, 2000). These studies and others have also provided a mechanistic insight and biochemical evidence of the importance of linker regions in controlling domain conformation and lends greater weight to previous observations that suggest careful consideration of these regions should be undertaken when attempting any combinatorial biosynthesis studies with a NRPS. Hence, understanding the nature of these linkers and their properties is substantial for successful combinatorial biosynthesis of novel BGCs.

Several methods and algorithms have been developed over the years to predict IDLs (Miyazaki et al., 2002; Udvary et al., 2002; Suyama and Ohara, 2003; Tanaka et al., 2003). Most of these methods use secondary structure predictions, amino acid propensity or a combination of the two to identify IDLs. Studies have also revealed that IDLs tend to have a relatively low secondary structure conservation and a relatively low hydrophobicity profile (Udvary et al., 2002; Bae et al., 2005). Overall, IDLs have been more well-studied (Reger et al., 2007; Doekel et al., 2008; Wu et al., 2009; Yu et al., 2013; Beer et al., 2014) as compared to IMLs (Lott and Lee, 2017; Tarry et al., 2017). When considering IMLs, the rule of thumb has been to keep them intact and not to remove, edit, or swap them. The assumption is that interfering with these linkers would prevent module-module association and therefore diminish product yield (Wriggers et al., 2005; Winn et al., 2016). This of course has led to a high level of uncertainty about the importance of IMLs, and no IML database currently exists to facilitate their analysis.

1.6 Objectives

In this study, we endeavored to address these deficiencies by scanning 39,232 bacterial genomes for potential NRPS BGCs and implementing a NRPS-Parser to extract and analyze all potential IMLs across this database. Using these data, we have established the first public IMLs database and investigated whether there is a relationship between IMLs and their adjacent A-domains. Our chief objective was to develop a better understanding of the role of IMLs in NRPSs in order to enable more efficient rational design of novel NRPs. Moreover, we implemented the NRP Discovery Pipeline, a pipeline that entails of five modules, that collectively will help accelerating the early discovery of novel NRPs with anti-bacterial activity.

CHAPTER 2

Inter-Modular Linkers play a crucial role in governing the biosynthesis of non-ribosomal peptides

This chapter is a reproduction, in whole, with permission of a publication in *Bioinformatics* by (Farag et al., 2019) at doi: 10.1093/bioinformatics/btz127

2.1 Introduction

As the threat of antibiotic resistance continues to rise and the number of available treatments continues to decline, the need to develop novel antibiotics is greater than ever. Non-ribosomal peptides (NRPs) are specialized metabolites produced by bacteria and fungi, many of which have clinical applications as antibiotics (e.g. daptomycin, vancomycin), anticancer agents (e.g. bleomycin), and immunosuppressants (e.g. cyclosporin). NRPs are synthesized by non-ribosomal peptide synthetases (NRPSs), which are exceptional mega-enzymes. Each NRPS protein consists of multiple modules, which consist of multiple catalytic domains that work together to assemble highly complex, bioactive secondary metabolites. These modules are joined together by linkers, or strings of amino acids (Mootz et al., 2000; Baltz, 2006; Felnagle et al., 2008).

Combinatorial biosynthesis of novel NRPs has been a longstanding goal in chemical biology. Five major strategies have been employed so far: a) exchanging entire NRPS genes across different biosynthetic gene clusters (BGCs) (Nguyen et al., 2006; Coëffet-Le Gal et al., 2006; Baltz et al., 2006); b) exchanging modules (Nguyen et al., 2006); c) exchanging domains (Calcott et al., 2014); d) exchanging sub-domains (Crüsemann et al., 2013); e) using well-defined exchange units (XUs) and not modules as functional units (Bozhüyük et al., 2017). Common across all of these strategies is that the adenylation domain (A-domain) is either swapped or edited in place. Since the A-domain is responsible for activating the substrate that will be incorporated into the final peptide product,

swapping or modifying it will potentially lead to the synthesis of a different peptide. Moreover, a recent study has shown that in addition to their gate-keeping function, Condensation-domains (C-domains) also exhibit a module specificity-regulatory role, which helps even further diversification of NRPs and other natural peptides (Meyer et al., 2016). Unfortunately, most of the NRP analogues derived using these strategies have resulted in either lower yield or no yield relative to the wild type (Stevens et al., 2005; Calcott et al., 2014; Winn et al., 2016).

One possible reason for the generally poor performance of these strategies could be due to an incomplete understanding of the importance of linkers within NRPSs. There are two types of linkers within NRPS assembly lines: the regions between domains known as Inter-Domain Linkers (IDLs) (Bhaskara et al., 2013) and the regions between modules known as Inter-Modular Linkers (IMLs). Studies have shown that IDLs can play an essential role in maintaining cooperative inter-domain interactions, as the composition and length of linkers affect protein stability, folding, and domain-domain orientation (Robinson and Sauer, 1998; Gokhale and Khosla, 2000). These and other studies have provided mechanistic insights and biochemical evidence of the importance of linker regions in controlling NRPS domain conformation and emphasize the relevance of linkers to combinatorial biosynthesis outcomes.

Overall, IDLs have been more well-studied (Reger et al., 2007; Doekel et al., 2008; Wu et al., 2009; Yu et al., 2013; Beer et al., 2014) than IMLs (Lott and Lee, 2017; Tarry et al., 2017). When considering IMLs, the rule of thumb has been to keep them intact and not to remove, edit, or swap them. The assumption is that interfering with these linkers would prevent module-module association and therefore diminish product yield (Winn et al., 2016). This has led to a high level of uncertainty about the importance of IMLs, and no IML database currently exists to facilitate their analysis.

In this study, we endeavored to address these deficiencies by scanning 39,232 bacterial genomes for potential NRPS BGCs and implementing a NRPS-Parser to extract and analyze all potential IMLs across this database. Using these data, we have established the first public IMLs database and investigated whether there is a relationship between each IML and its adjacent A-domains. Our chief objective was to develop a better understanding of the role of IMLs in NRPSs in order to enable more efficient rational design of novel NRPs.

2.2 Materials and Methods

2.2.1 Study design and dataset

Two major bacterial genome databases were used in this study: NCBI prokaryotic RefSeq genomes and ENA Ensembl bacterial genomes databases, comprising 70,844 and 41,610 bacterial genomes, respectively. In addition to that we also used the Minimum Information about a Biosynthetic Gene Cluster (MIBiG) repository, which contains 408 NRP BGCs (Medema et al., 2015). Due to the large amount of overlap between the two databases, 39,232 unique bacterial genomes were ultimately analyzed. We then downloaded the corresponding genomes (GenBank format) from the NCBI Genomes FTP site <ftp://ftp.ncbi.nlm.nih.gov/genomes/> and ran antiSMASH 3.0 (Weber et al., 2015), a tool that identifies and annotates specialized metabolite BGCs for the extraction of NRPS BGCs. We then applied our tool, NRPS-Parser, on all identified NRP clusters and extracted all possible IMLs. Next, we established the first IMLs database. We conducted a comprehensive analysis on all extracted IMLs in our database and investigated whether there is a relation between the IML and the activated substrates of adjacent A-domains (**Figure 1**).

2.2.2 IML NRPS-Parser

After identifying all possible NRP BGCs, a parser dedicated to extracting IMLs within NRPSs was developed and implemented. The parser extracts linkers in the following pattern: "A1-linker-A2" where A1 and A2 refer to the activated amino acid substrates of the A domains from module 1 and module 2, respectively (Supplementary Figure S1). The linker is defined as the segment of amino acids connecting these two successive NRPSs modules. All domain borders have been identified by antiSMASH 3.0 using profile Hidden Markov Models (pHMMs), which are based on multiple sequence alignments of experimentally characterized signature proteins or protein domains (proteins, protein subtypes or protein domains that are each exclusively present in a certain type of biosynthetic gene clusters).

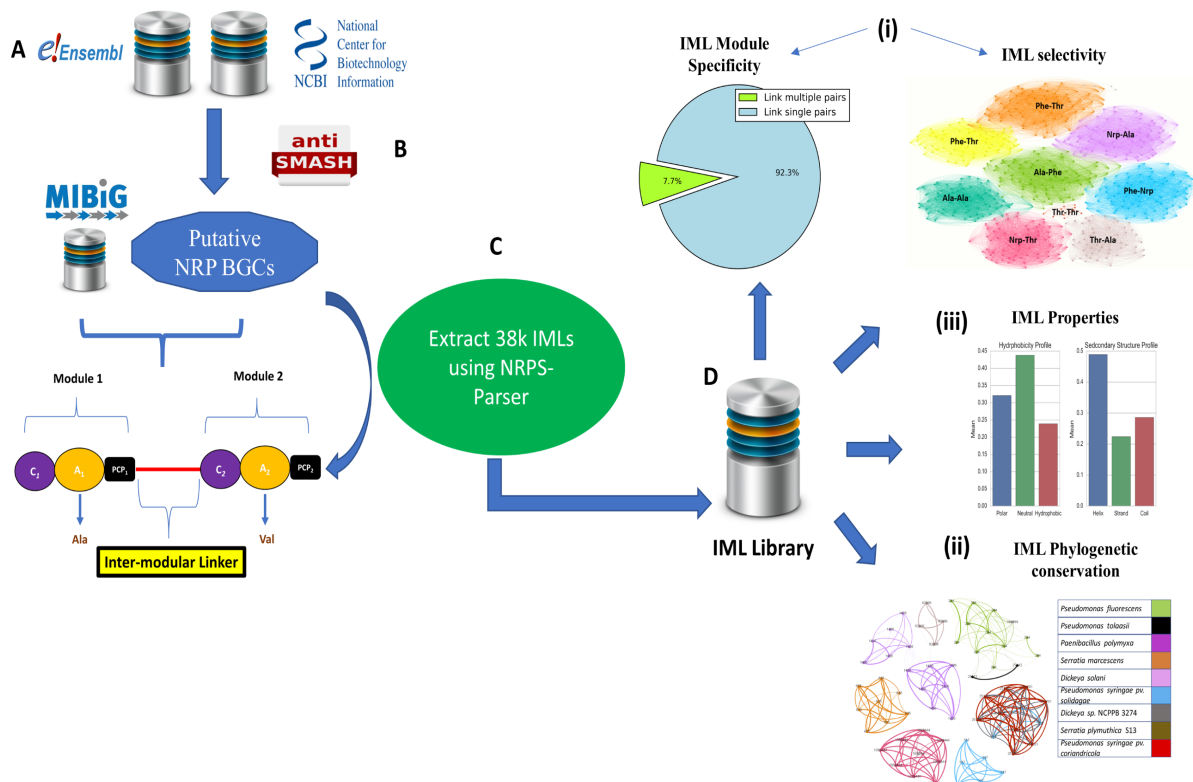


Figure 2.1: **Study design:** (A) Two bacterial genome databases and a BCG repository were processed and integrated: The NCBI prokaryotic RefSeq genomes, ENA Ensembl Bacteria and MIBiG, respectively. (B) In addition to NRPS clusters from the MIBiG repository, antiSMASH 3.0 was run on downloaded genomes to identify all potential NRP BGCs. (C) Our NRPS-Parser tool was applied to extract all possible IMLs. (D) A database of IMLs was established. All identified IMLs were then analyzed for (i) Selectivity and specificity, (ii) Phylogenetic conservation, and (iii) Properties.

2.2.3 Web-Server

All extracted IMLs are available on the following web-server (<https://nrps-linker.unc.edu>). The web-server has two major functionalities: a) a NRPS-Parser that helps to extract IMLs from uploaded antiSMASH-predicted NRPS BGCs, with support for both antiSMASH 3.0 and 4.0 outputs (Weber et al., 2015), and b) a filterable, searchable, and exportable IML database comprised of the 39,804 IMLs extracted in this study. The tool is implemented using Python 2.7 and the Flask micro-web framework. The web-server is hosted by Carolina-cloudapps, a platform for developing and deploying web applications managed by the University of North Carolina at Chapel Hill.

2.3 Results

2.3.1 IML extraction

Our overall goal was to investigate whether there is a relationship between NRPS IMLs and their adjacent A-domains. To do so, we used the well-annotated NRPS clusters from the MIBiG repository. Furthermore, we applied antiSMASH 3.0 to predict all potential NRPS BGCs from 39,232 genomes (Supplementary File 1). We then extracted all possible IMLs from the antiSMASH-predicted NRPS BGCs using our NRPS-Parser, which led to the extraction of 39,804 IMLs (902 from MIBiG NRPS clusters and 38,902 from predicted NRPS BGCs) (Supplementary File 2, File 3). The IML NRPS-Parser extracts linkers in the pattern "A1-linker-A2", where A1 and A2 refer to the activated amino acid substrates of the A domains from module 1 and module 2, respectively, and the linker is the string of amino acids joining these two successive NRPSs modules (Supplementary Figure S1). After obtaining this collection of IMLs, we then pursued two main questions: (a) How specific are IMLs with regards to particular pairs of amino-acid-incorporating modules? (b) How well conserved are IMLs within and across genera?

2.3.2 Analysis of IMLs

The 902 linkers obtained from the well-annotated MIBiG repository were extracted from 75 bacterial genera covering 196 species, while the 38,902 linkers extracted from the predicted NRPS BGCs were obtained from 138 bacterial genera covering 1,956 bacterial species. When considering all of

the extracted IMLs, their average GC nucleotide content was 13% (Supplementary Figure S2), and their average length was 42 residues. For a deeper analysis of linkers length distribution, please refer to **Figure 4** in the discussion section.

The amino acid characteristics of IMLs were composed, on average, of 44% neutral amino acids, 33% polar amino acids, and 23% hydrophobic amino acids (Supplementary Figure S3). This distribution agrees well with previous findings that linker regions tend to be less conserved in sequence and structure and contain more hydrophilic residues (Udwary et al., 2002; Bae et al., 2005). However, IMLs were found to exhibit more secondary structures than IDLs (Supplementary Figure S3). A study by George and Heringa 2002 showed that the largest proportion of IDL residues, 38.3%, adopts the α -helical secondary structure, while 13.6% are in β -strands, 10.5% are in turns and the rest, 37.6%, are in coil or bend secondary structures. On the other hand, for IML residues 49% adopt the α -helical secondary structure, while 22% adopt the β -strands and the remaining 29% are found to be in coils. This finding demonstrates the difference between IDLs and IMLs while also reflecting their distinct functional roles in coordinating pairs of modules within NRPSs.

2.3.2.1 Selectivity of unique IMLs toward pairs of modules

In this analysis, pairs of modules are represented by their activated amino acid substrates. For example, Ser-Ala indicates that module 1 activates serine and module 2 activates alanine. Here, we investigated whether IMLs act as specific linkers (i.e. bridging particular pairs of modules) or as universal linkers with no specificity towards their modules. To do so, we first considered the number of unique module pairs to which a linker could bind. In order to avoid any ascertainment bias, we performed two preprocessing steps. First, we clustered all the extracted IMLs using clust-fast from UCLUST (Edgar, 2010). Next, we removed all singletons from the dataset, so as to investigate whether the same IML tends to bind the same pairs of modules. These preprocessing steps resulted in 3,916 unique IML clusters. All clusters show less than 80% sequence similarity to each other. The pairwise centroid similarity distribution is depicted in Supplementary Figure S4.

Among all IML clusters, 92% (3,616) were associated with only a single pair of modules (Supplementary Figure S5A). For example, there are 427 occurrences of the linker 'SITDAAASQD-DWVIVHDPE' in our database, which have been extracted from five different bacterial genera and are involved in the biosynthesis of 45 distinct NRPs. Each occurrence of this linker, regardless of

genera or NRP product, links the same Gly-Cys module pair. Thus, a single IML typically bridges the same module pair. The remaining 8% of the linkers (300) tend to join only a limited number of module pairs (ranging between 2 and 13 unique pairs). For example, the linker 'ENTEVLPPIPLAPR', extracted from a single strain (*Burkholderia pseudomallei* 406e), bridges five distinct module pairs (Supplementary Figure S5B). In addition, our analysis has shown that module pairs are not reversible: the IMLs between Ser-Ala modules differ from those that link Ala-Ser. Overall, it appears that IMLs are highly selective linkers in regard to the amino acids incorporated by their flanking modules.

An alternative way to illustrate the high level of IML selectivity is to examine the IMLs of a single bacterial species in a network. We selected *Mycobacterium abscessus* to illustrate this method, since it contains a reasonable number of linkers to depict in a two-dimensional network and its linkers bridged a range of distinct module pairs. In this network visualization, the IMLs are represented as nodes, with edges connecting nodes that show at least 80% similarity based on pairwise sequence alignment using the Needleman Wunch algorithm. We then applied a Louvain community detection algorithm (Blondel et al., 2008), which detected nine distinct communities, each of which consisted of linkers that bind specifically a distinct pair of modules (**Figure 2**, Supplementary Table 1). Similar results were obtained when we conducted the same analysis on linkers extracted from *Burkholderia pseudomallei* (Supplementary Figure S6). These visualizations further supports the conclusion that module-specific IMLs dominate within NRPS BGCs.

2.3.2.2 Phylogenetic conservation of module-specific IMLs within and across genera

Our analysis so far indicates that IMLs are very selective towards module pairs. Here, we probe whether pairs of modules tend to be linked by the same IML regardless of the bacterial species from which they were extracted. We conducted an all-by-all comparison of module pairs vs. genera (computing the degree of conservation of IMLs linking a specific module pair both within and across genera) and then built a community network to visualize the phylogenetic distributions of IMLs that link the same module pair.

All-by-all comparison analysis: The NRPS BGCs from the MIBiG repository contain 116 unique module pairs. For every module pair we computed the similarity matrix of all its linkers using the Needleman Wunch algorithm, with an 80% similarity cut-off. Of the 2,854 pairwise comparisons,

Mycobacterium Abscessus

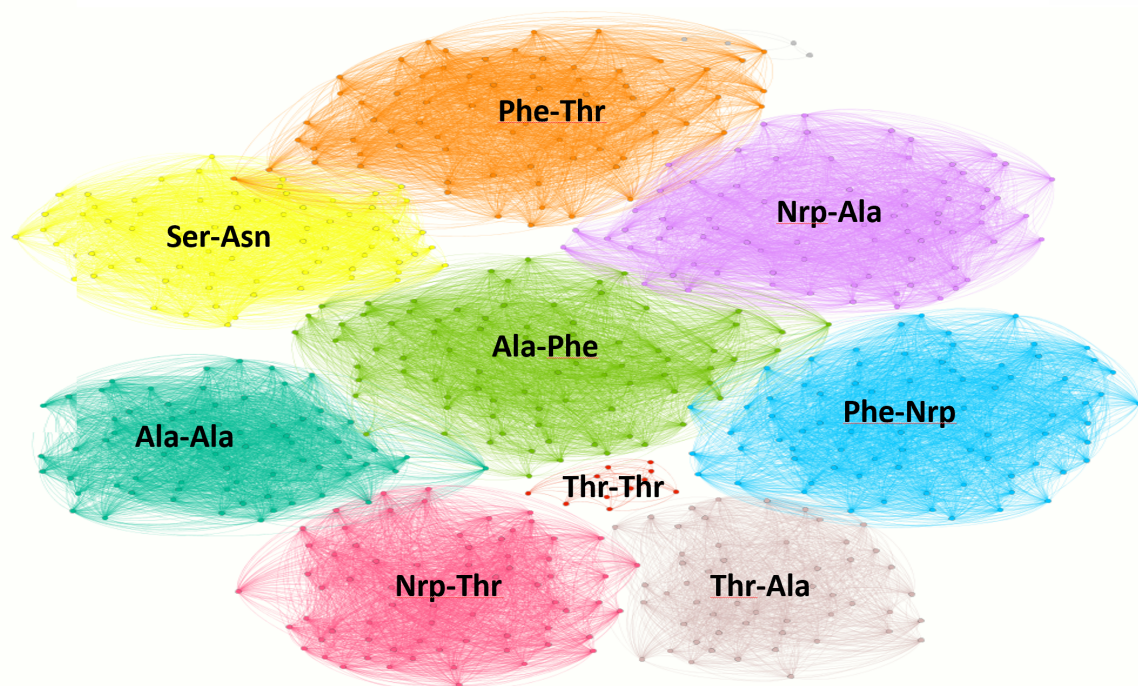


Figure 2.2: ***Mycobacterium Abscessus* Module-Specific IMLs**: Community network, where nodes refer to linkers, and edges are constructed between two linkers, if they share 80% or more sequence similarity. The graph depicts nine distinct communities. Each community represents all the linkers that bind a specific pair of modules. For instance, the orange community refers to all linkers that bind the pair of modules that activate phenylalanine and tyrosine.

just 10% (285) were able to reach or exceed the 80% similarity cut-off (Supplementary Figure S7A). Of these 285 comparisons that were highly similar, 85% were from IMLs obtained from the same bacterial genus, and 15% were from IMLs obtained from different genera (Supplementary Figure S7A). Of the remaining 90% (2569) of comparisons that exhibited a low degree of conservation, 60% were between linkers extracted from different genera (Supplementary Figure S7A). These findings indicate that module-specific IMLs tend to be more conserved within bacterial genera (Supplementary Figure S7B), whereas multiple distinct IMLs exist that link the same module pair across different genera (Supplementary Figure S7C). Furthermore, 83% of the IMLs that come from the same genera, yet show a low degree of conservation, were extracted from different species (Supplementary Figure S7D). When we expanded this same analysis to the larger set of predicted NRPS BGCs, very similar results were obtained (Supplementary Figure S8A). Both analyzed data sets show multiple cases of highly similar IMLs, if not completely identical, despite being extracted across distinct genera. The main reason behind such observation, is the horizontal gene transfer phenomena (HGT) which is the movement of genetic material between unicellular and/or multicellular organisms other than by the transmission of DNA from parent to offspring (vertical). For example, we found 27 instances of the IML 'VAL-ESKEEQTFEPIRQAP-ASP' across 3 different genera *Bacillus*, *Brevibacterium* and *Jeotgalibacillus* (Supplementary Figure S8B). Another example revealed 427 instances of the IML 'GLY-SITDAAASQDDWVIVHDPE-CYS' across 4 different genera *Citrobacter*, *Escherichia*, *Klebsiella* and *Enterobacter* (Supplementary Figure S8C).

Community network visualization of Thr-Val IMLs: We constructed a community network visualization to illustrate the phylogenetic specificity of IMLs. We took all IMLs for Thr-Val pair obtained across all species and created a graph as described above. After applying the Louvain community detection algorithm (Blondel et al., 2008) to this data, the nodes were colored based on the bacterial species they were obtained from. If IMLs were globally conserved across many bacterial species, we would expect to obtain a single large community network with multi-colored nodes. If instead IMLs were conserved within a single bacterial species, we would expect multiple distinct communities to be detected, where nodes within each community would have the same color. The data indicate that the latter is the case, underscoring the phylogenetic specificity of IMLs (**Figure 3**).

2.3.2.3 IMLs as independent building blocks

We next wanted to explore whether IMLs were highly associated with single NRP products, or whether the same IML was involved in the biosynthesis of distinct NRPs. If the latter was the case, then IMLs could potentially act as biosynthetic building blocks to generate novel NRPs. To begin this analysis, we first needed to define the NRP products produced by our extracted NRPS BGCs. The NRPs from the MIBiG repository were already well-annotated, but that was not the case for the NRPS BGCs predicted by antiSMASH. In order to carefully identify duplicates among the group of predicted BGCs, we developed an expedited homology comparison based on cluster-prints. A cluster-print is a string representation of a BGC where each character (separated by a comma) refers to a specific NRPS domain and hyphens are used as a delimiter to distinguish between different NRPS polypeptides. This method permits BGCs to be quickly compared to one another while avoiding complex sequence comparisons. For example, the cluster-print for tyrocidine, an NRP from *Bacillus brevis*, would be [A, T, E, -, C, A, T, C, A, T, C, A, T, E, -, C, A, T, C, A, T, C, A, T, C, A, T, C, A, T, C, A, T, C, A, T, -, T]. When two BGCs show identical cluster prints, we then compare the sequence of their predicted activated substrates. For example, for tyrocidine this would be [dPhe - Pro, Phe, dPhe, Asn - Gln, Tyr, Val, Orn, Leu]. We were thus able to determine how many unique NRPs a single IML was associated with (Supplementary File 4).

Our analysis has revealed the presence of 2,703 IMLs that were involved in the biosynthesis of at least two or more distinct NRP products based on their cluster-prints. For instance, the IML 'Gly-LAPAAQGGIVRCARDA-Thr' was found in 90 distinct NRP products across 3 different species. When we conducted the same analysis using the well-annotated NRPs from the MIBiG repository, we similarly observed that some IMLs are involved in generating distinct NRP products. For instance, the BGCs of syringomycin and syringopeptin, both produced by *Pseudomonas syringae*, share multiple identical IMLs. Moreover, we also observed that highly similar linkers with more than 90% similarity are involved in the biosynthesis of distinct NRP products. For example, the IML 'Ile-AGRSSLPPIVPVS-Rnp' is involved in the biosynthesis of sessilin (produced by *Pseudomonas sp.* CMR12a), while the IML 'Leu-AGRSSLPPILPVS-Rnp' is involved in the biosynthesis of

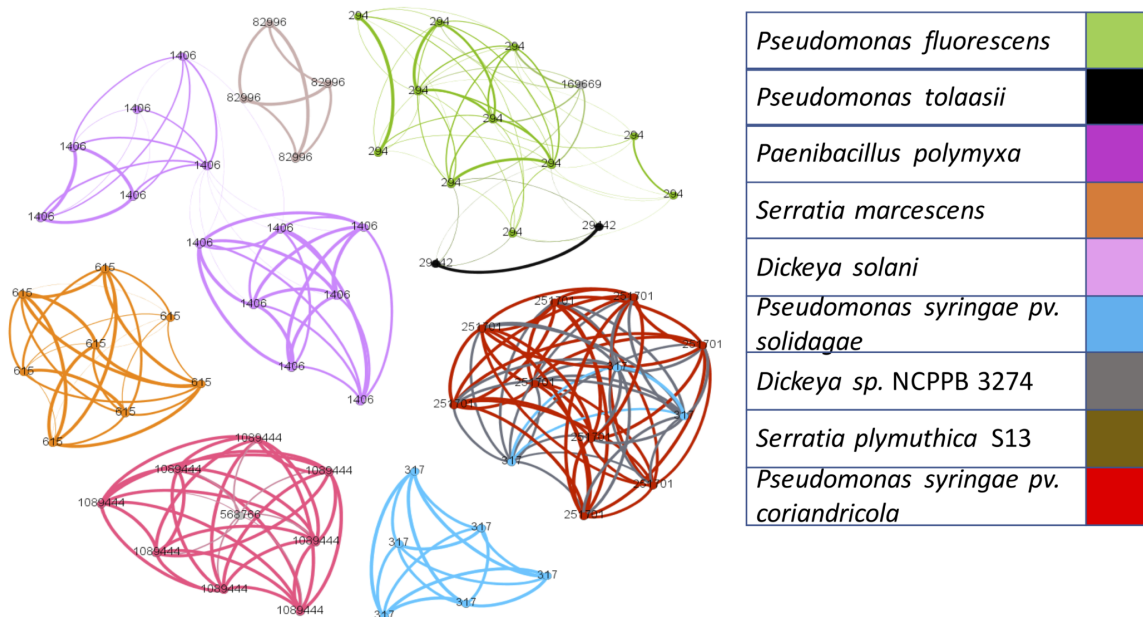


Figure 2.3: **(Thr-Val) IMLs community network:** A network of all the linkers that bridge the Thr-Val module pair. These linkers belong to various distinct communities, despite the fact that they are all linking the same pair. The coloring of the graph refers to the species from which linkers were extracted.

tolaaasin (produced by *Pseudomonas costantinii*). These results not only indicate that highly similar to identical module-specific IMLs can be utilized to generate distinct NRPs, but also validates the application of our cluster-print approach to detect distinct NRP-generating BGCs and reflect the major role HGT play in bacterial evolution.

2.4 Discussion

There are two major resources hosting well-identified NRPs:(a) the NORINE database (Caboche et al., 2008) and (b) the MIBIG repository (Medema et al., 2015). The former includes 1,187 NRPs, while the latter contains 433 NRPs. However, when it comes to putative NRPS BGCs, our study comprises a total of 51,810 potential NRPS clusters (Supplementary Table 2), from which 7,441 are identified as completely assembled NRPS clusters. We defined complete clusters as those possessing at least three modules and two IMLs. To the best of our knowledge this is the largest number of putative NRPS BGCs predicted from known genomic databases (39,232 bacterial genomes). Other

studies have reported only 6,351 (Dejong et al., 2016) and 1,704 (Cimermančić et al., 2014) NRPS clusters.

All of the extracted NRPS BGCs were classified into 7,365 unique cluster-prints, each of which potentially generates a novel NRP (Supplementary Figure S10A). If so, the inclusion of the additional genomes results in a 27-fold increase in potential NRPs compared to those captured in the MIBiG repository. This increase likely reflects the fact that the MIBiG repository is based on 243 unique bacterial strains, while our more complete genome analysis comprised over 31,338 unique bacterial strains (based on their NCBI taxonomy identity designator) (Supplementary Figure S10B). The large number of bacterial genomes processed in our study and the many NRPS BGCs identified using antiSMASH will certainly help the community by revealing potentially novel, not-yet-annotated NRPS BGCs. We are confident that a similar increase in NRPSs would result from expanding the scope of this analysis to include fungi and plants.

IMLs could be clustered into three clusters based on their lengths (**Figure 4**): (1) Linkers with lengths ranging between 9-120 amino acids. These are typical lengths and they represent 80% of all IMLs. (2) Linkers with lengths ranging between 160-280 amino acids. These are linkers that succeed an epimerization domain in a BGC and they represent 19.85% of all linkers. These seem longer, due to only-recently-annotated domain "TIGR01720" in TIGRFAMs protein family (Haft et al., 2001), which is located immediately downstream of the epimerization domain and upstream of the condensation domain of the successive module. (3) Linkers with length ≥ 300 amino acids. These are most certainly outliers and they represent less than 0.15% of all linkers. The genesis of these outliers is due to the limitation that antiSMASH tool sometimes has in properly defining border domains in case of new yet undefined and unannotated domains.

The identification of borders between distinct domains is crucial for our analysis. IML is the linker region between two successive modules. Precisely, the region between the peptidyl carrier protein domain (PCP or T-domain) of the first module and the condensation domain (C-domain) of the successive module. Thus, determining where the T-domain ends and the C-domain begins is vital for extracting the right linker region. Unfortunately, there is a lack of multi-modular crystal structures for NRPSs. Therefore, we used antiSMASH to predict all potential NRP BGCs, from which we extracted all our IMLs. antiSMASH depends on pHMMs to predict domains borders

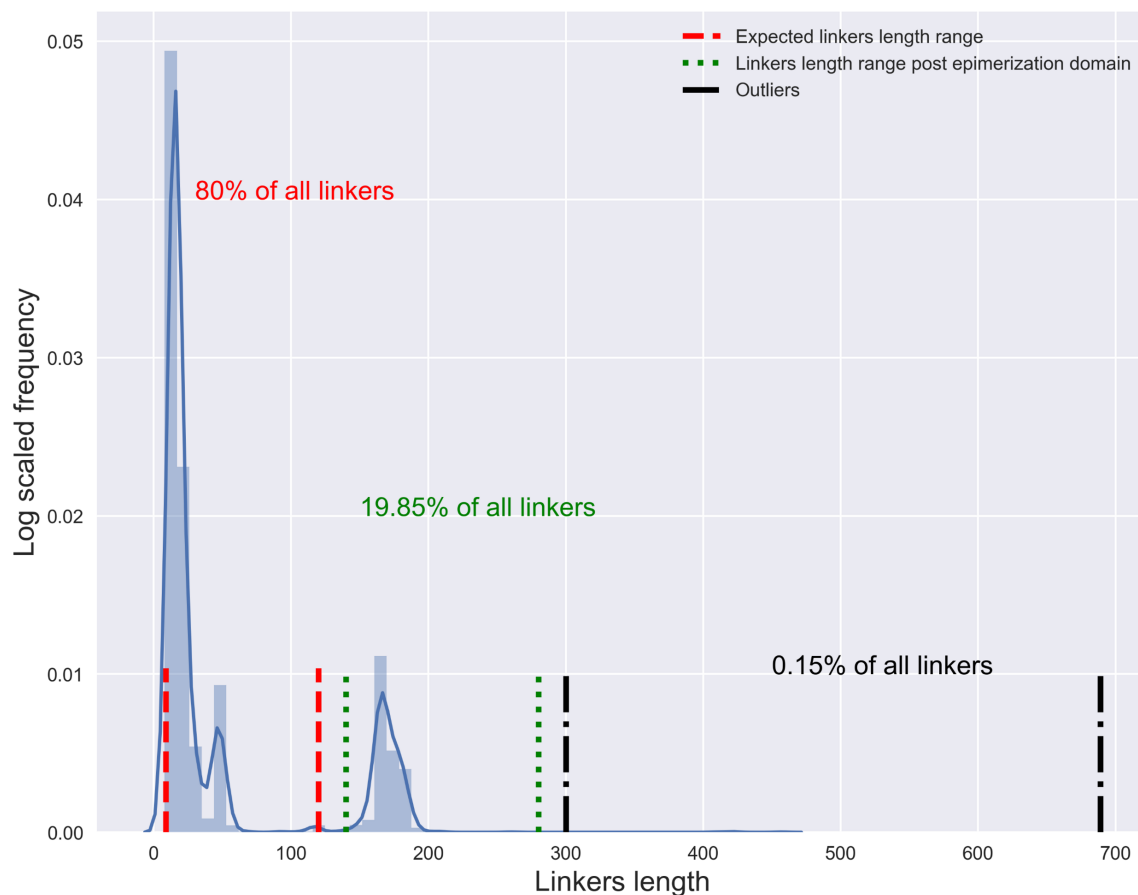


Figure 2.4: **Linkers length distribution:** There are three clusters of IMLs based on their lengths: (red) Linkers with lengths ranging between 9-120 amino acids. (green) Linkers with lengths ranging between 160-280 amino acids. (black) These are linkers with more than 300 amino acids in length (outliers).

in a BGC. Hence, the quality of our extracted linker regions is only as good as the antiSMASH domain border identification algorithm. Fortunately, these pHMMs are constantly being updated and re-trained, allowing for improved predictive power as new data and new annotated domains are added.

Historically the importance of IMLs compatibility with adjacent modules has not been considered during NRP biosynthesis strategies. For instance, Nguyen et al. 2006 have applied several combinatorial biosynthesis strategies to produce a library of daptomycin analogues. Among other approaches the authors replaced entire modules within NRPS subunits. Here, we will focus on the module replacement strategy, and the role that the IML considerations might have played in their results. All the derived peptides are based on daptomycin, a cyclic 13-amino acid lipopeptide

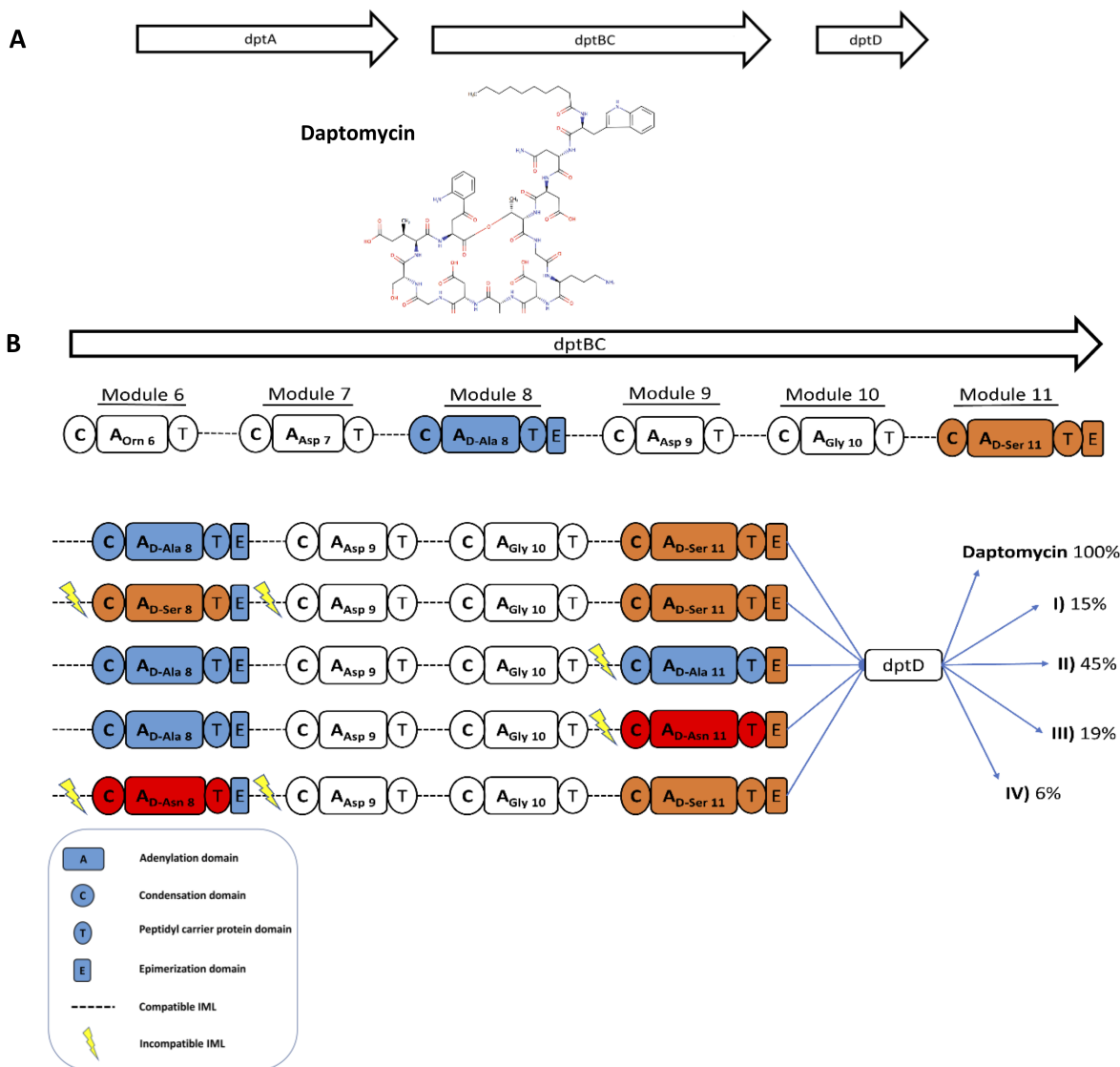


Figure 2.5: Retrospective analysis of daptomycin analogues biosynthesis: (A) Daptomycin BGC from *Streptomyces roseosporus*. **(B)** NRPS organization of the daptomycin cluster and schematic showing module exchange strategy. Modules 8 and 11 were swapped for each other, or for the Asn 11 module from A54145 biosynthesis from *Streptomyces fradiae*. The swapping resulted in the synthesis of four daptomycin analogues I, II, III, and IV each with 15%, 45%, 19% and 6% yield relative to the wild-type. The lightning bolts signify IML incompatibility.

obtained from *Streptomyces roseosporus* that is a product of three biosynthetic NRPS subunits, dptA, dptBC and dptD (**Figure 5A**).

Two strategies were conducted by Nguyen et al. 2006 : (A) Exchange of homologous modules within the dptBC NRPS subunit. The other one was to undergo an (B) Exchange of single heterologous modules.

Exchange of homologous modules within dptBC: Here, Nguyen et al. 2006 decided to conduct two experiments that involve replacing entire modules (C-A-T) within the dptBC NRPS subunit. Module 8 and module 11, which activate D-Ala8 and D-Ser11, respectively, were replaced. (I) The D-alanine encoding C-A-T from module 8 was deleted and replaced with the C-A-T from module 11 (change of Ala8 to Ser8). (II) The opposite replacement was also made where the C-A-T from module 11 was replaced with the C-A-T from module 8 (change of Ser11 to Ala11). The E domains of each module were left intact in an attempt to preserve the downstream inter-module associations. Production of the predicted D-Ser8 and D-Ala11 containing daptomycin analogues was observed, albeit at reduced production levels of approximately 15% and 45% relative to wild-type. The authors reasoned that the success of synthesizing those daptomycin analogues was due to the fact that both modules are highly homologous. However, the authors failed to explain why the yields were much lower than the wild-type and why the yield of (I) was lower relative to (II) (**Figure 5B**).

We hypothesize that a possible reason for these decreased yields is due to IML incompatibility after module replacement. In the first experiment a middle module was replaced, giving rise to two incompatible IMLs (one on each side of the replaced module). In the second experiment, a terminal module was replaced, causing a single incompatible IML. Thus, the yield was 15% and 45% for the first and the second experiment, respectively.

Exchange of single heterologous modules: Here, module 11, which is selective towards D-Asn11, was obtained from the A54145 BGC from *Streptomyces fradiae*. The extracted module was used to replace either D-Ala8 or D-Ser11 from dptBC NRPS subunit of the daptomycin BGC. This approach led to the isolation of two new analogues (D-Asn11 (III) and D-Asn8 (IV)); however, yields were even further reduced relative to wild type, i.e., 19% and 6%, respectively (**Figure 5B**).

We again hypothesize that the main reason for such a steep drop in the yield is due to the impact of incompatible IMLs post module replacement. Moreover, we know from our analysis that IMLs

are not conserved across species and thus replacing modules across species would further increase the level of IML incompatibility and would result in a more pronounced effect on the product yield. Similar analysis was conducted on the findings of Bozhüyük et al. 2017 (Supplementary information Retrospective analysis). These observations show that the compatibility of the IML with the entire adjacent A-domain is critical to ensure a proper yield of the NRP product. These data support the idea that module-specific IMLs are critical to the successful generation of NRPs.

2.5 Conclusion

Using our IML NRPS-Parser, we extracted more than 39k NRPS IMLs and analyzed their association with their adjacent A domain substrates. This led to the discovery that IMLs are very specific to the A domain modules that they connect, with more than 92% of the identified IMLs being associated with a specific pair of modules. We also determined that the same IML could be involved in the biosynthesis of different NRP products across various bacterial genera (Supplementary File 4). Overall, however, IMLs that link a particular module pair show a low degree of conservation across bacterial genera. We also determined that IMLs exhibit more secondary structures (α -helices) than IDLs, however, they share similar hydrophobic profile. Furthermore, as a proof-of-concept, we retrospectively analyzed the findings of (Nguyen et al., 2006) and (Bozhüyük et al., 2017) demonstrating that IMLs incompatibility could dramatically impact biosynthetic yields of daptomycin lipopeptides and ambactin analogues. Overall, our data indicate a strong relationship between NRPS IMLs and their adjacent A domains. This finding suggests that, going forward, combinatorial biosynthesis strategies to generate novel NRPs should consider IMLs in addition to other established parameters (Nguyen et al., 2006; Coëffet-Le Gal et al., 2006; Baltz et al., 2006; Crüseemann et al., 2013; Calcott et al., 2014; Meyer et al., 2016; Bozhüyük et al., 2017).

All 39,804 IMLs extracted in this study (Supplementary Table 2) as well as our parser are publicly available at <https://nrps-linker.unc.edu/>. We anticipate this tool will not only facilitate mining the data we have analyzed here, but will also enable interested researchers to expand their studies as new genomes (bacterial, fungal, and plant) are obtained. Our study lays the foundation for future experimental validations of our hypothesis that IMLs play a crucial role in governing the biosynthesis of NRPs. We expect that additional approaches and tools could be developed that rely

on this finding and facilitate the design of novel NRPS BGCs using the most appropriate IMLs for combinatorial biosynthesis of novel NRPs.

2.6 Supplementary

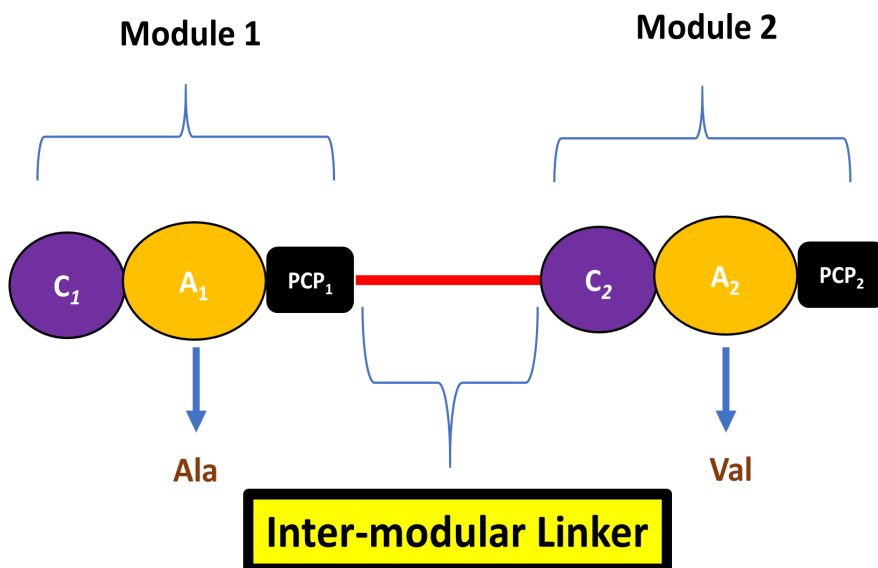


Figure S1: **Inter-Modular Linker:** The NRPS-Parser extracts linkers in the following pattern “A1-linker-A2”. A1 and A2 refer to the activated substrates of the Adenylation domains from module 1 and module 2, respectively. The linker is the segment of amino acids linking these two successive NRPSs modules.

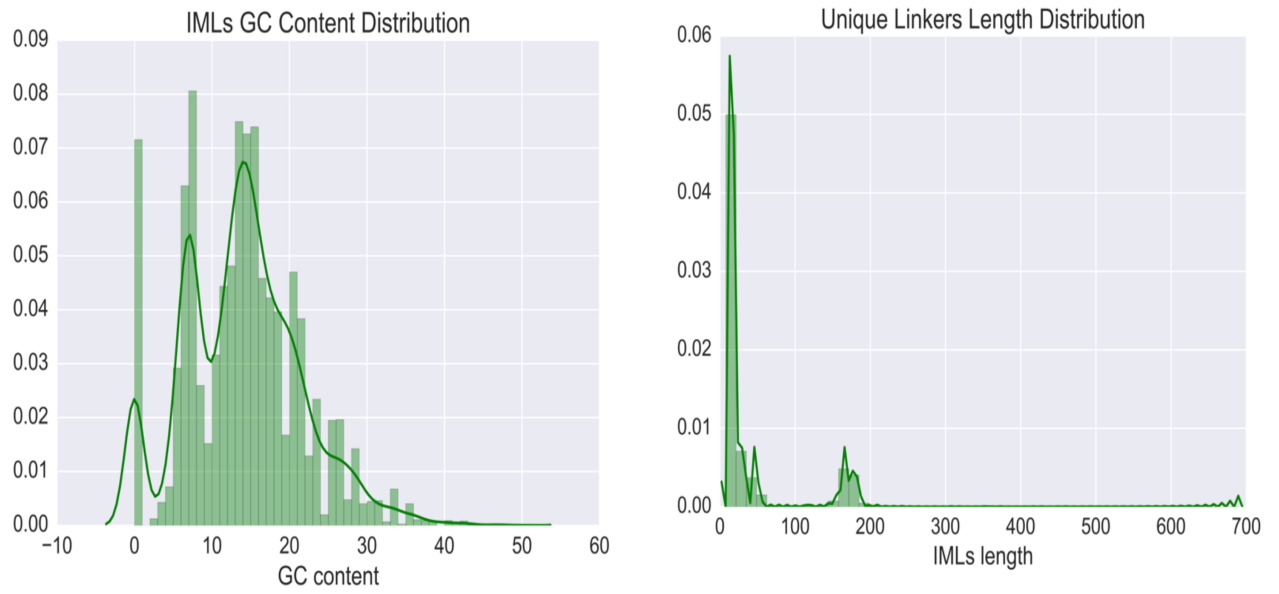


Figure S2: GC Context and IML length: (Left) Distribution of GC content across all species. (Right) Distribution of linker lengths across all species.

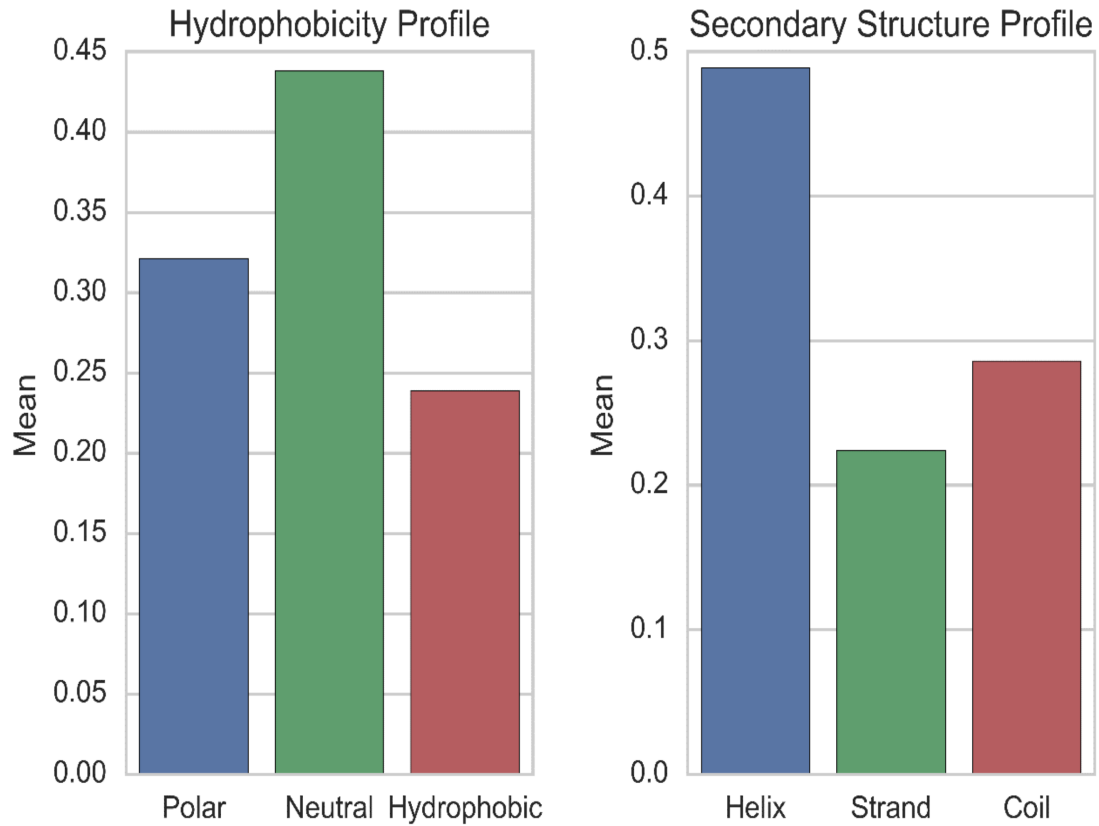


Figure S3: **IMLs Hydrophobicity and Secondary Structure Profiles:** (Left) On average linkers were found to be composed of 44% neutral amino acids, 33% polar amino acids and 23% hydrophobic amino acids. (Right) 49% of all secondary structures were α -helices, while strands and coils comprised 22% and 29%, respectively.

Linkers pairwise sequence similarity/ identity/ gap distribution

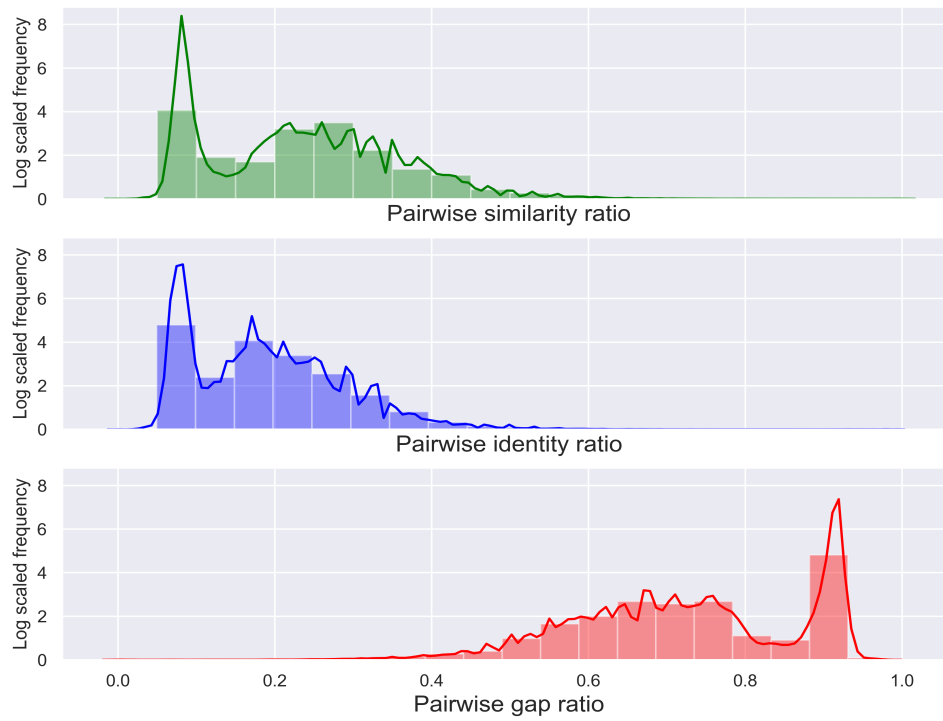


Figure S4: **Linkers pairwise distribution:** Based on sequence similarity (green), based on sequence identity (blue) and based on gap ratio (red).

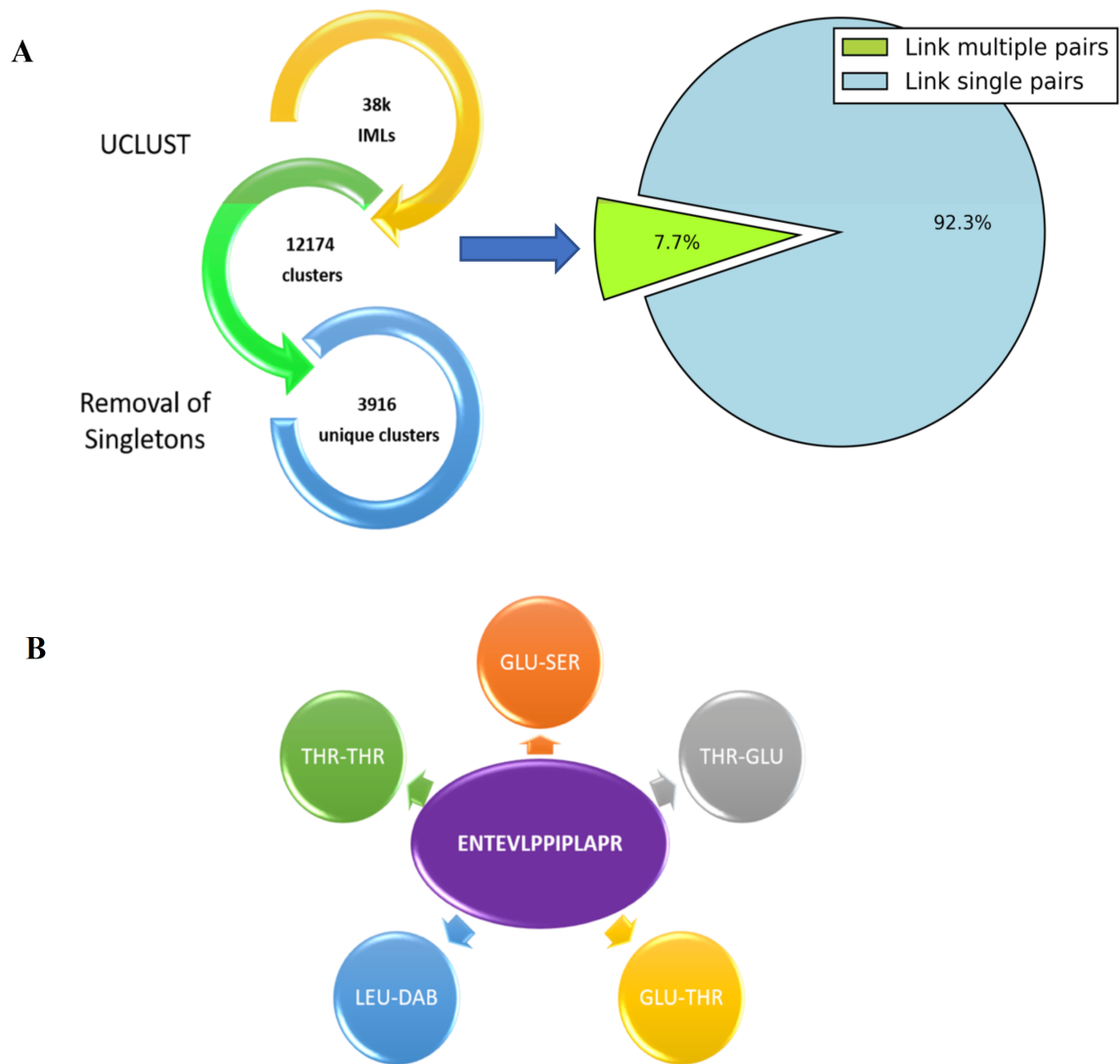


Figure S5: **IMLs Selectivity:** (A) Clustering the 39,804 extracted IMLs using the cluster-fast algorithm from UCLUST lead to 12,174 clusters. The removal of singletons resulted in 3,916 unique IMLs centroids (clusters). Only 8% of all unique IMLs tend to bridge multiple pairs of modules, while the remaining 92% link specifically just to a single pair of modules (B) An example of an IML, extracted from *Burkholderia pseudomallei* 406e linking five distinct pairs of modules.

Pairs of Modules	Community size	#BGCs	#Unique BGCs
NRP_ALA	135	134	10
ALA_PHE	130	129	53
NRP_NRP	124	241	78
PHE_NRP	124	124	52
PHE_THR	123	123	24
NRP_THR	122	241	78
ALA_ALA	122	121	21
THR_ALA	90	93	31

Table S1: **Mycobacterium Abscessus Module-Specific IMLs:** Metadata of the community network, showing the size of each community, number of unique and redundant BGCs from which the IMLs were extracted.

Burkholderia pseudomallei

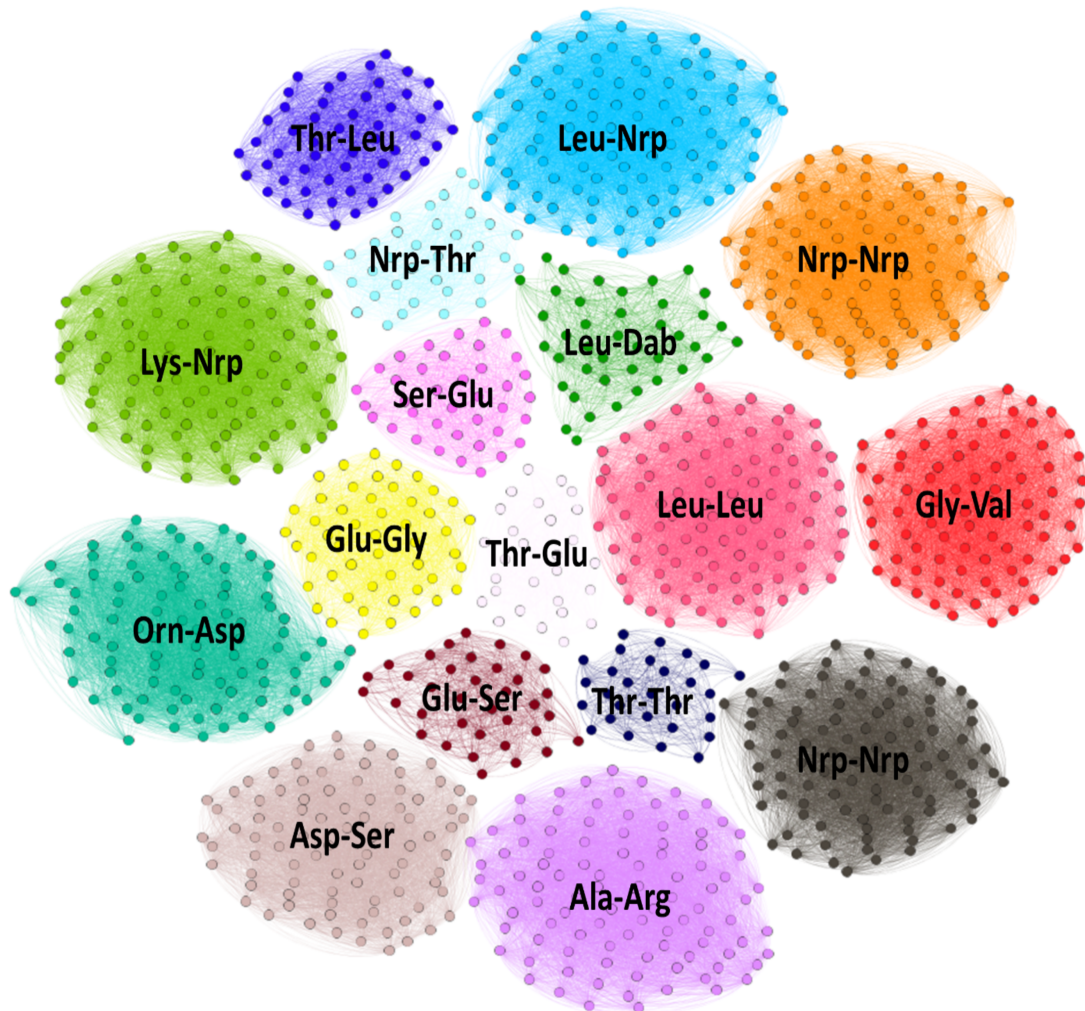


Figure S6: ***Burkholderia pseudomallei* module-specific IMLs:** Community network, where nodes refer to linkers, and edges are constructed between two linkers, if they share 80% or more sequence similarity. The graph depicts seventeen distinct communities. Each community represents all the linkers that bind a specific pair of modules. For instance, the red community refers to all linkers that bind the pair of modules that activate glycine and valine, respectively.

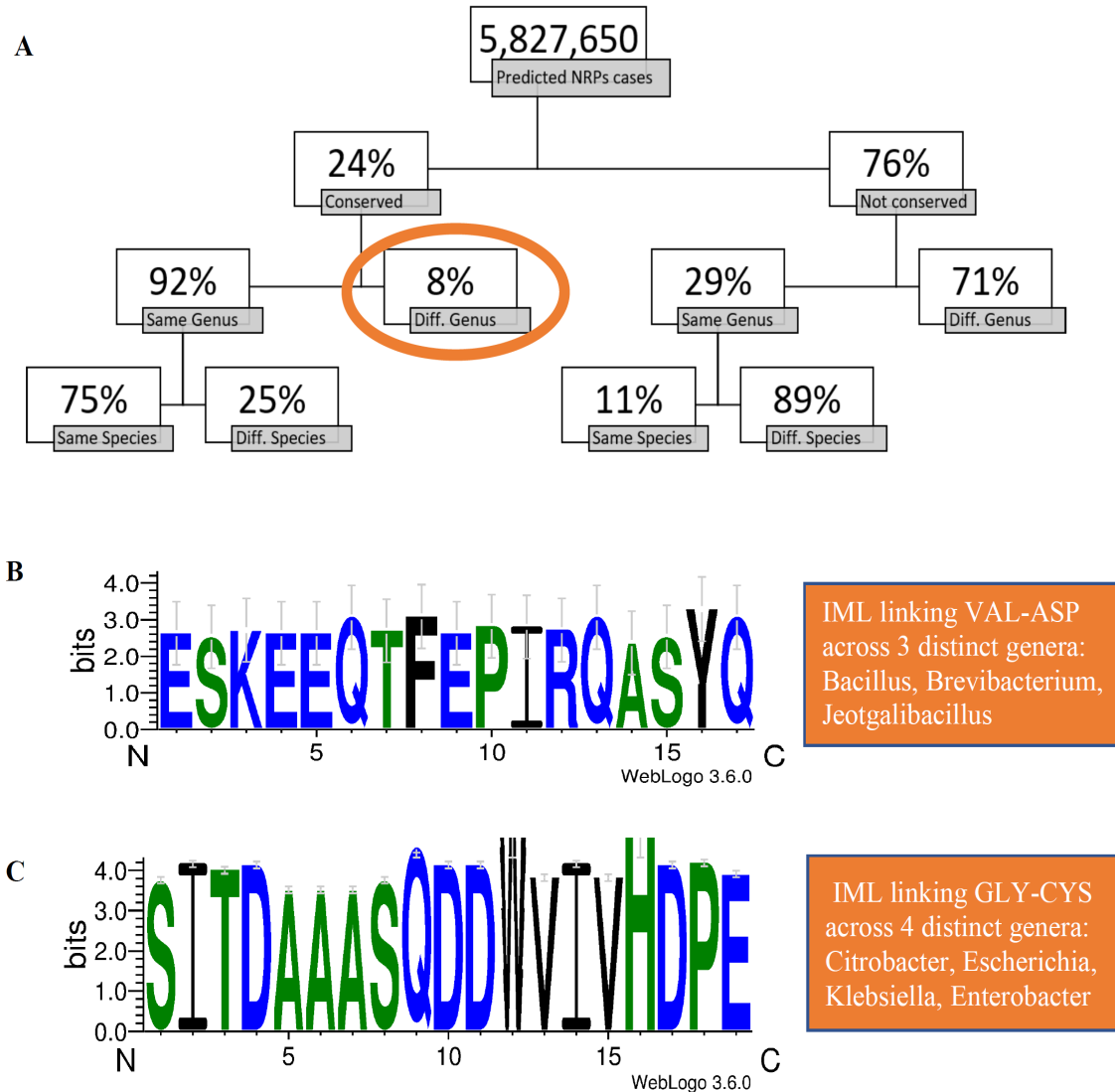


Figure S8: **Pairwise similarities of IMLs linking the same pair of modules:** (A) Total pairwise comparisons of linkers (5,827,650) that link the same module pairs retrieved from the 51,810 potential NRP clusters. Only 24% of all comparisons showed sequence similarity of over 80%, while the remaining 76% showed a lesser degree of conservation. (B) A sequence logo of 27 IMLs all linking the same pair of module VAL-ASP, extracted from 3 distinct genera: *Bacillus*, *Brevibacterium* and *Jeotgalibacillus*. (C) A sequence logo of 427 IMLs all linking the same pair of modules GLY-CYS, extracted from 4 distinct genera: *Citrobacter*, *Escherichia*, *Klebsiella* and *Enterobacter*.

Items	Count
Bacterial genomes	39232
Unique Taxonomy-ids	31338
NRP BGCs	51810
Unique NRP BGCs (based on cluster-print)	7326
Inter-modular linker	38902
Unique Inter-modular linker	12905
Unique pairs of modules	398

Table S2: **Data Summary:** Overview of data analyzed in this study.

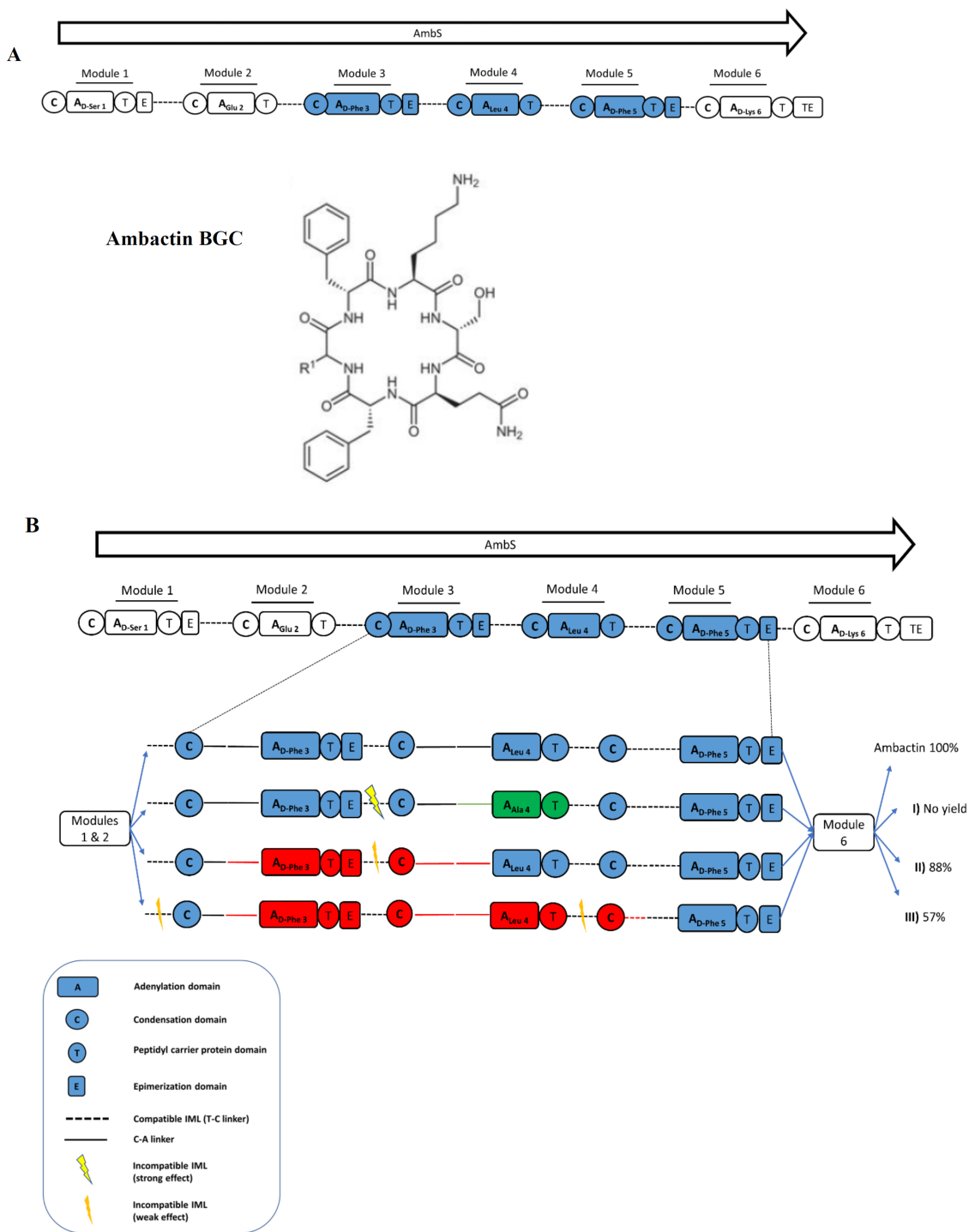
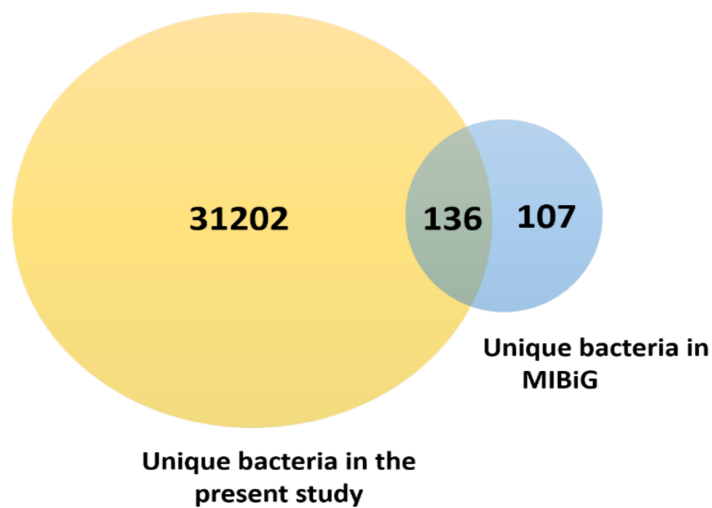


Figure S9: Retrospective analysis of Ambactin analogues biosynthesis: (A) Ambactin BGC from *Xenorhabdus miraniensis*. **(B)** NRPS organization of the Ambactin cluster and schematic showing XU units exchange strategy. I) Phe-specific XU3 was exchanged against an Alaspecific XU from the Kolossin NRPS BGC from *Photorhabdus luminescens*, resulted in no product. II) Phe-specific XU3 against a Phe-specific XU from the GxpS NRPS, resulted in approximately 88% yield relative to wild-type. III) Phe-specific XU3 and Lys-specific XU4 was exchanged against the two building blocks XU3 and XU2 from GxpS, respectively. This resulted in approximately 57% yield relative to wild-type.

A



B

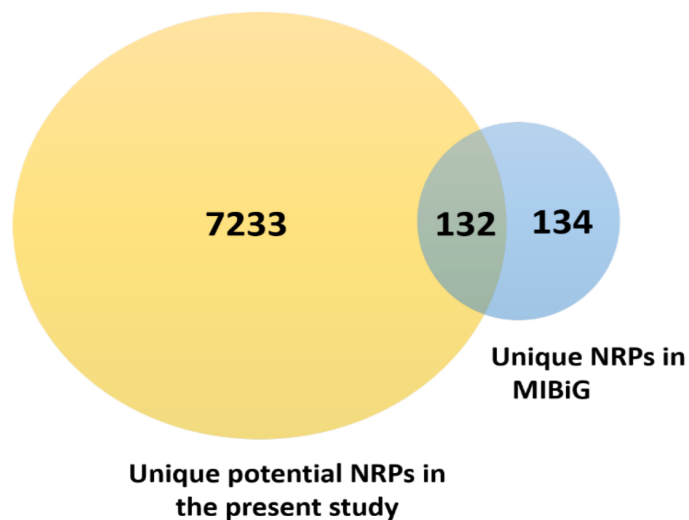


Figure S10: **Our Dataset vs. MIBiG:** (A) Venn diagram showing number of unique bacteria involved in our study (31,338) as compared to number of unique bacteria biosynthesizing NRPs in the MIBiG repository (243), with only 111 bacteria shared between them. (B) Venn diagram showing number of unique NRP cluster-prints identified in this study (7,365) as compared to 266 from the MIBiG repository, with only 132 NRP cluster-prints shared between them.

CHAPTER 3

NRP Discovery Pipeline

3.1 Introduction

Non-ribosomal peptide synthetases (NRPSs) are large multienzyme machineries that assemble numerous peptides with large structural and functional diversity. These peptides include more than 20 marketed drugs, such as anti-bacterials (penicillin, vancomycin), antitumor compounds (bleomycin), and immunosuppressants (cyclosporine) (Mootz et al., 2000; Keller and Schauwecker, 2003; Süßmuth and Mainz, 2017). Exploiting the modularity of these by reprogramming them genetically, would immediately spur chances to generate analogues of existing drugs or new compound libraries of otherwise nearly inaccessible compound structures (Nguyen et al., 2006; Coëffet-Le Gal et al., 2006; Baltz et al., 2006). However, the number of combinatorial possibilities is vast and extensive. As of early 2019, more than 500 monomers have been annotated and identified. For instance, with the current number of identified monomers, the number of possible to be generated peptides with an average length of 15 is $\sim 500^{15}$ NRPs (Caboche et al., 2007; Pupin et al., 2016). It goes without saying that the process of determining the right peptide combination, is very expensive and time-consuming as it requires a large amount of human overhead and expertise. Thus, with such a big data and large chemical space, the need for computational approaches is inevitable. Here we implemented “NRP Discovery Pipeline” a computational approach that leverages the importance of inter-modular linkers (IMLs) in combinatorial biosynthesis of novel NRPs (Farang et al., 2019) and uses machine learning techniques, to build rigorous and highly predictive classifiers, to help in the early discovery of novel NRPs with anti-bacterial activity.

3.1.1 Pipeline Overview

A computational approach that help discovering and guiding rational design of novel NRP(s). The pipeline entails five major phases:

1. **NRP comprehensive combinatorial biosynthesis (CCB):** A tool that helps virtual generation of large libraries of NRPs.
2. **NRP sequence-based predictor:** A binary statistical model based only on peptide sequences to filter out all inactive predicted peptides.
3. **Pep2struc:** A tool that helps converting peptide-sequences to their 2D structures.
4. **NRP structure-based predictor:** A binary statistical model based on peptide structures to filter out all inactive predicted peptides.
5. **NRPS Designer:** A tool that helps re-programming of the bacterial genome to produce the peptide of interest.

3.2 Materials and Methods

3.2.1 NRP Comprehensive Combinatorial Biosynthesis (CCB)

3.2.1.1 Objective

Combinatorial biosynthesis of novel NRPs has been a longstanding goal in chemical biology. Farag et al. (2019) reported and demonstrated a strong relationship between NRPS IMLs and their adjacent A domains. This finding suggests that, going forward, combinatorial biosynthesis strategies to generate novel NRPs should consider IMLs in addition to other established parameters (Nguyen et al., 2006; Coëffet-Le Gal et al., 2006; Baltz et al., 2006; Crüseemann et al., 2013; Calcott et al., 2014; Meyer et al., 2016; Bozhüyük et al., 2017). NRP-CCB is a tool that allows the generation of large libraries of valid non-ribosomal peptides (NRPs). An NRP is considered valid, if and only if, there is an existing IML for every pair of monomers within that generated peptide (Fig 3.1).

A

Activated Substrate Module 1	Linker	Activated Substrate Module 2
Ala	GAGRARPVLEPWRR	Ser
Ser	LRKGDKKREIPPLIPMER	Leu
Leu	RTGGDSMVTESFSNLLETAPQFAVD	Lys
Lys	AGVQADTAPVIQAVGR	Gly

B

1. Ala-Ser-Leu-Lys-Gly Valid

2. Ala-Ser-Leu-Lys-Phe Invalid

Figure 3.1: **CCB-peptide validity:** (A) A subset of valid IMLs from the IMLs database. (B) An example of two virtually generated peptides, the first one would be considered valid (green) by the CCB algorithm while the second one would be considered invalid (red) due to the lack of an IML between serine and phenylalanine in the IMLs database.

3.2.1.2 CCB algorithm

The theoretical number of possible sequence combinations is computed by n^l , where n refers to the number of unique monomers and l refers to the length of the sequence. Here we used a pre-compiled list of monomers retrieved from the NORINE database (Caboche et al., 2007; Pupin et al., 2016). The list comprised of 539 distinct monomers belonging to various chemical classes including natural & unnatural amino acids, fatty acids and others. Moreover, our sequence generator constructs sequences at different lengths ranging between 2 and 14 monomers. Thus theoretically, the generator could create up to $\sum_{i=2}^{14} 539^i$ distinct peptides sequences, which is a monumental number of peptides. However, this is not the case with CCB as it discards all invalid sequences automatically and retains only the valid ones.

Algorithm 1: CCB Generates a finite set of virtual peptides

Input: A finite set $P = \{p_1, p_2, \dots, p_n\}$ of pairs, A finite set $M = \{m_1, m_2, \dots, m_n\}$ of monomers, an integer l for length of generated peptide, a boolean for *replacement* and a boolean for *genus*

Output: A finite set of virtually generated peptides

- 1 $allPeptides \leftarrow getAllPeptides(replacement, M, l, genus)$
- 2 $putativePeptides \leftarrow getPutativePeptides(allPeptides, P)$
- 3 $writePeptides(putativePeptides)$
- 4 **return** $putativePeptides$

Algorithm 1 generates a finite set of virtual peptides.

Algorithm 2: GETPUTATIVEPEPTIDES retrieves a finite set of putative peptides

Input: A finite set $allPeptides = \{seq_1, seq_2, \dots, seq_n\}$ of all generated peptides, A finite set $P = \{p_1, p_2, \dots, p_n\}$ of pairs

Output: A finite set of putative peptides

- 1 $putativePeptides \leftarrow \emptyset$
- 2 **foreach** $s \in allPeptides$ **do**
- 3 $extractedPairs \leftarrow pariwise(s)$
- 4 **if** $\forall extractedPairs \in P$ **then**
- 5 $putativePeptides \leftarrow putativePeptides + s$
- 6 **return** $putativePeptides$

Algorithm 2 retrieves a finite set of putative peptides, with all their pairs linked with valid IMLs.

3.2.1.3 CCB distinct runs

We ran four versions of the CCB algorithm, for every one of them, we generated peptides of distinct peptide lengths ranging between two and fourteen monomers.

1. **No replacement and No genus:** *No replacement:* Here we ran the tool without allowing the replacement of any of the available monomers during the combinatorial process. *No Genus:* We considered all available pairs of modules across all genera.
2. **No replacement and genus:** *No replacement:* Here we ran the tool without allowing the replacement of any of the available monomers during the combinatorial process. *Genus:* We considered all available pairs of modules within just a specific genus.
3. **Replacement and No genus:** *Replacement:* Here we ran the tool while allowing the replacement of any of the available monomers during the combinatorial process. *No Genus:* We considered all available pairs of modules across all genera.
4. **Replacement and genus:** *Replacement:* Here we ran the tool without allowing the replacement of any of the available monomers during the combinatorial process. *Genus:* We considered all available pairs of modules within just a specific genus.

Finally, we concatenate the outcome of all the runs into a single output data-frame that entails all valid generated NRPs.

3.2.1.4 Code availability

The CCB algorithm source code is hosted in GitHub repository under <https://github.com/SWFarag/CCB>.

3.2.1.5 Notes

1. **Installation:** This script uses Python 3.7.x. If you don't have Python, I would recommend downloading it from [Anaconda](<https://www.continuum.io/downloads>).

Copy or clone this package from Github.

Open the Terminal/Command Line and navigate to where you copied the package:

```
cd path/to/copied/directory
```

2. **Linux and MacOS:** Install the dependencies by entering:

```
pip install -r requirements.txt
```

3. **Usage:** To run from the command-line, just do:

```
python CCB.py
```

Example: Running tool with replacement and with a particular genus

```
python CCB.py -in path_to/IML_genus_db.csv -o path_to_output/CCB/ -l 3 -r 1 -g Bacillus
```

To list all the parameters needed from the command-line, just do:

```
python CCB.py --help
```

4. **Questions and Comments:** Feel free to direct any questions or comments to the Issues page of the repository.
5. **License:** See the LICENSE.md file for license rights and limitations (MIT).

3.2.2 Sequence Based Model

3.2.2.1 Objective

The NRP-CCB algorithm results in the generation of thousands and even millions of virtual peptides. Hence, there is a need for a selection step, that help reducing the number of generated peptides to be conveyed for further steps downstream the pipeline. Here, we propose a binary statistical model based only on peptide sequences to filter out all inactive predicted peptides.

3.2.2.2 Modeling Strategy

Modeling is a multi-step approach, that starts with curating the original data set, preparing the training set, generating descriptors and applying the right machine learning algorithm and finally validating the built models. Figure 3.2. illustrates the complete statistical modeling work-flow (Tropsha, 2010).

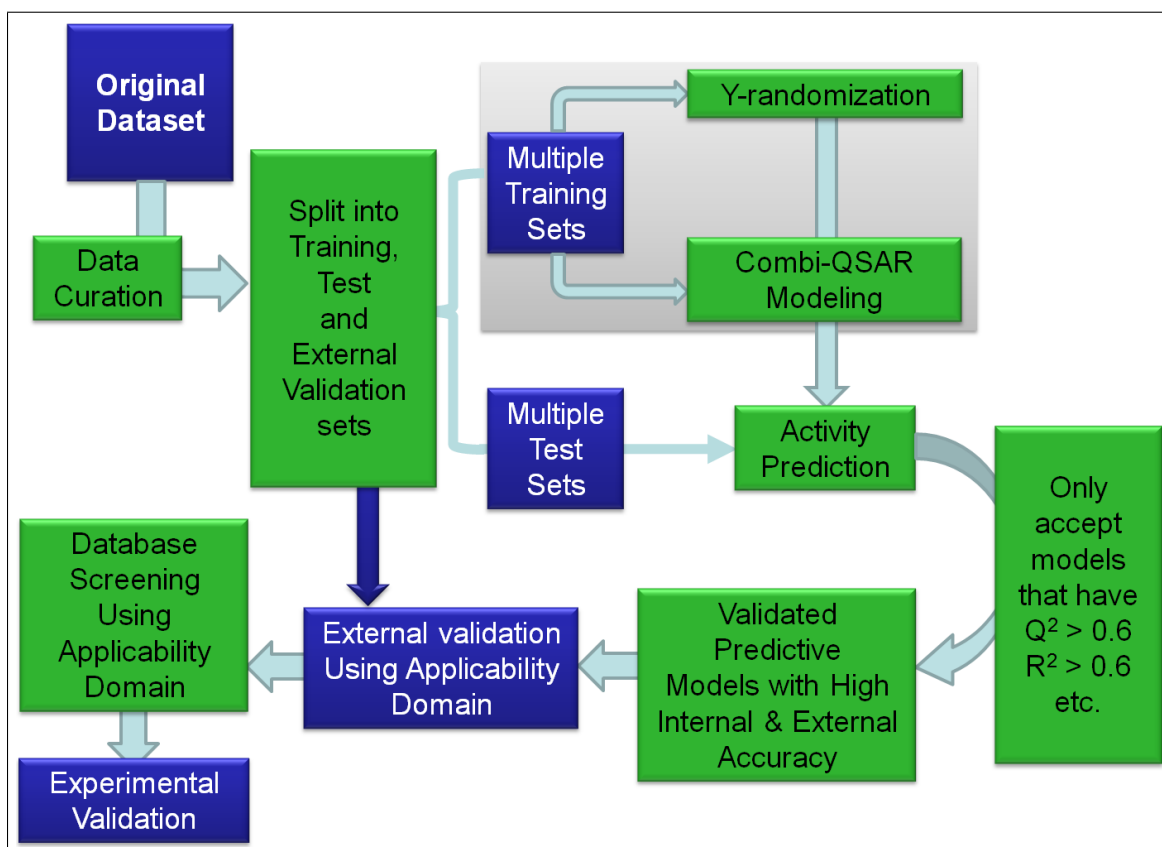


Figure 3.2: Predictive statistical modeling workflow

3.2.2.3 Data Set

- **Training Set:** The source is the NORINE NRPS database (Caboche et al., 2007; Pupin et al., 2016). The database is compiled and annotated thanks to the collaborative effort between the Bonsai bioinformatics research group of CRISAL (Centre de Recherche en Informatique, Signal et Automatique de Lille, ex-LIFL (Laboratoire d'Informatique Fondamentale de Lille) and Inria (Institut National de Recherche en Informatique et en Automatique) and the NRPS team from the ProBioGem laboratory (Laboratoire des Procédés Biologiques Génie Enzymatique et Microbien). It comprises of 1191 non-ribosomal peptides. Only 440 NRPs show antibiotic activities, while the remaining 751 NRPs show other classes of activities such as anti-cancer, immunomodulating, protease inhibitor and siderophore.
- **Prediction Set:** This is the set that would be used for virtual screening. The source of this set is basically the outcome of the CCB algorithm (section 3.2.1.2).

3.2.2.4 End Point

Every statistical model must have a well defined end point prior model building. An end point is the feature that the model is trying to predict. In this step we construct a binary classifier to predict the class activity of the generated peptides. Precisely, we want to predict whether a peptide has anti-bacterial activity or nor. The model is trained with peptides with known anti-bacterial activities and the assumption here is that peptide sequences with similar motifs and sequence patterns would show similar biological activity.

3.2.2.5 Data curation

Data curation is the process of cleaning and correcting the original data, generated by experimental scientist, prior any modeling step. It may include correcting wrong annotations such as peptide names associated with the wrong peptide sequences or a data set full of duplicates that if used as is, will certainly bias the predictive power of the generated model. Last but not least making sure that all peptides are assigned the right label (class activity). We successfully managed to curate our training set and retain all 1191 peptide sequences.

- **Training and test sets:** Here we split our data-set randomly into 80% trainingset and 20% test set. However, the splitting is conducted in a stratified manner, where we ensure that subgroups (group1: actives, group2: inactives) of our original dataset are each adequately represented within the training set and test set.
- **Machine learning algorithms:** Four different machine learning techniques have been applied to develop our models, (a) Logistic Regression (LR) (Pang et al., 2017), (b) Support vector machine (SVM) (Hearst et al., 1998) (c) Random forest (RF) (Breiman, 2001) and (d) Deep neural network (DNN) (LeCun et al., 2015). For the first three we used the python scikit-learn library (Pedregosa et al., 2011) to develop our models, while for the DNN one we used the Keras python library with tensorflow backend (Chollet et al., 2015; Abadi et al., 2015, 2016).
- **5 folds cross-validation:** This is the process where the training set is subjected to 5-folds internal cross-validation procedure as detailed in (Tropsha, 2010). Basically, our modeling set was partitioned into 5 subsets of similar size. Models were then independently developed such that peptides in 4 of the 5 subsets were used as the modeling set and peptides in the remaining subset were used as the evaluation set.
- **Y-Randomization:** This is the process of re-training the models, however, after we first shuffle the y-labels. Basically, we randomly assign labels to our training set. The rationale behind this step is to ensure the statistical significance of our originally trained model. At the end, we compare the outcome of the original model with that of the Y-randomized one, if both models reveals similar results, that is an indication, that the original model has failed in finding a true statistical significant correlation between the data-points and their labels and that the outcome is just based on mere chance. However, if the Y-randomized model resulted in a much worse outcome compared to the original one, then that is a good sign of statistical significance.
- **External validation:** Here, we use our trained models to predict the outcome of a hidden test set with known labels.

3.2.2.8 Code availability

The sequence based model source code is hosted in GitHub repository under <https://github.com/SWFarag/NRP-structure-classifier>.

3.2.2.9 Notes

1. **Installation:** This script uses Python 3.7.x. If you don't have Python, I would recommend downloading it from [Anaconda](<https://www.continuum.io/downloads>).

Copy or clone this package from Github.

Open the Terminal/Command Line and navigate to where you copied the package:

```
cd path/to/copied/directory
```

2. **Linux and MacOS:** Install the dependencies by entering:

```
pip install -r requirements.txt
```

3. **Usage:** To run conventional machine learning models from the command-line, just do:

```
python conventional_models.py
```

Example: Running tool with `model_type=0` [Categorical model]

```
python deep_learning_models.py -in path_to/sequences.csv -o path_to_output/ -mt 0
```

To run deep learning models from the command-line, just do:

```
python deep_learning_models.py
```

Example: Running tool with `max_length=50` and `embedding_length=32`

```
python deep_learning_models.py -in path_to/sequences.csv -o path_to_output/ -ml 50 -el 32
```

To list all the parameters needed from the command-line, just do:

```
python conventional_models.py --help python deep_learning_models.py --help
```

4. **Questions and Comments:** Feel free to direct any questions or comments to the Issues page of the repository.
5. **License:** See the LICENSE.md file for license rights and limitations (MIT).

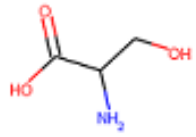
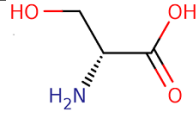
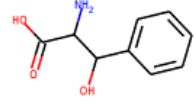
3.2.3 Pep2Struc

3.2.3.1 Objective

Pep2Struc tool that helps converting peptide-sequences to their 2D structures in the form of **SMILES** which stands for 'Simplified *Molecular Input Line Entry System*'. SMILES is a specification in form of a line notation for describing the structure of chemical molecules using short ASCII strings (Weininger, 1988). The tool entails of five steps described as follows:

1. **Compiling and curating monomers:** The NORINE database (Weininger, 1988) hosts a pre-compiled list of all possible monomers that are known to be involved in the synthesis of non-ribosomal peptides (NRPs). The list comprises of 539 distinct monomers belonging to distinct chemical classes including natural & unnatural amino acids and fatty acids. After gathering our list of monomers, we iterate through all of them and made sure that they have the correct smile annotations. Table 3.1 shows a subset of the compiled list demonstrating that monomers includes not only natural amino acids but also unnatural ones such as the D-isofom as well as β -substituent.

Table 3.1: A Subset of Monomers

Name	Molecular Formula	Smile	Code	Figure
Serine	C ₃ H ₇ NO ₃	NC(CO)C(=O)O	Ser	
D-Serine	C ₃ H ₇ NO ₃	NC(CO)C(=O)O	D-Ser	
Phenylserine	C ₉ H ₁₁ NO ₃	NC(C(=O)O)C(O)c1ccccc1	Ph-Ser	

2. **Inter-chemical reactions:** Here, we define and rank a list of possible and desired chemical reactions to take place between any pair of monomers within a non-ribosomal peptide sequence. For instance, since, the vast majority of our monomers are amino acids with few fatty acids, hence a condensation reaction (Fig 3.3), is ranked first in our list followed by an esterification reaction (Fig 3.4).

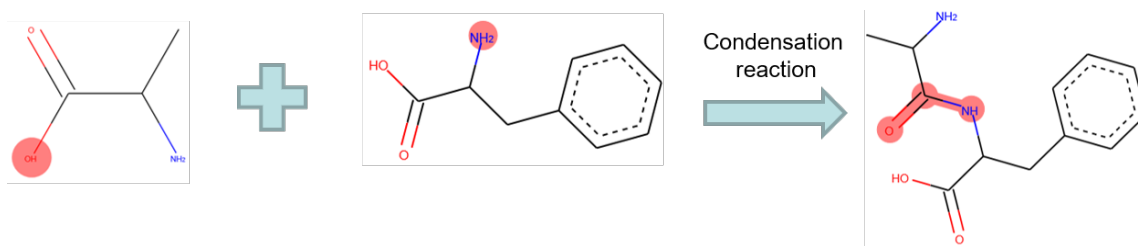


Figure 3.3: **Condensation reaction:** A chemical reaction between alanine and phenylalanine amino acids, that will result in the expulsion a of a water molecule and formation of a peptide bond (amide)

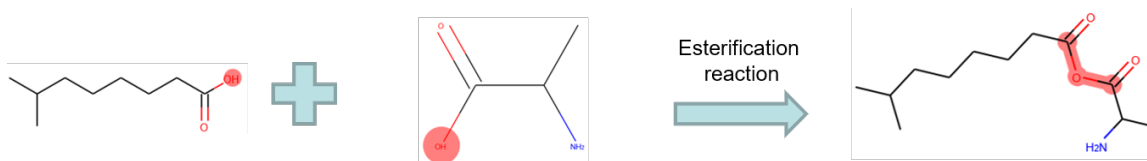


Figure 3.4: **Esterification reaction:** A chemical reaction between 7-methyloctanoic acid and alanine amino acid, that will result in the expulsion a of a water molecule and formation of an ester bond.

In order to define a chemical reaction computationally, we used SMIRKS reaction language and the RDKit python library (Editor RDKit, 2013). SMIRK is a Reaction Transform Language (Editor Daylight, 2013) which is defined for generic reactions. It is a hybrid of SMILES (Weininger, 1988) and SMARTS (Editor Daylight, 2012), in order to meet the dual needs for a generic reaction: expression of a reaction graph and expression of indirect effects. It is a restricted version of reaction SMARTS involving changes in atom-bond patterns. Here are the rules for SMIRKS as stated by “Daylight Chemical Information System”:

- “The reactant and product sides of the transformation are required to have the same numbers and types of mapped atoms and the atom maps must be pairwise. However, non-mapped atoms may be added or deleted during a transformation”.
- “Stoichiometry is defined to be 1-1 for all atoms in the reactant and product for a transformation. Hence, if non-unit stoichiometry is desired, reactants or products must be repeated”.
- “Explicit hydrogens that are used on one side of a transformation must appear explicitly on the other side of the transformation and must be mapped”.
- “Bond expressions must be valid SMILES (no bond queries allowed)”.

- “Atomic expressions may be any valid atomic SMARTS expression for nodes where the bonding (connectivity & bond order) doesn’t change. Otherwise, the atomic expressions must be valid SMILES”.

3. **Intra-chemical reactions:** Oftentimes, linear peptides undergo a macro-cyclization step. Thus, in order to integrate such a chemical step computationally, we defined and ranked a list of possible and desired chemical reactions to take place within the structure of a single molecule. Here, we used an edited version of the same SMIRKS reactions defined in 2. The cyclization step could be classified into one of the following three categories:

- **Head and tail cyclization:** This is the case when the cyclization occurs between the last monomer (tail) and the first one (head) and it is known to be the most common sort of cyclization to occur. Figure 3.5 (A) shows an example of a head to tail cyclization.
- **Partial cyclization:** This is the case when the cyclization occurs between the last monomer (tail) and any monomer but the first one. Figure 3.5 (B) shows an example of a partial-cyclization.
- **Double cyclization:** This is the case when the cyclization occurs either between any two monomers’ backbones or between a monomer backbone and a side chain. Figure 3.5 (C) shows an example of a double-cyclization.

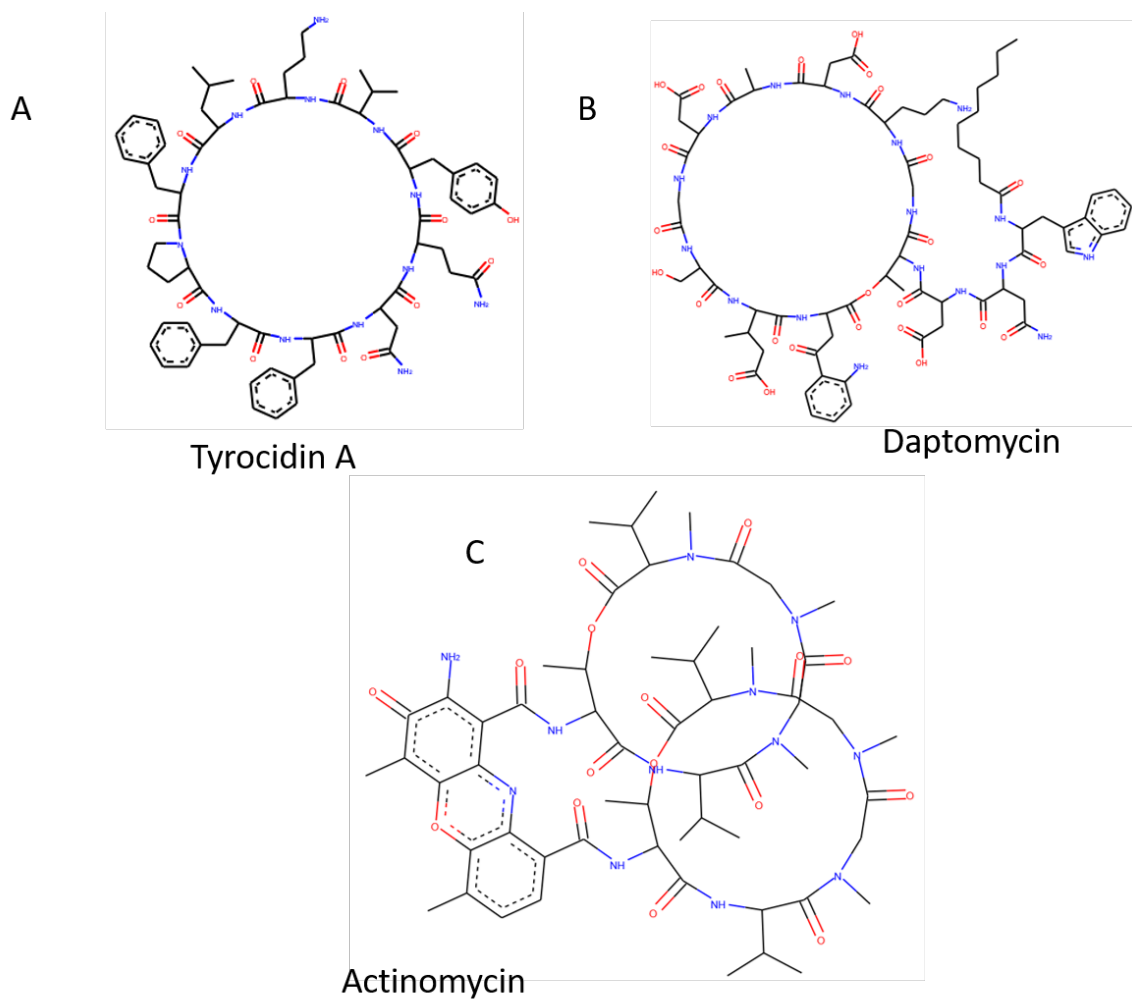


Figure 3.5: **Examples of NRPs with different cyclizations:** (A) Tyrocidin represents a head to tail cyclization. (B) Daptomycin represents a partial cyclization. (C) Actinomycin represents a double cyclization.

4. **Protection:** Generally, when working with reaction informatics, it is difficult to express a reaction exactly enough to not end up with extraneous products. Thus, atoms of a molecule, that are not supposed to take place in a chemical reaction, should be masked and protected, which in turn will help reducing the number of unwanted products. Fig 3.6 (A) illustrates a condensation reaction between an acid and a base, where all their atoms are unprotected. Meaning, the hydroxyl group from the acid could interact with either one of the two available amines in the base molecule. That would eventually lead to the creation of two products instead of one [a desired one and an undesired one]. Fig 3.6 (B) Shows the exact same condensation reaction between an acid and a base, however this time the secondary amine group in the base molecule is protected, which will lead to the creation of just a single product [desired one].

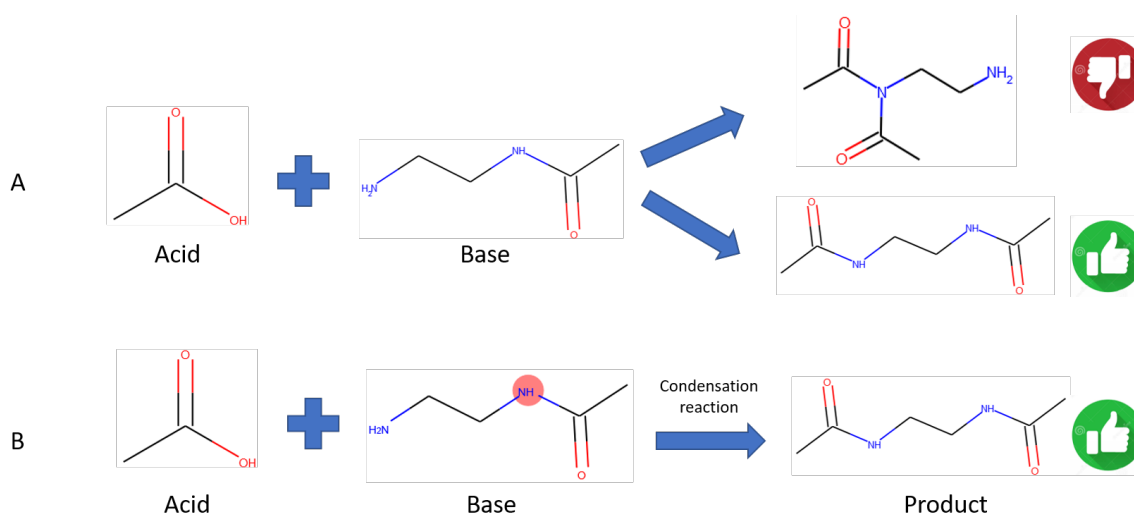


Figure 3.6: **Protection:** (A) A condensation reaction between an acid and a base, where all their atoms are unprotected, leading to the creation of two products instead of one [a desired one and an undesired one]. (B) A condensation reaction between an acid and a base, however this time the secondary amine group in the base molecule is protected, which will then lead to the creation of just a single product [desired one].

5. **Execution rules:** After we have defined our list of inter and intra chemical reactions, there is a need to set some execution rules. Those rules must be respected while running the algorithm. For instance, we need to specify which chemical reaction to be conducted first, the esterification or condensation reaction? which functional groups should be protected throughout the reaction and which should be kept unprotected?

3.2.3.2 Pep2Struc algorithm

Here, we depict the chain of subroutines needed to successfully convert peptides sequences to their 2D structure.

Algorithm 3: PEP2STRUC Generates a finite set of smiles.

Input: A finite set $P = \{p_1, p_2, \dots, p_n\}$ of peptides, A dictionary $M2SmilesMap = \{mo_1, mo_2, \dots, mo_n\}$ which maps monomers to their curated smile string, A reaction type $reaction_type \in R = \{linear, cyclic, partial_cyclic\}$

Output: A finite set of generated smiles

- 1 $convertedPeptides \leftarrow \{\}$
- 2 **foreach** $p \in peptides$ **do**
- 3 $new_smiles \leftarrow convertPeptide(p, M2SmilesMap, reaction_type)$
- 4 $convertedPeptides \leftarrow convertedPeptides + new_smiles$

Algorithm 3 Generates a finite set of converted peptides as smiles.

Algorithm 4: CONVERTPEPTIDE converts a sequence of peptide into its 2D structure.

Input: A single peptide $peptide$, A dictionary $M2SmilesMap = \{mo_1, mo_2, \dots, mo_n\}$ which maps monomers to their curated smile string, A reaction type $reaction_type \in R = \{linear, cyclic\}$

Output: A list of 2D structure(s) in the form of smile(s)

- 1 $new_smiles \leftarrow \{\}$
- 2 $intermediate \leftarrow \{\}$
- 3 $intermediate \leftarrow intermediate + M2SmilesMap.get(p[0])$
- 4 **for** $i \leftarrow 1$ **to** $p.length()$ **do**
- 5 $reactant1 \leftarrow intermediate.getLastelement()$
- 6 $reactant2 \leftarrow M2SmilesMap.get(p[i])$
- 7 $new_smiles \leftarrow run_Reaction(reactant1, reactant2, reaction_type)$

Algorithm 4 converts a sequence of peptide into its 2D structure in the form of a smile.

Algorithm 5: RUN_REACTION

Input: A monomer smile *reactant1*, A monomer smile *reactant2*, A reaction type $reaction_type \in R = \{linear, cyclic\}$

Output: A list of 2D structure(s) in the form of smile(s)

```
1 if reaction_type == linear then
2   | reactant1_protected = protect_atoms(reactant1)
3   | reactant2_protected = protect_atoms(reactant2)
4   | find_pattern(reactant1_protected)
5   | find_pattern(reactant2_protected)
6   | linear_Reaction(reactant1_protected, reactant2_protected)
7 else
8   | reactant1_protected = protect_atoms(reactant1)
9   | reactant2_protected = protect_atoms(reactant2)
10  | find_pattern(reactant1_protected)
11  | find_pattern(reactant2_protected)
12  | cyclic_Reaction(reactant1, reactant2)
```

Algorithm 6: PROTECT_ATOMS

Input: A molecule *molecule*

Output: a molecules where all its atoms are protected.

```
1 foreach atom ∈ peptides.getAtoms() do
2   | a.protect()
```

Algorithm 5 and 6 protect all atoms and run the chemical reaction.

Algorithm 7: FIND_PATTERN

Input: A molecule *molecule_protected* with all its atoms are protected, A SMART pattern *smart_pattern*

Output: a molecules where all its atoms are protected.

```
1 pattern_matches ← {}
2 pattern_matches ← molecule.match_pattern(smart_pattern)
3 foreach match ∈ pattern_matches do
4   | foreach atom ∈ match do
5     | a.unprotect()
```

Algorithm 7 search for structural patterns to unprotect and to render susceptible for reaction.

Algorithm 8: LINEAR_REACTION

Input: A monomer smile *reactant1*, A monomer smile *reactant2*, A reaction type
reaction_type = linear

Output: A list of 2D structure(s) in the form of smile(s)

- 1 *amide_inter* \leftarrow *SMIRK_RXN_INTER_1*
- 2 *ester_inter* \leftarrow *SMIRK_RXN_INTER_2*
- 3 *aromatic_condensation_inter* \leftarrow *SMIRK_RXN_INTER_3*
- 4 *amine_inter* \leftarrow *SMIRK_RXN_INTER_4*
- 5
- 6 *product* = *amide_inter*(*reactant1*, *reactant2*)
- 7 **if** *product* not empty **then**
- 8 | return *product*
- 9 *product* = *ester_inter*(*reactant1*, *reactant2*)
- 10 **if** *product* not empty **then**
- 11 | return *product*
- 12 *product* = *aromatic_condensation_inter*(*reactant1*, *reactant2*)
- 13 **if** *product* not empty **then**
- 14 | return *product*
- 15 *product* = *amine_inter*(*reactant1*, *reactant2*)
- 16 **if** *product* not empty **then**
- 17 | return *product*
- 18 **else**
- 19 | return "No_reaction"

Algorithm 9: CYCLIC_REACTION

Input: A monomer smile *reactant1*, A monomer smile *reactant2*, A reaction type
reaction_type = cyclic

Output: A list of 2D structure(s) in the form of smile(s)

- 1 *amide_intra* \leftarrow *SMIRK_RXN_INTRA_1*
- 2 *ester_intra* \leftarrow *SMIRK_RXN_INTRA_2*
- 3
- 4 *product* = *amide_intra*(*reactant1*, *reactant2*)
- 5 **if** *product* not empty **then**
- 6 | return *product*
- 7 *product* = *ester_intra*(*reactant1*, *reactant2*)
- 8 **if** *product* not empty **then**
- 9 | return *product*
- 10 **else**
- 11 | return "No_reaction"

Algorithms 8 and 9 depict the two types of reactions namely, linear or cyclic.

3.2.3.3 Pep2Struc VS CycloPs

CycloPs (Duffy et al., 2011) is a computational approach for generating virtual libraries of cyclized and constrained peptides. At the first glance CycloPs seems to conduct the exact same functionality as our Pep2struc tool. However, here are the main differences between the two:

- **Unnatural amino acids:** Cyclops can only handle natural amino acids and D-amino acids, while our tool can deal with any sort of generic amino acids including the β -substituent amino acids.
- **Flexibility:** Unlike CycloPs, Pep2struc shows more flexibility as it allows its users to upload a list of newly defined monomers. This feature helps increasing the chemical diversity of the generated peptides.
- **Defined chemical reactions:** CycloPs deals only with a single type of chemical reaction, namely the condensation reaction, which takes place between two amino acids. CycloPs undergoes smile manipulation rather than defining a chemical reaction. Hence, it requires that smiles of both reactants to be annotated in the exact same way. Moreover, even if it was possible for CycloPs to upload a new monomer, and the uploaded monomer happened to be differently annotated, then this would cause CycloPs to either throw an error or to create a false and an undesired product as illustrated in (Fig 3.7 A). On the other hand, our tool, doesn't depend on how the smile of a monomer is annotated, as any annotation will work, as long as it reflects the correct structure of the molecule. The reason behind that, is due the nature of our algorithm, as it searches for a specific pattern to conduct a chemical reaction, regardless of smile annotation rather than just conducting smile manipulation (Fig 3.7 B)

The limitations of Pep2struc tool:

- **Updating chemical reactions list:** New building blocks from natural products are being constantly identified and some of them don't even have amino acids characteristics, hence, there is a need to regularly update our list of reactions to ensure an adequate chemical reaction between the former as well as the newly added monomers.

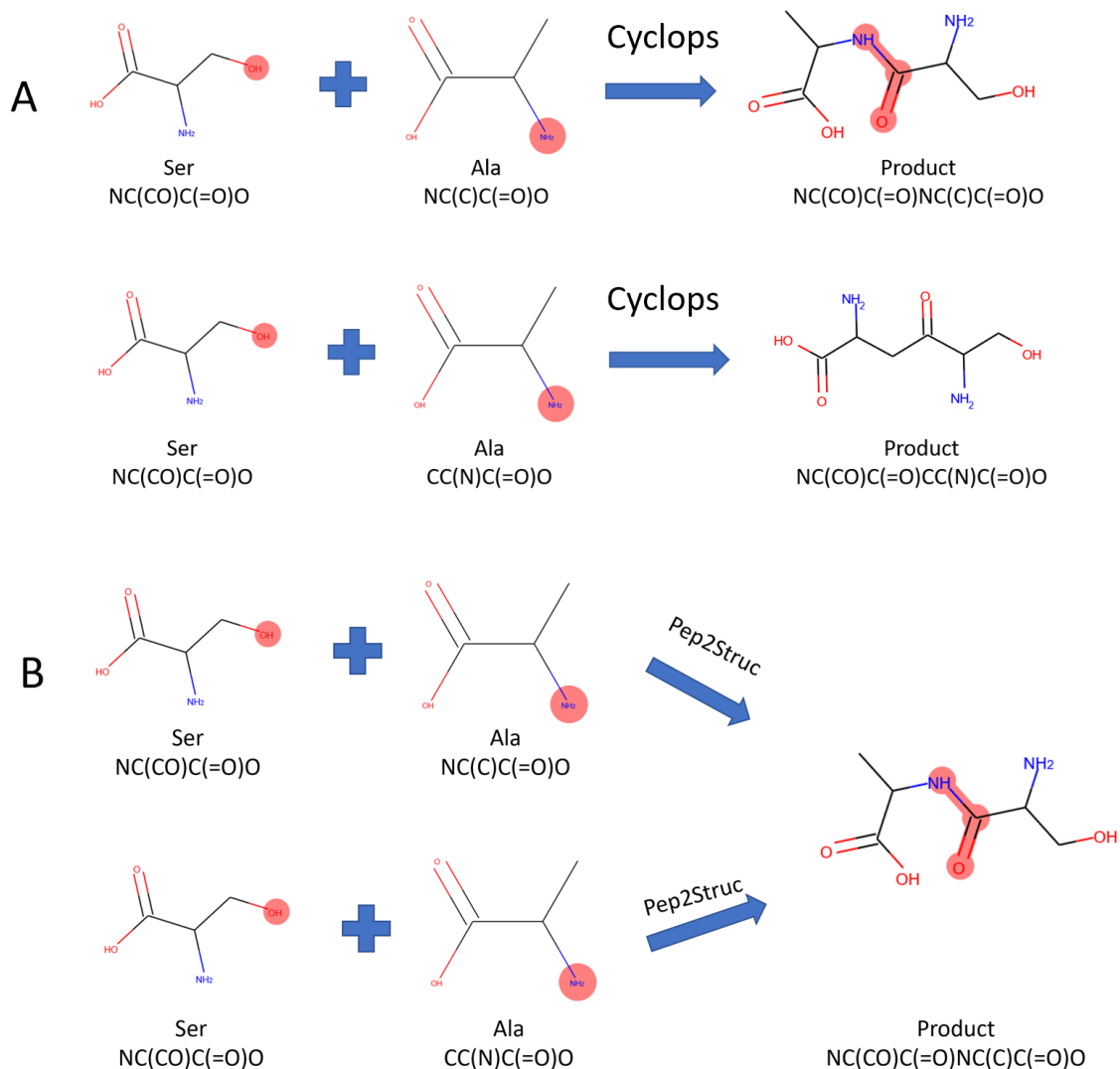


Figure 3.7: **Pep2struc vs. CycloPs:** (A) Two condensation reactions conducted by cyclop, using two different smile annotations for the alanine molecule, that leads to the creation of two distinct products. (B) Two condensation reactions conducted by Pep2struc, using two different smile annotations for the alanine molecule, that leads to the creation of just a single product.

- **Protection:** This is a crucial step for our tool as if not conducted correctly, it would either lead to some undesired extraneous products or even worse hinder the production of the right ones (Fig 3.3)

3.2.3.4 Code availability

The Pep2struc source code is hosted in GitHub repository under <https://github.com/SWFarag/pep2struc>.

3.2.3.5 Notes

1. **Installation:** This script uses Python 3.7.x. If you don't have Python, I would recommend downloading it from [Anaconda](<https://www.continuum.io/downloads>).

Copy or clone this package from Github.

Open the Terminal/Command Line and navigate to where you copied the package:

```
cd path/to/copied/directory
```

2. **Linux and MacOS:** Install the dependencies by entering:

```
pip install -r requirements.txt
```

3. **Usage:**

To run pep2struc from the command-line, just do:

```
python pep2struc.py
```

Example: Running tool with model_type=0 [Categorical model]

```
python pep2struc.py -in path_to/sequences.csv -o path_to_output/ -t linear
```

To list all the parameters needed from the command-line, just do:

```
python pep2struc.py --help
```

4. **Questions and Comments:** Feel free to direct any questions or comments to the Issues page of the repository.
5. **License:** See the LICENSE.md file for license rights and limitations (MIT).

3.2.4 Structure Based Model

3.2.4.1 Objective

The pep2struc tool will result in the conversion of thousands of peptide sequences into their 2D structures. Hence, with so many virtually synthesized macrocycles there is still a need for another aggressive triaging step. Therefore, we are developing a 'Quantitative Structure Activity Relationship'(QSAR) model (Tropsha, 2010; Cherkasov et al., 2014). This is another classifier, however, this time it is based on peptide 2D structures rather than just their sequences. The developed models will help filtering out all inactive predicted macrocycles.

3.2.4.2 Modeling Strategy

Modeling is a multi-step approach, that starts with curating the original data set, preparing the training set, generating descriptors and applying the right machine learning algorithm and finally validating the built models. Figure 3.8 is a good illustration of predictive QSAR modeling work-flow that focuses on delivering highly predictive and vigorously validated models. Ultimately, all potential computational hits would be then confirmed by the experimental validation step (Tropsha, 2010).

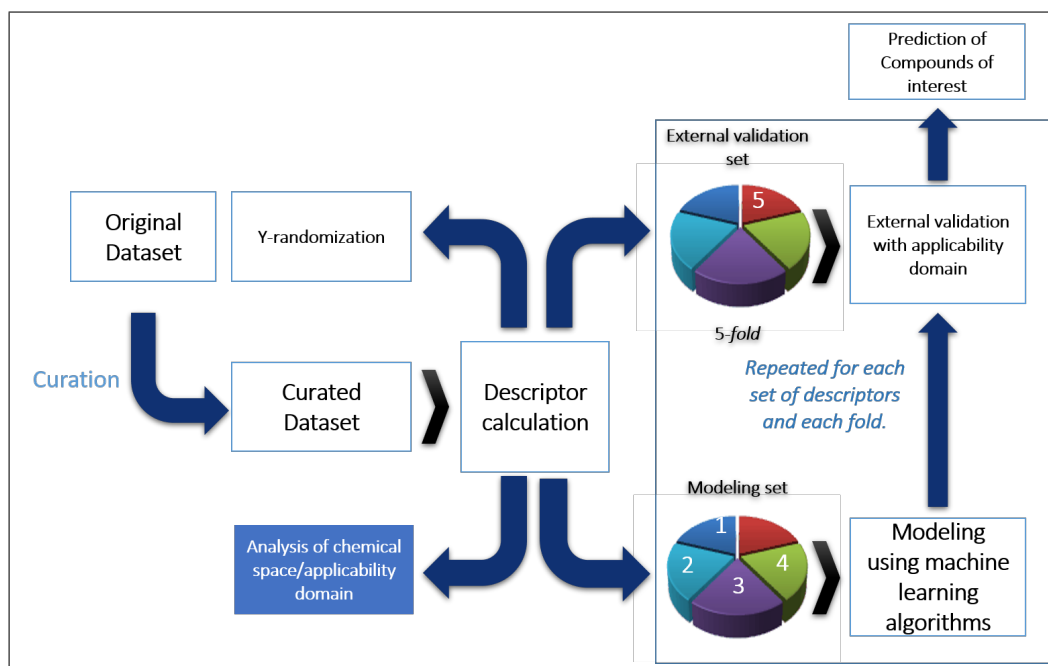


Figure 3.8: Predictive QSAR modeling workflow.

3.2.4.3 Data Set

We used the same data source as in section 3.2.2.3, which is the 1191 NRPs (Caboche et al., 2007; Pupin et al., 2016) from the NORINE database.

- **Training Set:** Since we are trying to build QSAR models, our training set should entail peptides with known structures and well determined activities. In the original data-set only 440 NRPs show antibiotic activities while the remaining 751 NRPs don't. However, only 180 and 380 NRPs out of the 440 and 751, have well annotated structures, respectively. This leaves us with a training set of 560 NRPs.
- **prediction Set:** Our prediction set comprises of thousands of virtually synthesized peptide structures which is basically the outcome of the pep2struc algorithm.

3.2.4.4 End Point

We used the same end point mentioned in section 3.2.2.4. The model is trained with NRPs with known chemical structures and anti-bacterial activities. The hypothesis here is that NRPS with similar chemical structure would show similar biological activity.

3.2.4.5 Data curation

Despite the importance and the necessity of data curation prior model development, there appear to be no commonly accepted guidelines or set of procedures for chemical data curation. This was the case till 2010 when Fourches, et al. published '**Trust, but verify: On the importance of chemical structure curation in cheminformatics and QSAR modeling research**' (Fourches et al., 2010). The paper emphasizes on the importance of data curation for molecular modeling and suggests the very first standardized chemical data curation strategy.

Assuming the input data is a list of canonical **SMILES**, data curation could be broken down into two main steps: (a) Cleaning step, which includes removal of broken smiles, inorganics and mixtures (Patterson et al., 2003). (b) Standardization and harmonization step, which involves normalization of specific chemo-types, treatment of tautomeric forms and removal of duplicates (Young et al., 2008). The last step of molecular curation entails manual inspection of every structure.

Obviously, this step is very tedious and time consuming and hence, it is only recommended for small data sets (Few hundreds compounds). These guidelines resonated well at the cheminformatics community and thus it was adopted by some major cheminformatics software providers such as ChemAxon (Editor ChemAxon, 2011), Molecular Operating environment (MOE) (Chemical Computing group, 2010), and OpenBabel (O'Boyle et al., 2008). For more details, please refer to (Fourches et al., 2010; Young et al., 2008).

Training Set curation: As previously stated prior any modeling step, the chemical structures of the training set must undergo a curation step in order to prepare a QSAR-ready data set. After applying the above mentioned procedure on our training set which comprises of 560 compounds, we successfully managed to curate and retain all 560 compounds.

Prediction set curation: A chemical data curation step is not only required prior QSAR modeling step but also prior any virtual screening step. Thus we applied the same procedure on our prediction set which comprise of ~39 thousand compounds.

3.2.4.6 Descriptors

A molecular descriptor is defined as the “final result of a logical and mathematical procedure, which transforms chemical information encoded within a symbolic representation of a molecule into a useful number or the result of some standardized experiment” (Todeschini and Consonni, 2000). In this study we used multiple types of descriptors as listed below:

- **Holistic chemical descriptors:** Molecular descriptors (represented with explicit hydrogen atoms) were computed for each compound using Dragon software (version 5.5) (Todeschini and Consonni, 2000). Descriptors with low variance or missing values were removed. Furthermore, if the squared correlation coefficient (R^2) between values of two descriptors over the entire data set exceeded 0.95, one of the descriptors was randomly removed. The final descriptor set used in this study contained 316 descriptors ranged-scaled to the [0, 1] interval.
- **Fingerprints descriptors:** Fingerprint representations of molecular structure and properties are a particularly complex form of descriptors. Fingerprints are typically encoded as binary bit strings whose settings produce, in different ways, a bit “pattern” characteristic of a given molecule (O'Boyle and Sayle, 2016). Fingerprints are designed to account for different sets of

molecular descriptors, structural fragments, possible connectivity pathways through a molecule, or different types of pharmacophores. Here we used the following fingerprints:

- **Atom Pair Fingerprint:** generates the atom-pair fingerprint for a molecule as an explicit bit vector (Carhart et al., 1985; Awale and Reymond, 2014).
- **Topological-torsion fingerprint:** is generated by exhaustively enumerating all linear fragments of a molecular graph up to a given size and then hashing these fragments into a fixed-length bitvector (Nilakantan et al., 1987).
- **MACCS Keys Fingerprint:** MACCS keys are 166 bit structural key descriptors in which each bit is associated with a SMARTS pattern. basically, The MACCS keys are a set of questions about a chemical structure. Here are some of the questions: Are there fewer than 3 oxygens? Is there a S-S bond? Is there a ring of size 4? Is there at least one F, Cl, Br, or I present? The result of this is a list of binary values either true 1 or false 0. This list of values for a given chemical structure is called the MACCS key fingerprint for that structure. (Durant et al., 2002).

3.2.4.7 Modeling approaches

Here, we discuss in details the needed steps towards building a predictive QSAR model and how to evaluate it.

- **Training and test sets:** Here we split our data-set randomly into 80% trainingset and 20% test set. However, the splitting is conducted in a stratified manner, where we ensure that subgroups (group1: actives, group2: inactives) of our original dataset are each adequately represented within the training set and test set.
- **Machine learning algorithms:** Three different machine learning techniques have been applied to develop our models, (a) Logistic Regression, (b) SVM, and (c) Random forest. For model development we used the python scikit-learn library.
- **5 folds cross-validation:** This is the process were the training set is subjected to 5-folds internal cross-validation procedure as detailed in (Tropsha, 2010). Basically, our modeling set was partitioned into 5 subsets of similar size. Models were then independently developed such

that peptides in 4 of the 5 subsets were used as the modeling set and peptides in the remaining subset were used as the evaluation set.

- **Y-Randomization:** This is the process of re-training the models, however, after we first shuffle the y-labels. Basically, we randomly assign labels to our training set. The rationale behind this step is to ensure the statistical significance of our originally trained model. At the end, we compare the outcome of the original model with that of the Y-randomized one, if both models reveals similar results, that is an indication, that the original model has failed in finding a true statistical significant correlation between the data-points and their labels and that the outcome is just based on mere chance. However, if the Y-randomized model resulted in a much worse outcome compared to the original one, then that is a good sign of statistical significance.
- **External validation:** Here, we use our trained models to predict the outcome of a hidden test set with known labels.

3.2.4.8 Code availability

The structure based model source code is hosted in GitHub repository under <https://github.com/SWFarag/NRP-structure-classifier>.

3.2.4.9 Notes

1. **Installation:** This script uses Python 3.7.x. If you don't have Python, I would recommend downloading it from [Anaconda](<https://www.continuum.io/downloads>).

Copy or clone this package from Github.

Open the Terminal/Command Line and navigate to where you copied the package:

```
cd path/to/copied/directory
```

2. **Linux and MacOS:** Install the dependencies by entering:

```
pip install -r requirements.txt
```

3. **Usage:** To run conventional machine learning models from the command-line, just do:

```
python sb_models.py
```

Example: Running tool with `model_type=0` [Categorical model]

```
ppython sb_models.py -in path_to/sequences.csv -o path_to_output/outputFolderName/ -mt 0
```

To list all the parameters needed from the command-line, just do:

```
python sb_models.py --help
```

4. **Questions and Comments:** Feel free to direct any questions or comments to the Issues page of the repository.
5. **License:** See the LICENSE.md file for license rights and limitations (MIT).

3.2.5 NRPS Designer

3.2.5.1 Objective

This tool is a NRP bio-synthetic gene cluster (BGC) editor. It has two main functionalities: (1) it edits an existing NRPS-BGC, (2) it creates a whole new one from scratch. All edits are conducted at the nucleotides level within the bacterial genome, which leads to a construct that will eventually be cloned and transformed into a bacteria to produce the peptide of interest (Fig 3.9).

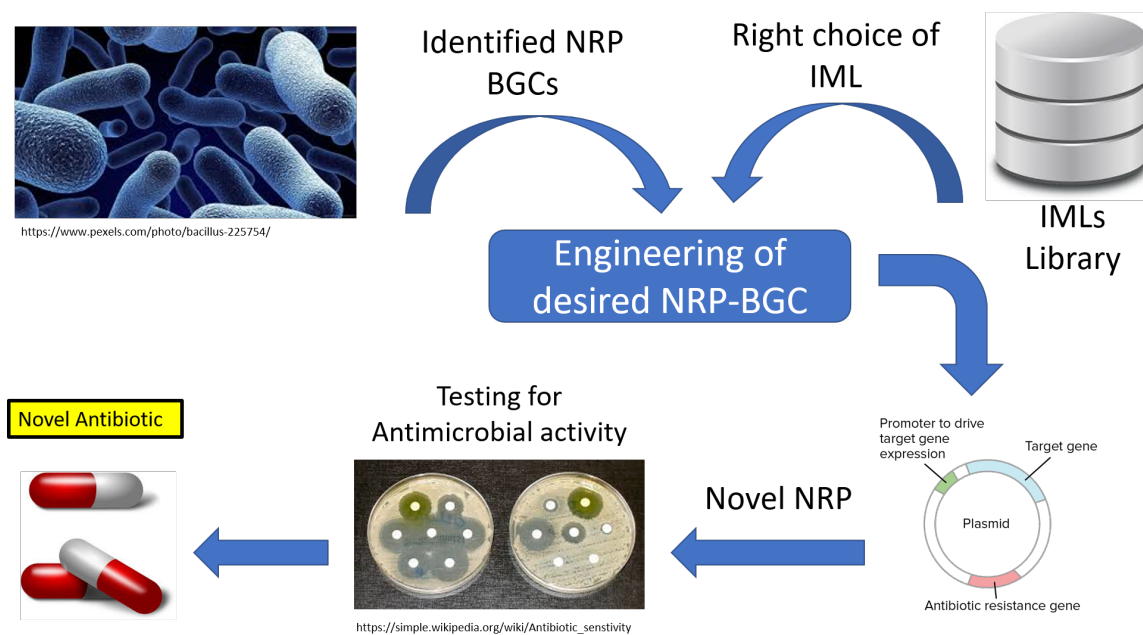


Figure 3.9: **NRPS-designer**: A scheme showing the role and impact of the NRPS-designer tool.

3.2.5.2 NRPS Designer algorithm

As mentioned above the algorithm works in two modes: (1) Template Based and (2) From Scratch. The former refers to an algorithm that uses an existing NRPS-BGC as a template and undergoes few edits to it. For instance Figure 3.10 illustrates the use of tyrocidin BGC as a template and performs a couple of edits to its Tyc NRPS subunit. The edits involve exchanging two modules, namely module five and module eight that activate asparagine and valine, respectively with two modules that activate lysine and glycine, respectively. Since, we are dealing with a terminal module and a non-terminal one, which will then cause the rise of three incompatible inter-modular linkers. Thus, we use the IMLs database to provide us with the corresponding required IMLs to finish editing the template.

The latter denotes to an algorithm that help building a novel NRPS BGC without the use of any templates.

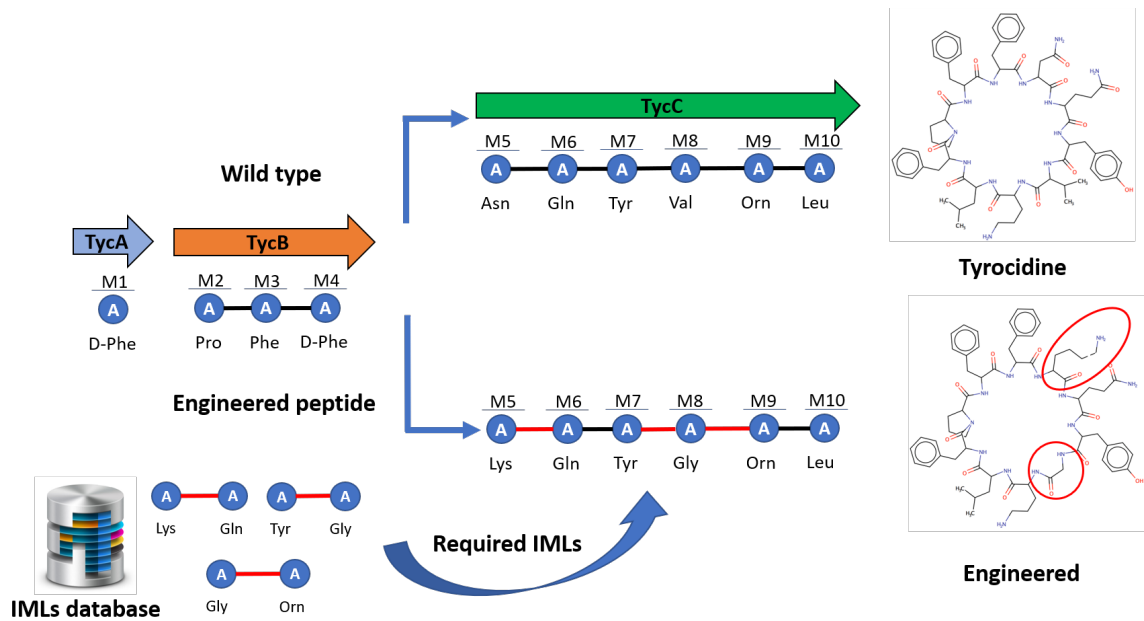


Figure 3.10: **NRPS-designer template based:** A scheme showing the usage of tyrocidin BGC as a template and applying couple of edits to its Tyc NRPS enzyme, namely exchanging module 5 on TycC that activates alanine to a module that activate lysine. Moreover, exchanging module 8 which activates valine to a module that activates glycine. In addition to exchanging modules also the corresponding inter-modular linkers are also add to the mix, denoted in red lines.

- **NRPS Designer Template Based:**

Algorithm 10: NRPS_DESIGNER_TEMPLATE Create an edited NRPS cluster.

Input: A template peptide sequence *templateSeq*, A edited peptide sequence *editedSeq*, template peptide name *templateName*, species name *speciesName*

Output: An edited NRP BGC

```

1 if IsValidSequence(editedSeq, speciesName) and IsActive(editedSeq) then
2   newModulesMap ←
   getNewModules(templateSeq, editedSeq, templatePeptideName, speciesName)
3   novel_NRPS ← buildNovelNRPS(newModulesMap)
4   write_NRPS_FASTA(novel_NRPS)
5   return novel_NRPS
6 else
7   return "bad_sequence"

```

- **NRPS Designer From Scratch:**

Algorithm 11: NRPS_DESIGNER_SCRATCH Create a novel NRPS cluster from scratch

Input: A edited peptide sequence $editedSeq$, species name $speciesName$
Output: A novel NRP BGC

```

1 if  $IsValidSequence(editedSeq, speciesName)$  and  $IsActive(editedSeq)$  then
2    $newModulesMap \leftarrow getNewModules(editedSeq, speciesName)$ 
3    $novel\_NRPS \leftarrow buildNovelNRPS(newModulesMap)$ 
4    $write\_NRPS\_FASTA(novel\_NRPS)$ 
5   return  $novel\_NRPS$ 
6 else
7   return "bad_sequence"

```

Algorithm 12: BUILDNOVELNRPS concatenates a set of modules while including their corresponding inter-modular linkers.

Input: A map of $newModulesMap = \{M_1 : [linker_{before}, linker_{after}], M_2 : [linker_{before}, linker_{after}], \dots, M_n : [linker_{before}, linker_{after}]\}$
Output: A novel NRP BGC

```

1  $novel\_NRPS\_cluster \leftarrow \emptyset$ 
2 foreach  $key \in newModulesMap.keys()$  do
3   if not last key then
4      $novel\_NRPS\_cluster \leftarrow key + newModulesMap.get(key)[1]$ 
5   else
6      $novel\_NRPS\_cluster \leftarrow key + newModulesMap.get(key)$ 
7 return  $novel\_NRPS\_cluster$ 

```

3.2.5.3 Code availability

The NRPS Designer source code is hosted in GitHub repository under <https://github.com/SWFarag/NRPS-designer>.

3.2.5.4 Notes

1. **Installation:** This script uses Python 2.7.x. If you don't have Python, I would recommend downloading it from [Anaconda](<https://www.continuum.io/downloads>).

Copy or clone this package from Github.

Open the Terminal/Command Line and navigate to where you copied the package:

```
cd path/to/copied/directory
```

2. **Linux and MacOS:** Install the dependencies by entering:

```
pip install -r requirements.txt
```

3. **Usage:** To run from the command-line, just do:

```
python wsgi.py
```

4. **Questions and Comments:** Feel free to direct any questions or comments to the Issues page of the repository.

5. **License:** See the LICENSE.md file for license rights and limitations (MIT).

3.3 Results

3.3.1 Model evaluation

3.3.1.1 Five fold cross validation(5-FCV)

Before discussing the result of the 5 fold cross validation mentioned in section 3.2.2.7, we first need to define the measures we used to evaluate the performance of the generated models namely Specificity, Sensitivity, Correct Classification Rate (CCR) and Receiver Operating Characteristic Curve & Area Under the Curve (ROC-AUC). To calculate these metrics, we need first to define the following four rates:

TP The true positive rate is the number of peptides with observed antibacterial activity, which are correctly predicted peptides with antibacterial activity.

TN The true negative rate is the number of peptides with no observed antibacterial activity, which are correctly predicted as peptides with no antibacterial activity.

FP The false positive rate is the number of peptides with no observed antibacterial activity, which are incorrectly predicted as peptides with antibacterial activity.

FN The false negative rate is the number of peptides with observed antibacterial activity, which are incorrectly predicted as peptides with no antibacterial activity.

Specificity The proportion of the number of correctly predicted inactive peptides to the number of all inactive peptides and is formulated as follows:

$$Specificity = \frac{TN}{TN + FP} \quad (3.1)$$

Sensitivity The proportion of the number of correctly predicted active peptides to the number of all active peptides and is calculated as:

$$Sensitivity = \frac{TP}{TP + FN} \quad (3.2)$$

Correct Classification Rate (CCR) Also known as balanced accuracy, is defined as the number of correctly predicted data point to the number of all predicted data points. The following equation shows, how CCR is calculated

$$CCR = \frac{Specificity + Sensitivity}{2} \quad (3.3)$$

ROC-AUC The ROC curve is the plot of the True Positive Rate (TPR) (on the y-axis) versus the False Positive Rate (FPR) (on the x-axis) for every possible classification threshold. As a reminder, the True Positive Rate answers the question, "When the actual classification is positive (meaning antibiotics), how often does the classifier predict positive?" The False Positive Rate answers the question, "When the actual classification is negative (meaning not antibiotics), how often does the classifier incorrectly predict positive?" Both the TPR and the FPR range from 0 to 1.

Naturally, one can use the ROC curve to quantify the performance of a classifier, and assign it a score. That is the purpose of AUC, which stands for Area Under the Curve. AUC is the definite integral of the area that is under the ROC curve. The AUC score ranges between 0 (worst classifier) to 1 (best classifier).

3.3.2 NRPS Comprehensive Combinatorial Biosynthesis (NRPS-CCB)

As demonstrated in section 3.2.1.3, we ran four versions of the CCB algorithm. At each run we generated peptides of distinct lengths ranging between two and fourteen monomers. (1) The **No replacement and No genus** resulted in 395985 peptides, (2) The **No replacement and genus** resulted in 14545 peptides, (3) The **Replacement and No genus** resulted in 6215232 peptides, and (4) The **Replacement and genus** resulted in 611507 peptides.

The concatenation of the outcomes across the four runs resulted in a single output dataframe that entails 6006168 generated NRPs. Table 3.2 demonstrates a subset of the final output.

3.3.3 Sequence Based Model

3.3.3.1 Model evaluation

As mentioned in section 3.2.2.7, we built four binary classifiers using four distinct machine learning algorithms: (a) DNN, (b) SVM, (c) RF and (d) LG. We evaluated the models based on two criteria:

Peptides	length	Genus	Replacement
['aad', 'cys', 'gln', 'glu']	4	No Genus	False
['aad', 'cys', 'gln', 'gly']	4	No Genus	False
['gly', 'pro', 'thr', 'val']	4	Actinokineospora	False
['gly', 'ser', 'thr', 'val']	4	Actinokineospora	False
['aad', 'cys', 'cys', 'cys']	4	No Genus	True
['aad', 'cys', 'cys', 'gln']	4	No Genus	True
['asp', 'asp', 'phe', 'val']	4	Actinomadura	True
['asp', 'asp', 'thr', 'thr']	4	Actinomadura	True

Table 3.2: **A subset of the final outcome of the CCB algorithm:** A subset demonstrating two peptide examples from each of the four runs.

1. **AUC levels:** We have conducted a five fold cross validation on every algorithm except the DNN model, hence we ended up with 5 distinct AUC scores (one for each fold) and additionally computed the average AUC score. We found out that the RF based classifiers not only have reached an average AUC level as high as 98% but also outperformed the other classifiers by a margin of almost $\sim 17\%$ as shown in Figure 3.11. Moreover, in order to ensure the statistical significance of our models, we applied the y-randomized protocol as explained in section 3.2.4.7. This resulted in a steep drop in all AUC levels confirming a high level of statistical significance to all our developed models. Figure 3.12 shows an example of the disparity in AUC levels between a regular RF model and y-randomized one. To see the same effect on the rest of our developed models, please refer to the Supplementary section (Figure S1 and Figure S2).
2. **Correct Classification Rate:** Here, we used our classifiers to predict the labels of a hidden test set. Across all four classifiers we were able to compute four distinct CCR scores. Figure 3.13 shows that the DNN classifier outperforms all others classifiers by reaching a CCR score as high as 97%, followed by RF, LG and SVM with a CCR score of 94%, 72% and 67%, respectively. Figure 3.14 illustrates the steady increase in the model accuracy on both the training and validation sets versus the constant decline in classification error across both the training and validation sets throughout the training of the DNN model.

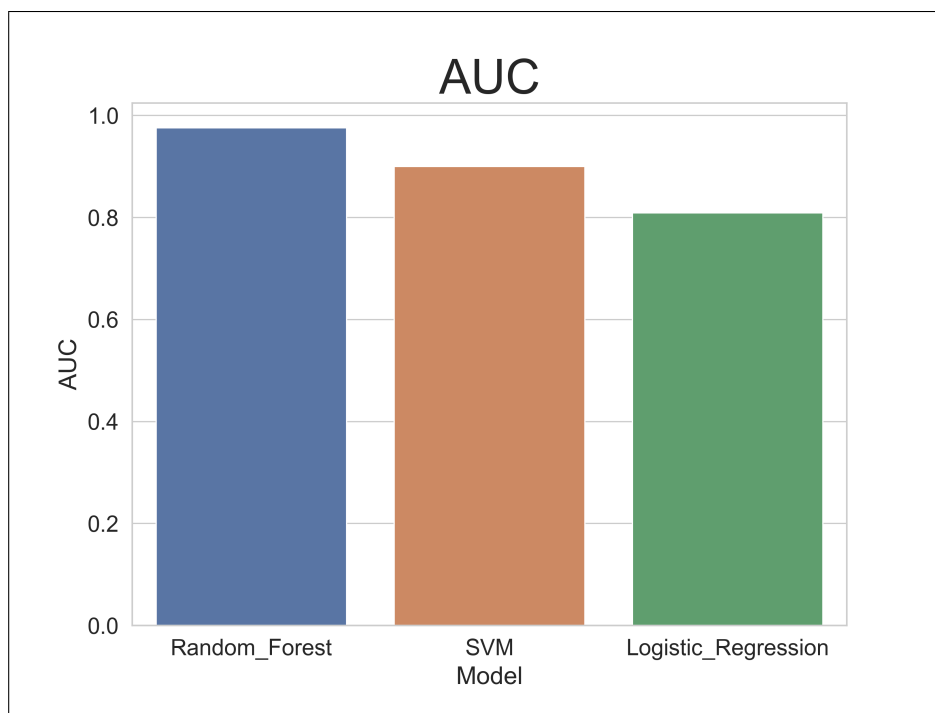


Figure 3.11: **AUC performance:** This is a model performance summary based on the AUC levels achieved by three different classifiers, namely, Random Forest (RF), Support Vector Machine (SVM) and Logistic Regression (LR)

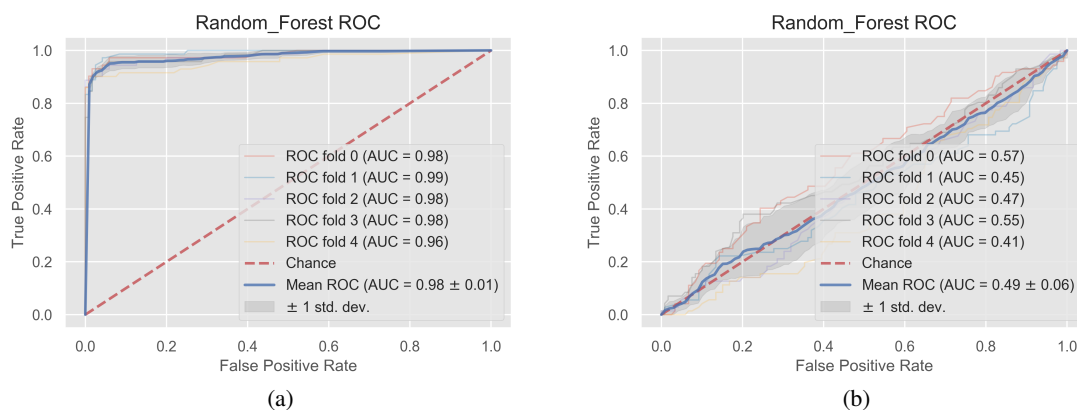


Figure 3.12: **Regular vs. randomized:** The AUC levels of the five fold cross validation in a regular vs. a randomized random forest based models.

3.3.3.2 Virtual screening

After developing and validating our models, we applied them to screen our prediction set. Figure 3.15 illustrates the results of the virtual screening process using all our four models. It shows that the deep learning models resulted in the highest number of hits where 2217622 out of 6006168 were

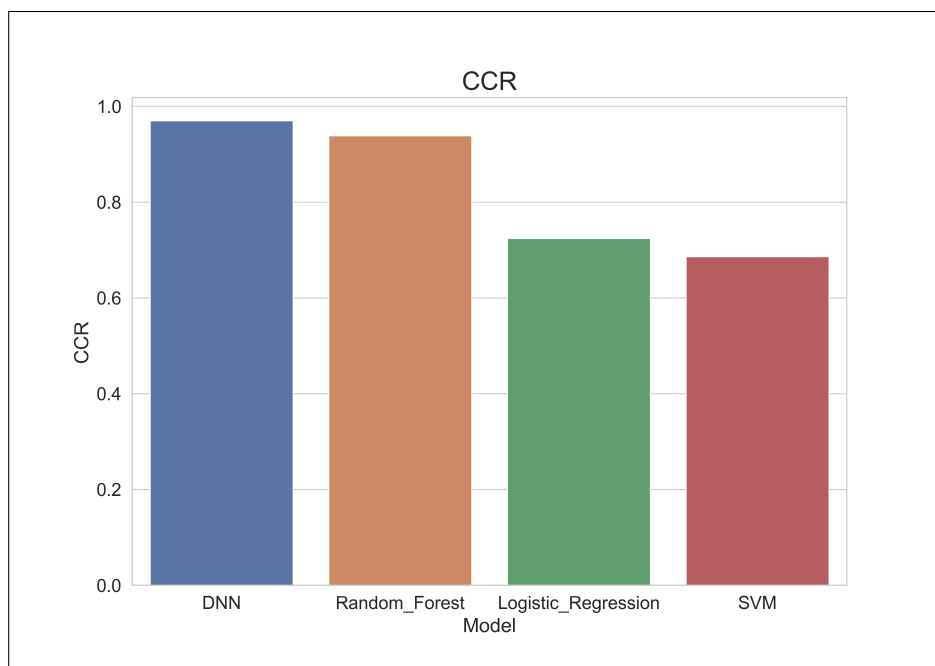


Figure 3.13: **CCR performance:** This is a model performance summary based on the CCR levels achieved by four different classifiers, namely, Deep Neural Network (DNN), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR)

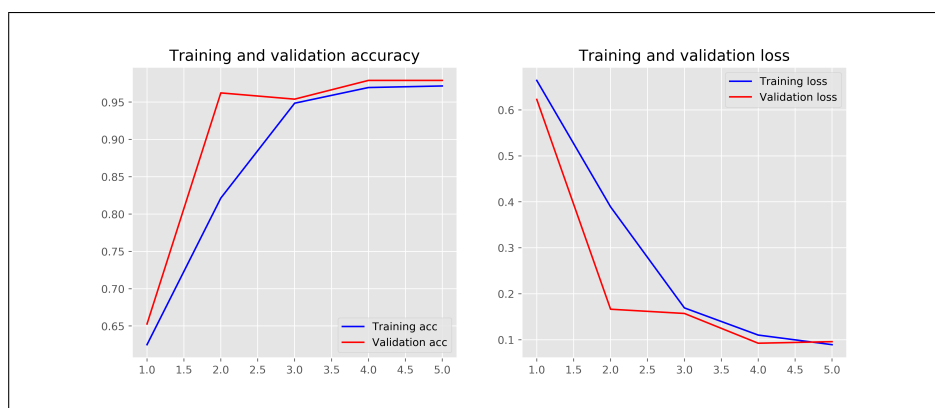


Figure 3.14: **DNN Training and Validation:** (Left) The training and validation accuracy of the DNN model throughout all five epochs. (Right) The training and validation loss of the DNN model throughout all five epochs.

predicted as actives while the remaining 3788546 compounds as inactive. On the other hand, for the rest of the classifiers (RF, LG and SVM) the inactive class strongly dominated the prediction outcome.

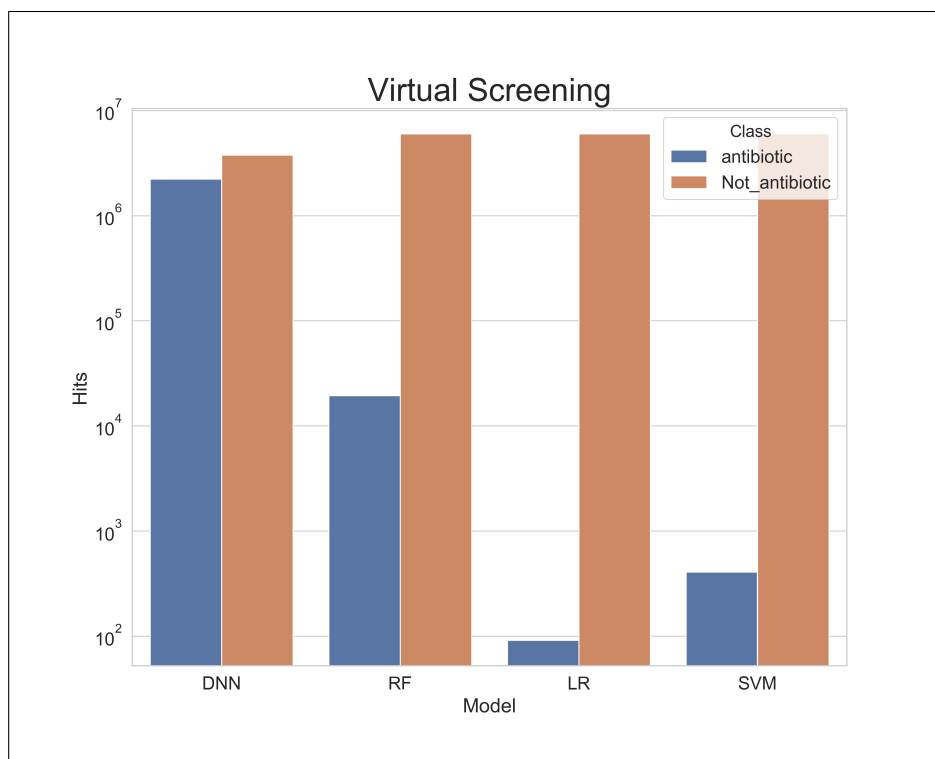


Figure 3.15: **Virtual screening of the Prediction set:** 6006168 virtually generated peptides have been screened against our four classifiers namely, DNN, RF, LG and SVM.

3.3.3.3 Reducing the number of hit list

Despite the fact that our models were capable to classify the vast majority of the peptides as inactive, the volume of hits (potential antibiotics) that were predicted still lies in the range of millions as in DNN: 2217622 or thousands as in RF: 11518, which would be a very large number to pass on to the next step in the pipeline. Hence, in order to further reduce the number of potential hits we conducted these two extra filtration steps.

1. We kept only the peptides that were predicted as active by our top two classifiers, namely DNN and RF. This helped reducing the number of hits to just 16675 peptides.
2. We eliminated all peptides that contain the monomer “nrp”, which denotes to an inconclusive monomer. This step resulted in a further reduction of our active hit list by almost a factor of two, leading to a final hit list in size of 9000 peptides.

3.3.4 Pep2Struc: Peptide to Structure

Here we fed our tool the 9000 predicted peptides from phase two to convert them into their 2D structures. We ran pep2struc twice each with a different mode (linear & cyclic). The linear mode resulted in 7274 linear peptides, while the cyclic version resulted into 30869 peptides. The fact that the cyclic mode has led to almost 3.5 times more peptides than the original input, is mainly due to the improper protection to some of the chemical functional groups while conducting the reaction. This, of course, would lead sometimes to multiple products including the desired one. It goes without saying that this is one of the main limitations of the current version of our tool, which will be certainly addressed in future updates. However, at the time, this was not of a big concern to us as eventually all converted peptides will be then pushed to phase four to be predicted by our QSAR models and only those who retain positive activity (potential antibiotics) will make it to the last phase of our pipeline.

3.3.5 Structure Based Model

3.3.5.1 Model evaluation

As mentioned in section 3.2.4.7. We applied three distinct binary machine learning algorithms: (a) RF, (b) LR, and (c) SVM . For each one of them we computed five descriptors (Section 3.2.4.6), leading to a total of 12 different classifiers. Each one was evaluated based on two criteria:

1. **AUC levels:** For each classifier, we applied a five fold cross validation, which led to 6 distinct AUC scores, one for each fold and their average AUC score as well. We found out that the RF based classifiers not only has reached the highest score with an average AUC level of 96% and outperformed the other classifiers but it did that across all descriptors (Fig 3.16). As elaborated earlier, in order to ensure the statistical significance of our models, we applied the y-randomized protocol as explained in Section 3.2.4.7. This resulted in a steep drop in all AUC levels as demonstrated in the Supplementary section (Fig S3, Fig S4, Fig S5, Fig S6, Fig S7 and Fig S7) confirming a high level of statistical significance to all our developed models.
2. **Correct Classification Rate:** Here, we used our classifier to predict the labels of a hidden test set. Across all 12 classifiers we were able to compute twelve distinct CCR scores. Figure 3.17

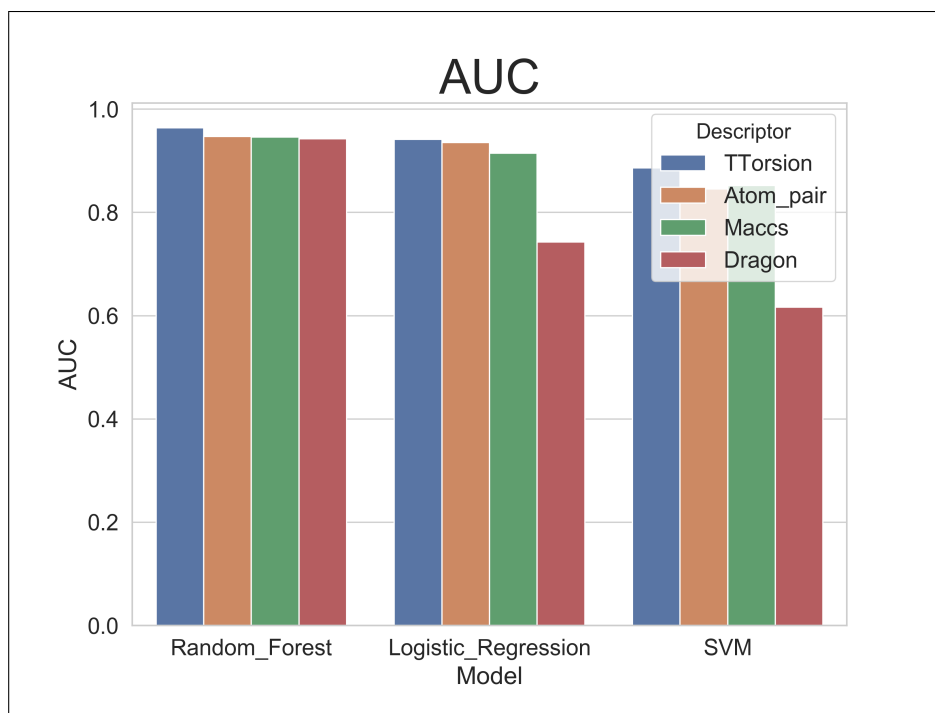


Figure 3.16: **AUC performance:** This is a model performance summary based on the AUC levels achieved by twelve classifiers, namely, Random Forest (RF), Support Vector Machine (SVM) and Logistic Regression (LR). For each machine learning algorithm four different descriptors were computed: (1) Dragon, (2) Atom pair fingerprints, (3) Topological torsion fingerprints, (4) MACCS keys fingerprints.

shows that the LR based models using atom pair fingerprints achieved the highest CCR score of 91% outperforming both the RF (90%) and the SVM (70%) based models using the same descriptor.

3.3.5.2 Virtual screening

After developing and validating our models, we applied them to screen our prediction set. Figure 3.19 illustrates the results of the virtual screening process using nine of our twelve models. It shows that the LR based classifiers resulted in the highest number of hits where 19222 out of 30869 were predicted as active while the remaining 11647 compounds as inactive. The remaining classifiers RF, LR and SVM were also able to predict a large numbers of hits precisely 18108 and 18185, respectively.

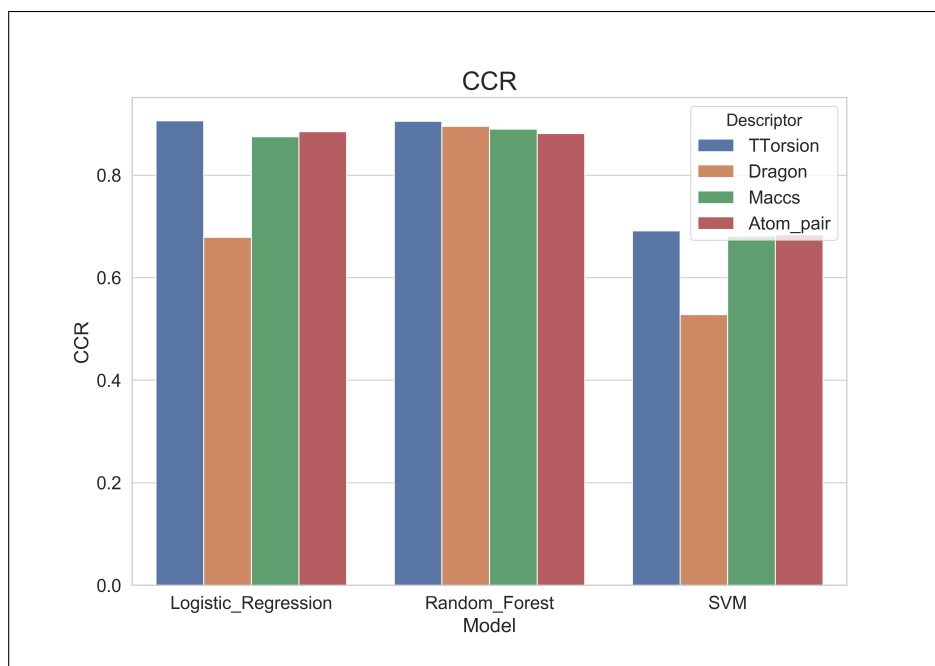


Figure 3.17: **CCR performance:** This is a model performance summary based on the CCR levels achieved by four different classifiers, namely, Deep Neural Network (DNN), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR). For each machine learning algorithm four different descriptors were computed: (1) Dragon, (2) Atom pair fingerprints, (3) Topological torsion fingerprints, (4) MACCS keys fingerprints.

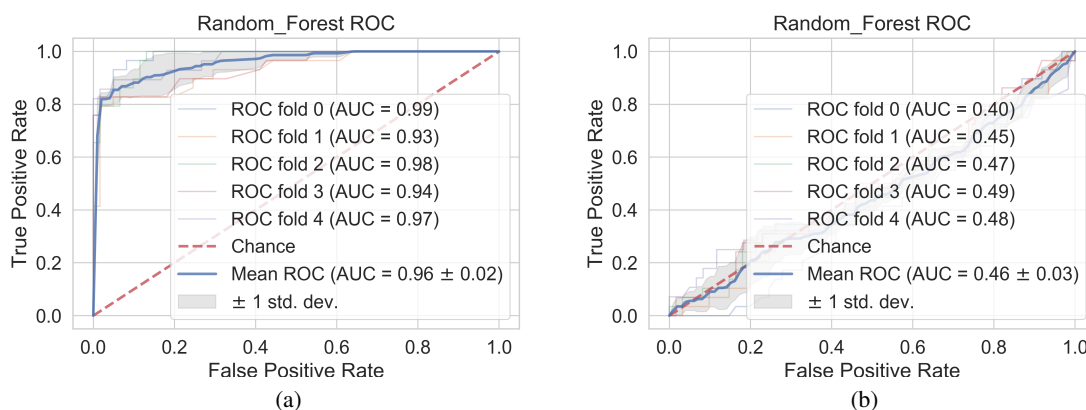


Figure 3.18: **Regular vs. randomized:** The AUC levels of the five fold cross validation in a regular vs. a randomized random forest based models.

3.3.5.3 Reducing the number of hit list

Despite the fact that our models were capable to classify the vast majority of the peptides as inactive, the volume of hits (potential antibiotics) that were predicted still lies in the range of thousand, which

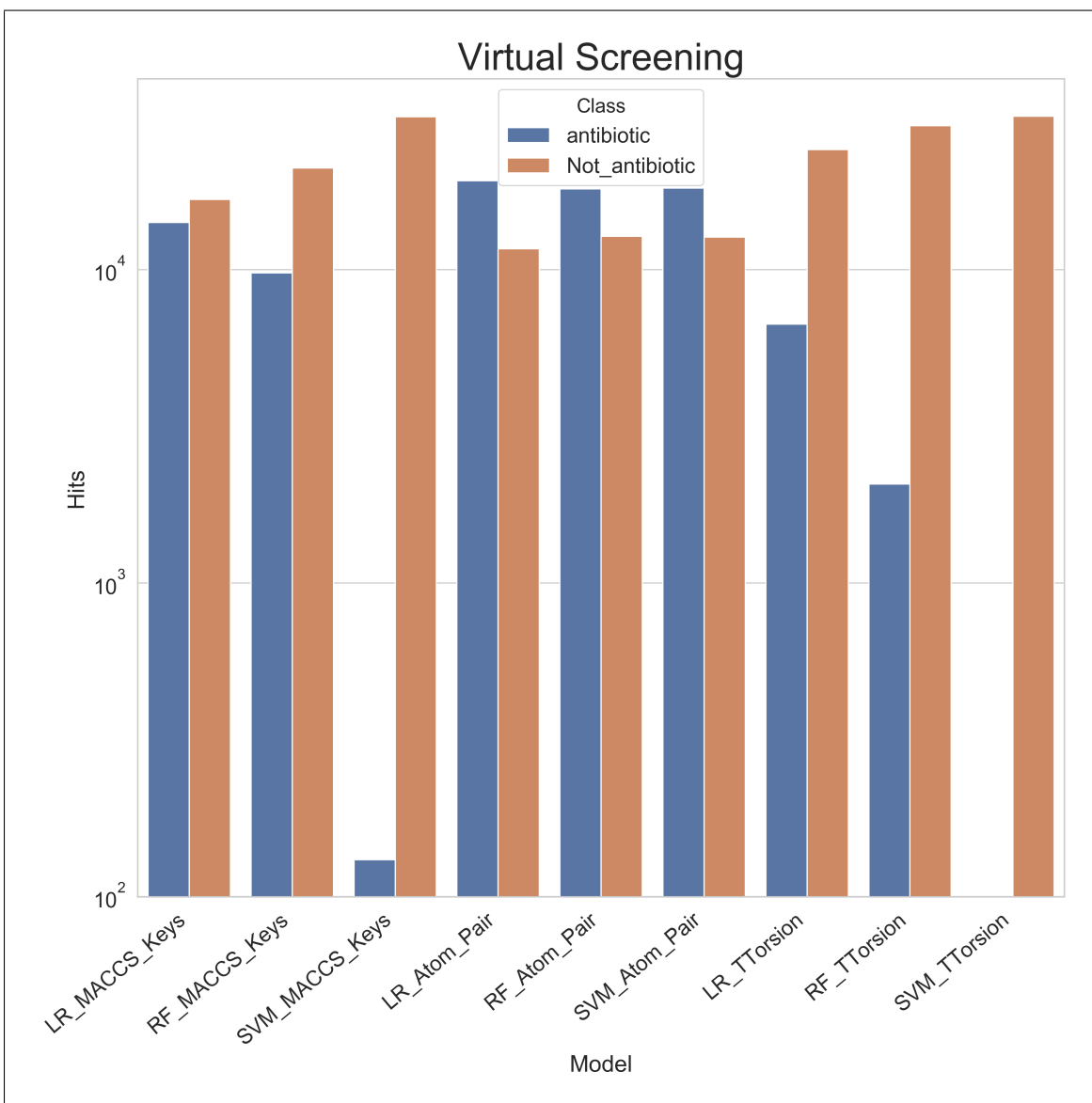


Figure 3.19: **Virtual screening of the Prediction set:** 30869 peptides have been screened against nine classifiers

would be a very large number to pass on to the next phase of our pipeline. Hence, in order to further reduce the number of potential hits we conducted these two extra filtration steps.

1. We kept only the peptides that were predicted as active by 7 of our 9 classifiers. This help reducing the number of hits drastically to just 490 peptides.
2. A structural similarity search was conducted using RDkit fingerprints between the 490 predicted active compounds and 3 known families of antibiotics namely: Tyrocidin, Polymyxin and Bacitracin. For each family we picked the peptide with the highest similarity as depicted in Figures 3.20 to 3.25.

Three peptides show structure similarity of more that 85% towards a know family of antibiotics namely Gramicidin (Editor Drug-bank, 2010). Finally, we choose only those 3 peptides to be passed onto the final phase of our pipeline.

3.3.6 NRPS Designer

The three peptides that have reached this stage of the pipeline were then desgined using our NRPS-designer tool and three NRPS-cluster constructs were generated in the form of a FASTA format file.

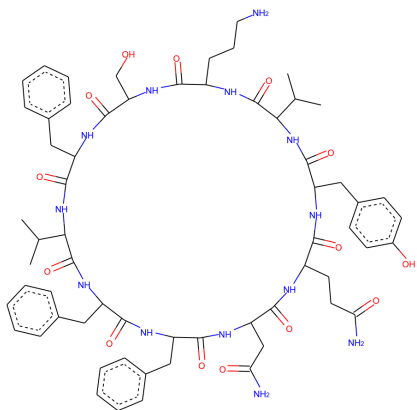


Figure 3.20: **Candidate 1 (62%)**

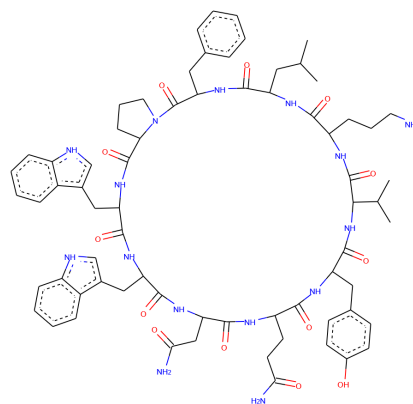


Figure 3.21: **Tyrocidin A**

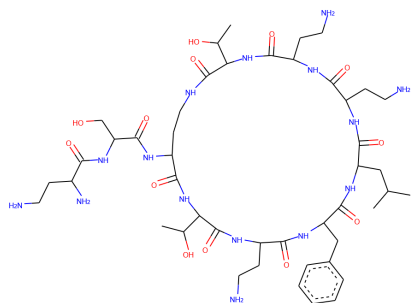


Figure 3.22: **Candidate 2 (98%)**

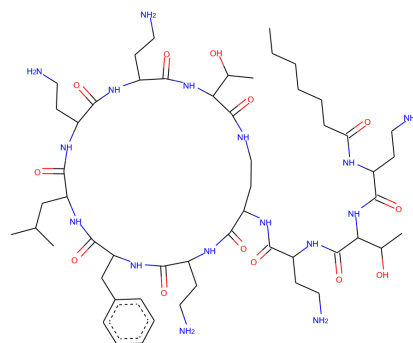


Figure 3.23: **Ile-Polymyxin B1**

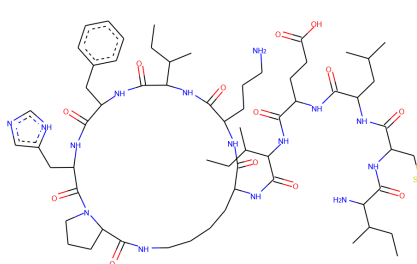


Figure 3.24: **Candidate 3 (72%)**

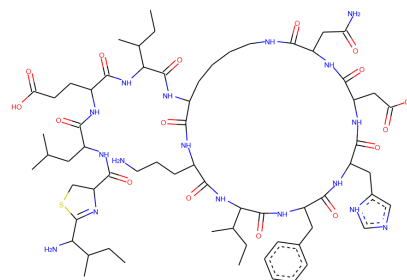


Figure 3.25: **Bacitracin A1**

3.4 Discussion

The NRP Discovery Pipeline is a collection of modules that will push forward early discovery of novel NRPs. The pipeline comprises of 5 tools, each one of them could be used either individually (standalone tool) or integrated into a pipeline.

CCB: Running the CCB algorithm resulted in 6006168 new peptides. This number is way below the theoretical possible number of generated peptides. This is due to the hard condition we applied, namely, we eliminated all peptides for which the pairs of monomers lacked a valid IML. Therefore, the number of possibly generated peptides rely heavily on the size of our IML database. We are certain that with more new species of bacteria and fungi being discovered and their genomes being sequenced, our IMLs database will keep growing and so would the outcome of the CCB tool.

Sequenced based classifier: All our trained sequence based classifiers showed a relatively high levels of AUCs when conducting a five fold cross validation (Fig 3.12, Supplementary Fig S1 and Fig S2). However, when we validated our model using a validation test set only the deep neural network and random forest models showed a relatively high CCR score, precisely 97% and 94%, respectively, while LR and SVM models didn't do better than 72%. The rationale behind the out-performance of DNN and RF over LR and SVM is due to the non-linear nature of the first two algorithms and their ability to learn complex nonlinear relationships between highly dimensional data. The disparity between the aforementioned models, could also be seen when we applied them to conduct virtual screening step. Figure 3.15 shows that only DNN and RF were able to predict both classes of activity, while the LR and SVM returned predominantly inactive predictions.

Pep2struc: This tool helps converting simple non-ribosomal peptide sequences into their complex 2D structures. The tool can be used to build the structure of linear peptides as well as cyclic peptides. For the latter, we had to make a choice between applying only the head-tail cyclization and thus limiting the ability of our tool in creating other forms of cyclic peptides (partial cyclic and double cyclic) or allowing all sorts of possible cyclizations. Therefore, in order to ensure that we can capture all possible cyclic forms of a linear peptide, we decided to relax the protection step in a way that enables all possible cyclization reactions. The drawback of such a measure, is that the number of converted peptides will be doubled and sometimes even tripled as compared to the input data. When

we applied our tool in cyclic mode on the 9000 peptides, we therefore obtained 30869 peptides which is almost ~ 3.5 times more peptides than the original input data. This was anticipated and was not of a big concern to us as eventually all converted peptides will be pushed to phase four to be predicted by our QSAR models. Only those who retain positive activity (potential antibiotics) will reach the last phase of our pipeline.

NRPS designer: Here, a construct for the desired peptide will be built and exported as a FASTA format file. In Section 3.2.5.2, we mentioned that there are two modes for the algorithm, a template based mode and a from scratch mode. The former approach is more desired as it is more accurate with no assumptions being made regarding the interactions between the distinct NRPS subunits and it requires less edits to the template NRPS, leading to a more homogeneous construct. The latter, is harder to achieve as few questions need to be answered prior building the construct: (1) The desired peptides will be distributed across how many NRPS subunits? (2) If multiple subunits is considered, then would all of share the same number of modules or it would differ from one NRPS subunit to another? These questions are not intuitive to answer. Moreover, the poor understanding of how NRPS subunit interact with each other makes it even harder to predict their effect on synthesizing the desired peptide.

3.5 Conclusion

In this study we introduced NRP Discovery Pipeline, which is a cluster of bioinformatics and cheminformatics tools that will help facilitating early discovery of novel NRPs. Our pipeline comprises of five modules that will be involved in generating virtual libraries of NRPs, converting peptide sequences into their 2D structures, developing highly predictive models to predict anti-bacterial activity of generated peptides and finally editing and designing novel NRP BGCs at the molecular level. Running our pipeline resulted in 28 novel NRPs, from which we designed 3 peptides to be validated experimentally. We are certain that our pipeline would help pushing forward the early discovery of novel NRPs.

3.6 Supplementary

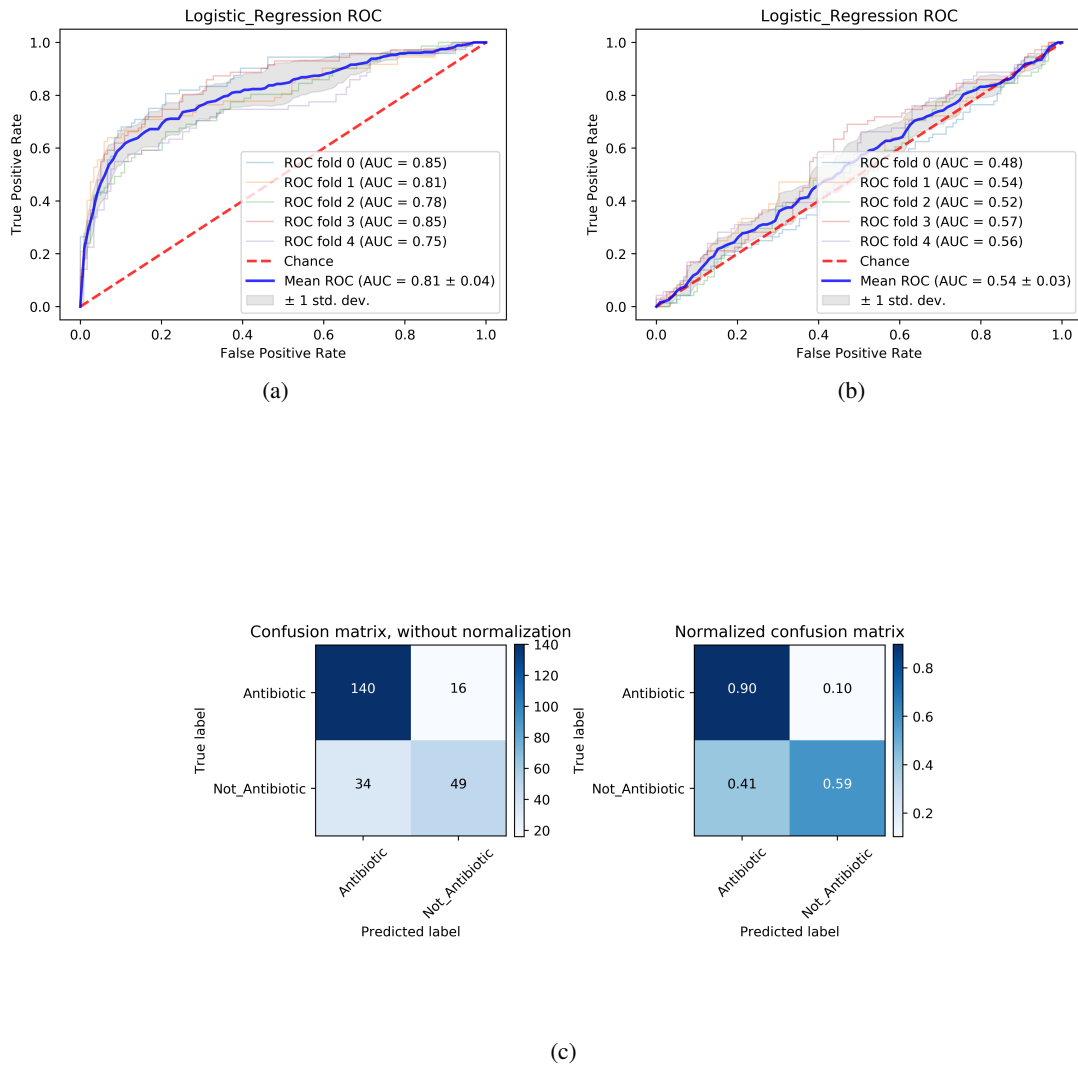
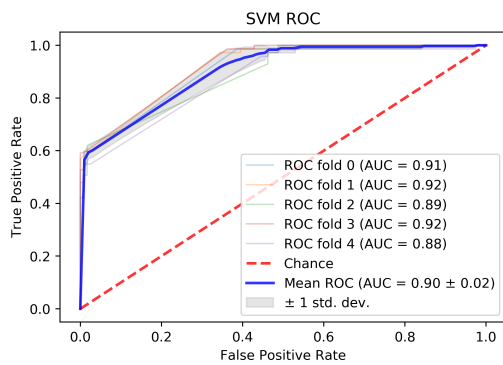
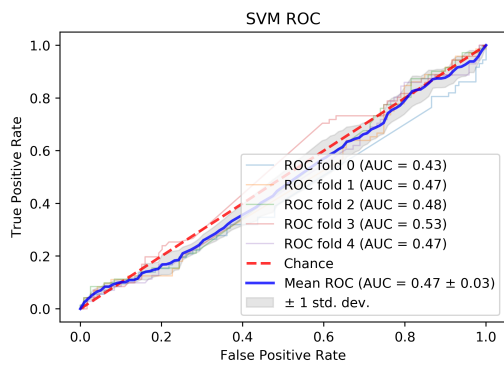


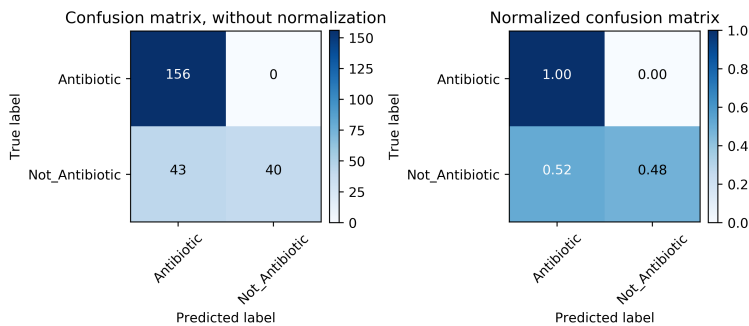
Figure S1: **Sequence based model - Logistic regression:** (a) The AUC levels of the five fold cross validation in a regular model. (b) The AUC levels of the five fold cross validation in a randomized model. (c) Correct classification rate of test set



(a)



(b)



(c)

Figure S2: **Sequence based model - Support vector machine:** (a) The AUC levels of the five fold cross validation in a regular model. (b) The AUC levels of the five fold cross validation in a randomized model. (c) Correct classification rate of test set

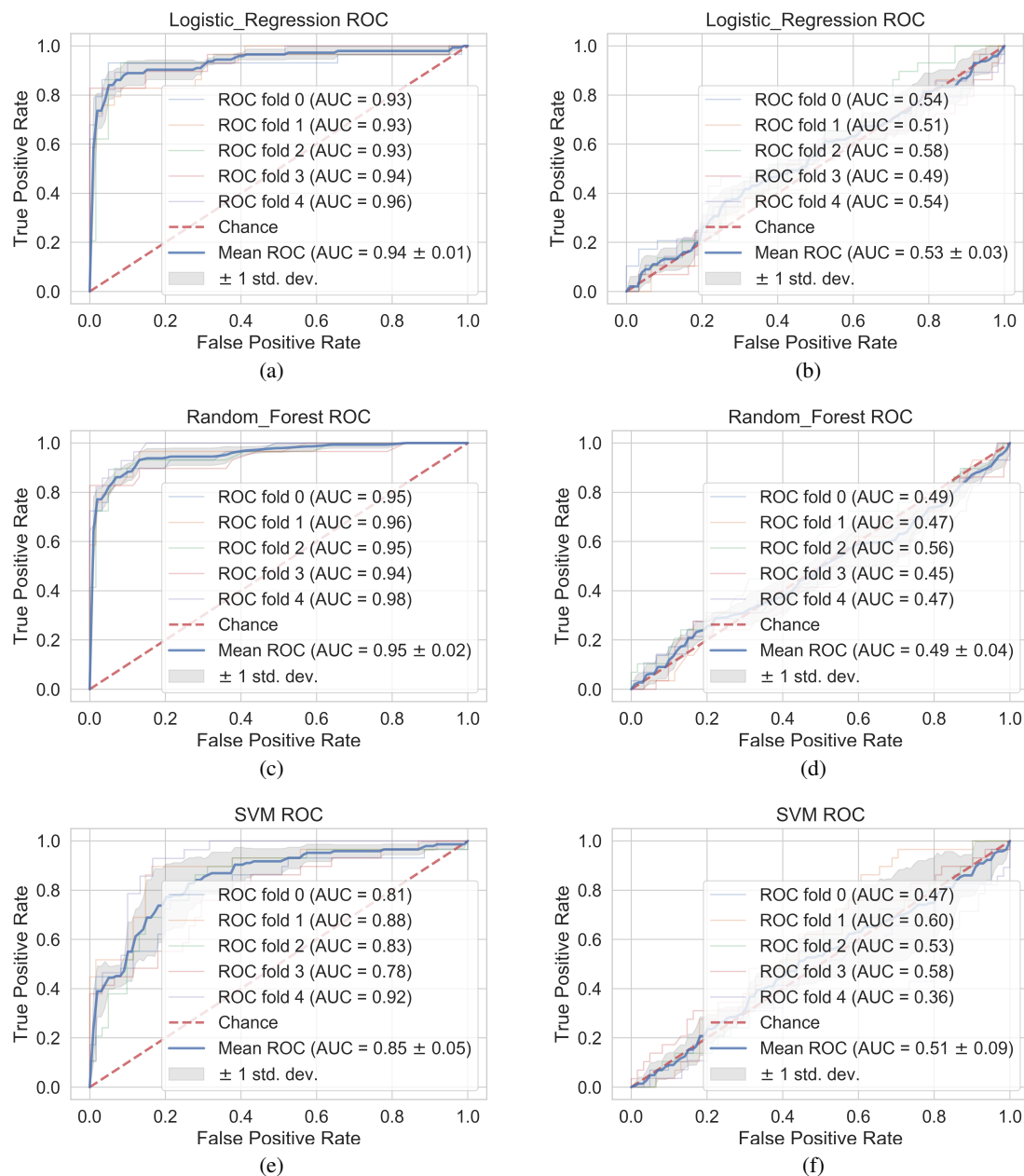


Figure S3: **Structure based models with Atom pair fingerprint:** The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.

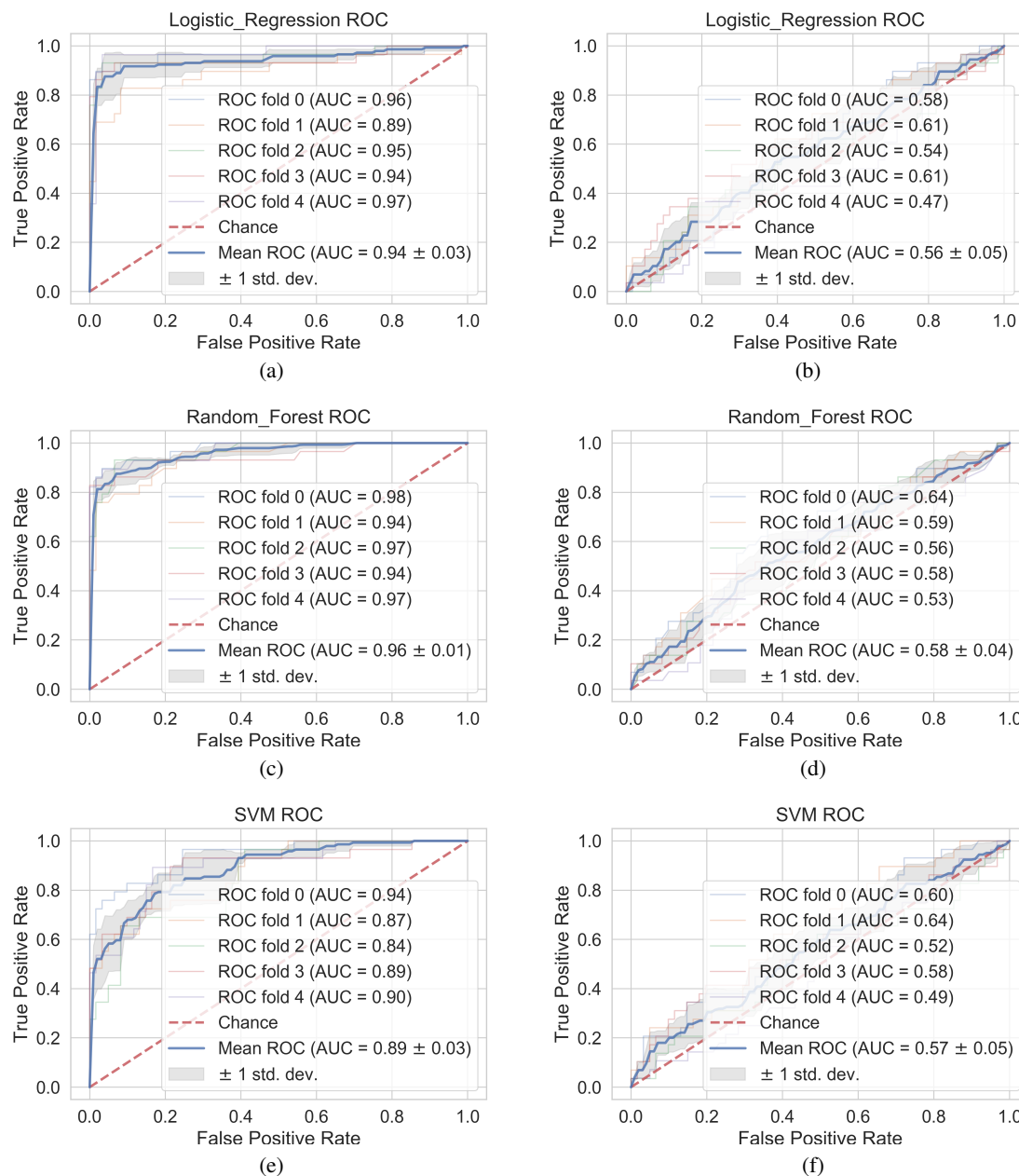


Figure S4: **Structure based models with Topological torsion fingerprint:** The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.

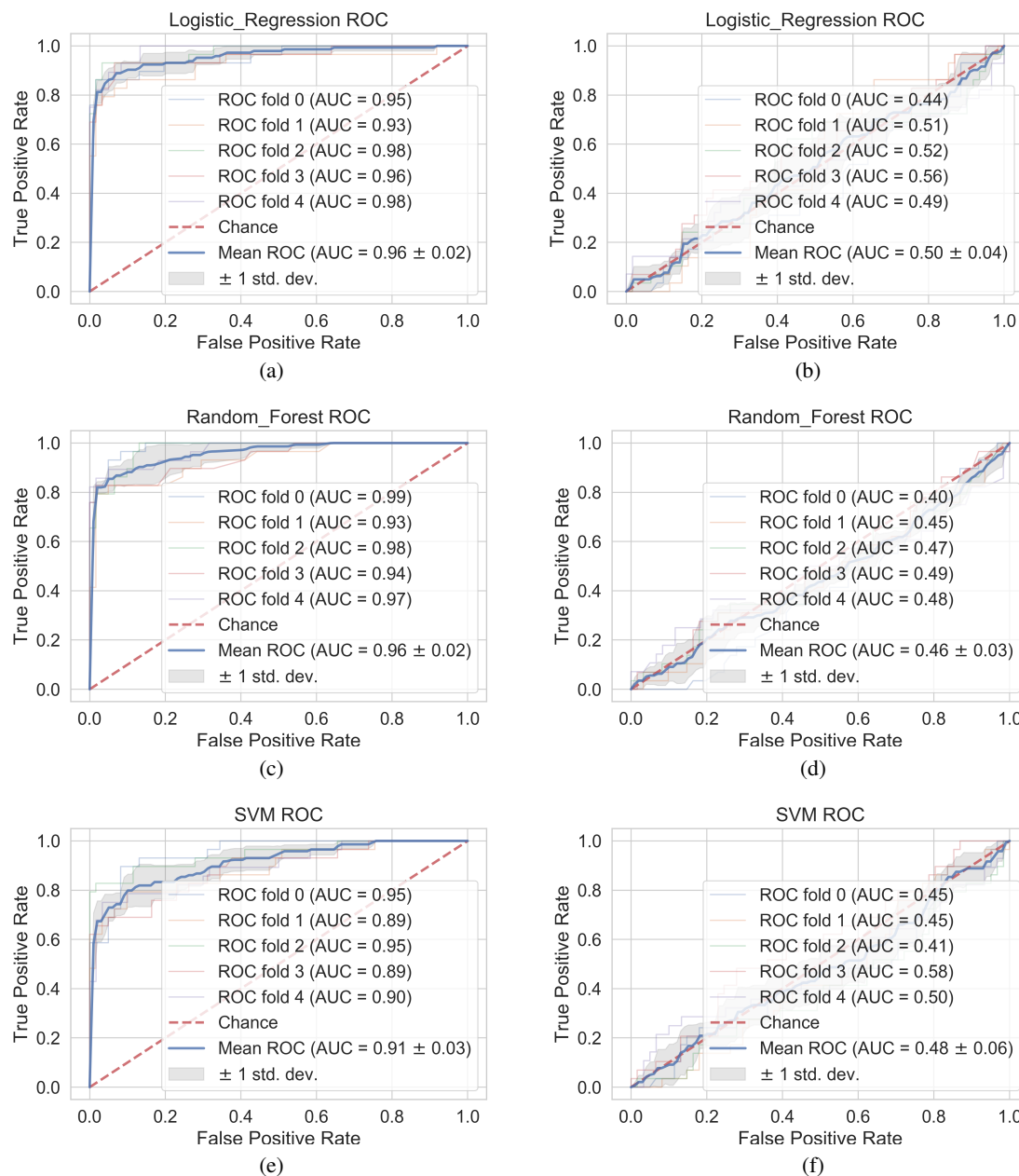


Figure S5: **Structure based models with Morgan circular fingerprint:** The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.

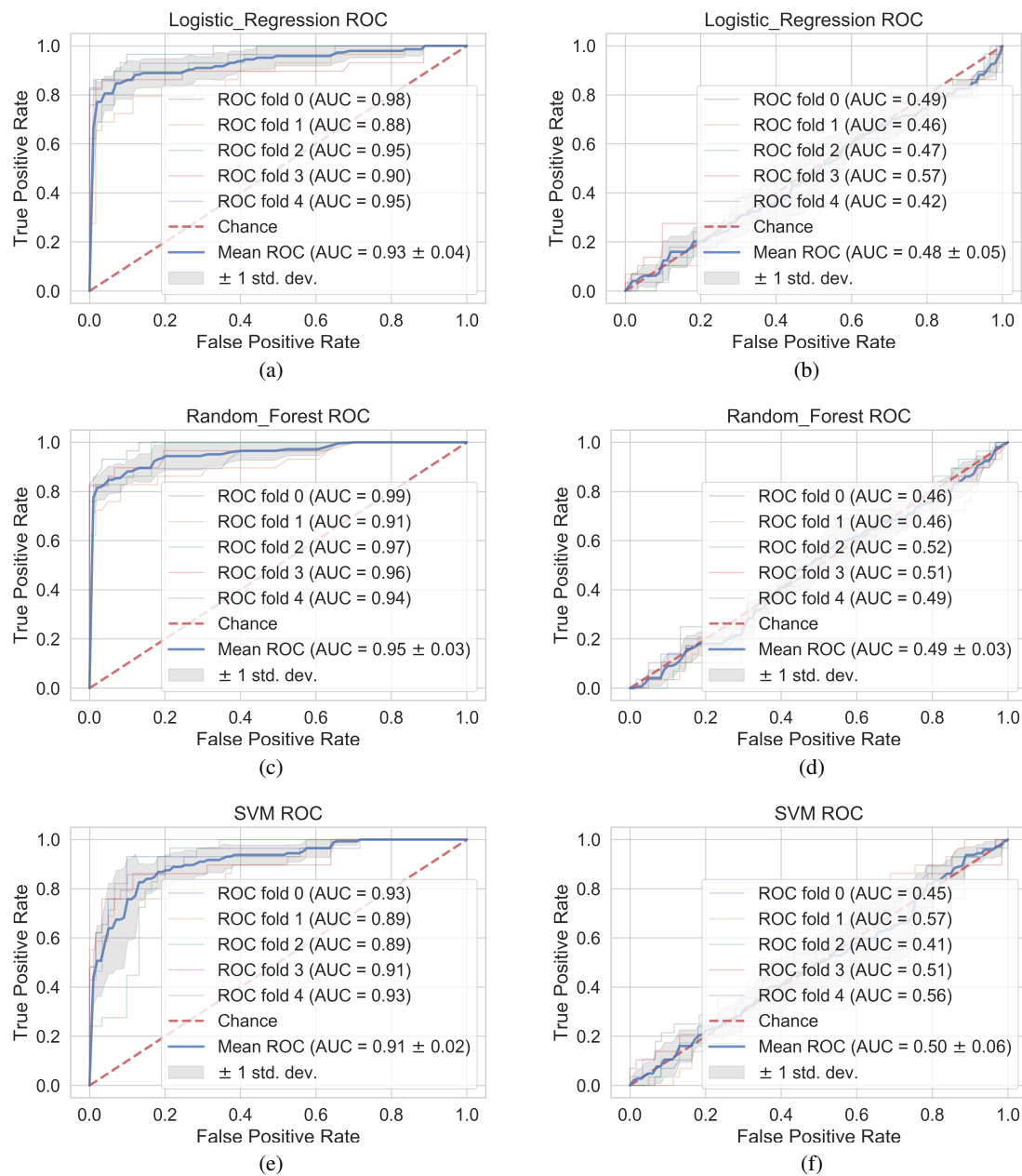


Figure S6: **Structure based models with RDKit fingerprint:** The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.

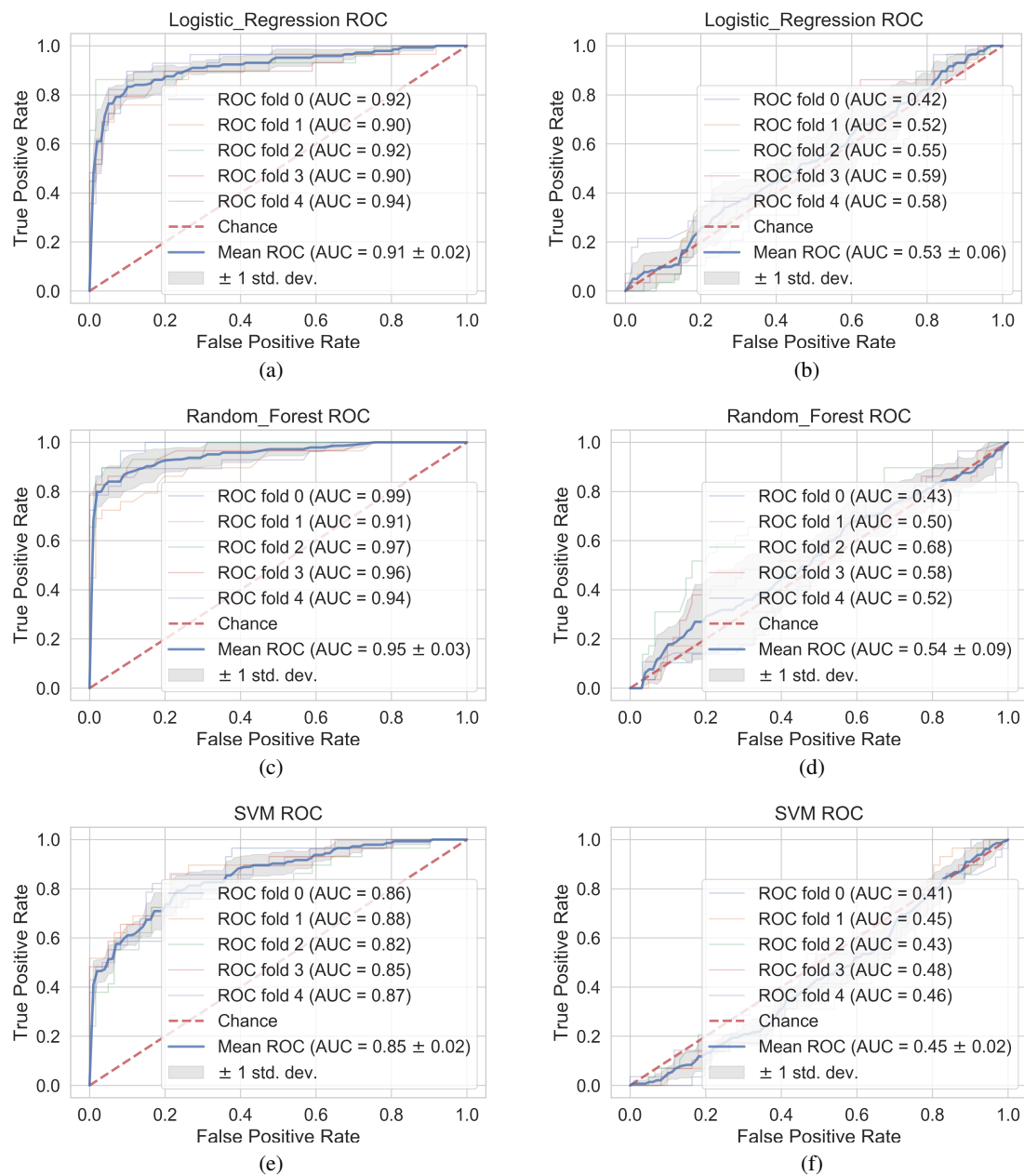


Figure S7: **Structure based models with RDKit MACCSkeys:** The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.

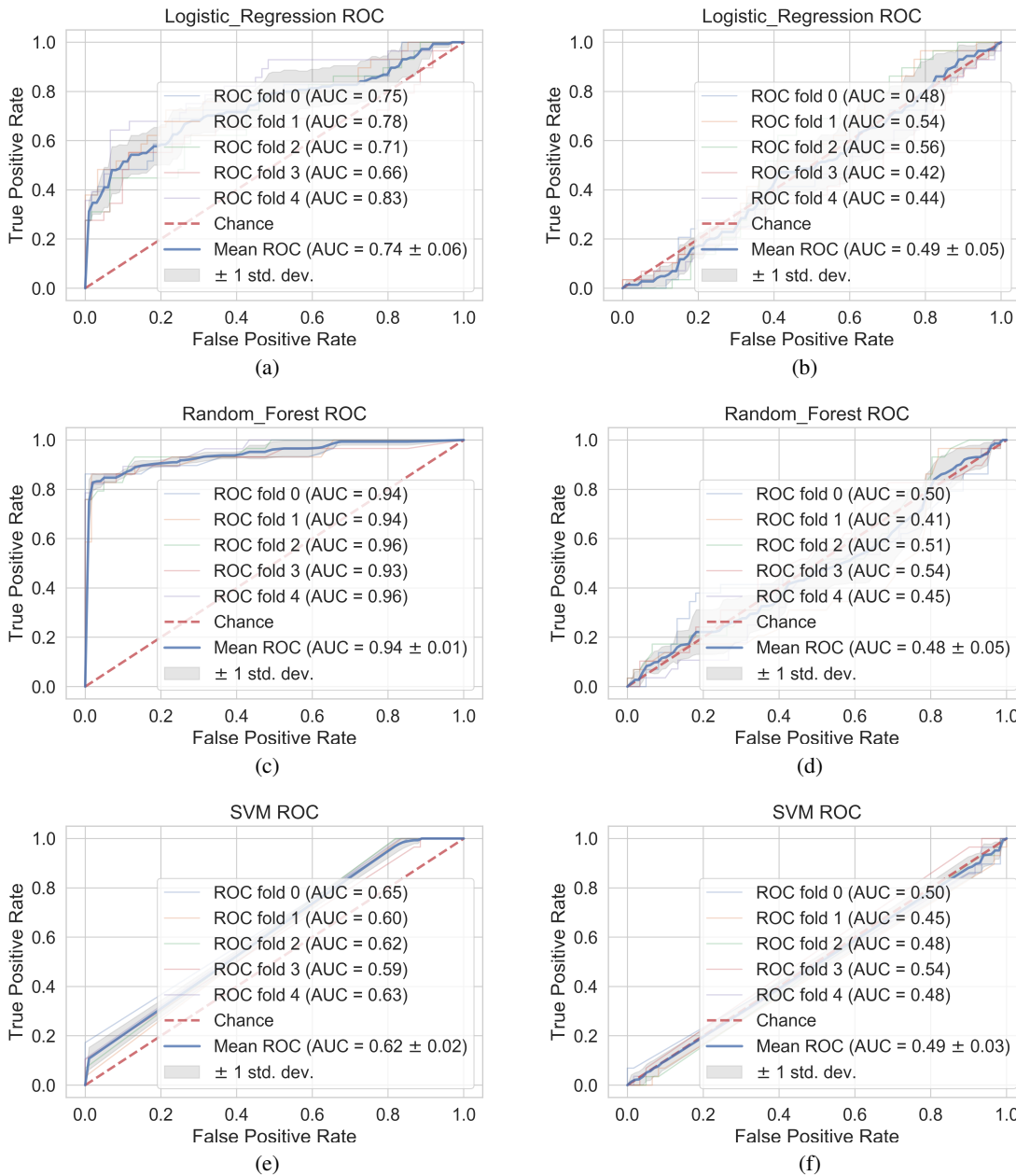


Figure S8: Structure based models with Dragon descriptors: The AUC levels of the five fold cross validation in a regular model vs a randomized model across three machine learning algorithms: logistic regression, random forest and support vector machine.

CHAPTER 4

Conclusion and Future Directions

Using our IML-Parser, we extracted more than 39k NRPS IMLs and analyzed their association with their adjacent A domain substrates. This led to the discovery that IMLs are very specific to the A domain modules that they connect, with more than 92% of the identified IMLs being associated with a specific pair of modules. We also determined that the same IML could be involved in the biosynthesis of different NRP products across various bacterial genera. Overall, however, IMLs that link a particular module pair show a low degree of conservation across bacterial genera. We also determined that IMLs exhibit more secondary structures (α -helices) than IDLs, however, they share similar hydrophobic profile. Furthermore, as a proof-of-concept, we retrospectively analyzed the findings of Nguyen et al. (2006) and Bozhüyük et al. (2017) demonstrating that IMLs incompatibility could dramatically impact biosynthetic yields of daptomycin lipopeptides and ambactin analogues. Overall, our data indicate a strong relationship between NRPS IMLs and their adjacent A domains. This finding suggests that, going forward, combinatorial biosynthesis strategies to generate novel NRPs should consider IMLs in addition to other established parameters (Nguyen et al., 2006; Coëffet-Le Gal et al., 2006; Baltz et al., 2006; Crüsemann et al., 2013; Calcott et al., 2014; Meyer et al., 2016; Bozhüyük et al., 2017).

Furthermore, we have introduced NRP Discovery Pipeline, which is a set of bioinformatics and cheminformatics tools that will help facilitating early discovery of novel NRPs. Our pipeline comprises of five modules: (1) NRP comprehensive combinatorial biosynthesis (CCB): A tool that helps generating virtual libraries of NRPs. (2) NRP sequence-based predictor: A classifier based only on peptide sequences to help triaging all peptides with no anti-bacterial activity. (3) Pep2struc: A tool that helps converting peptide-sequences to their 2D structures form both linear and constrained peptides. (4) NRP structure-based predictor: A second classifier based on peptide structures to

filter out all inactive predicted peptides. (5) NRPS Designer: A tool that help re-programming of the bacterial genome by editing its NRP BGC to synthesize the peptide of interest. Running our pipeline resulted in 28 novel putative NRPs, from which we designed 3 peptides to be validated experimentally.

All 39,804 IMLs extracted in this study (Supplementary Table 2) as well as our parser are publicly available at <https://nrps-linker.unc.edu/>. Moreover, all source code for the NRP Discovery Pipeline is hosted in GitHub repository under <https://github.com/SWFarag/CCB>, <https://github.com/SWFarag/NRP-structure-classifier>, <https://github.com/SWFarag/pep2struc>, <https://github.com/SWFarag/NRP-structure-classifier>, <https://github.com/SWFarag/NRPS-designer>.

We anticipate that both the NRPS-Linker tool and the NRP Discovery Pipeline will not only facilitate mining the data we have analyzed here, but will also enable interested researchers to expand their studies as new genomes are obtained. Our study lays the foundation for future experimental validations of our hypothesis that IMLs play a crucial role in governing the biosynthesis of NRPs. We expect that additional approaches and tools could be developed that rely on this finding and facilitate the design of novel NRPS BGCs using the most appropriate IMLs for combinatorial biosynthesis of novel NRPs.

Future directions of this work should include exploring more data and expanding our IMLs database by scanning more novel yet unidentified NRPs from not only bacteria but also fungi and marine microbiomes. Additionally, we want to expand our database by including all known NRP domains and their linkers across all species. With such a large scale database, we would be able to conduct a comprehensive sequence analysis that indeed would help us gain more insights about the nature and evolution of these domains and will help us unravel the true relation between them and their adjacent linkers. Finally, a major step forward would be the development of the NRP Discovery Hub 1.0, which is a web-accessible platform that will eventually integrate all our tools and databases to facilitates early discovery of novel NRPs.

Unfortunately, Many important bioinformatics tool are yet not suitable to handle NRPs. This is mainly due the fact that most of the widely used bioinformatics tools such as Pairwise sequence alignment (PSA), Multiple sequence alignment (MSA), Basic Local Alignment Search Tool (BLAST), Sequence Logo and many others are built specifically for peptides that are made only of the 20 natural amino acids. Thus, in order to leverage their importance in the field of combinatorial biosynthesis

of novel NRPs, there is a need to tweak and re-adjust most of these algorithms to enable them to work with NRPs. The same problem could be extended to cheminformatics, where most of its descriptor generator software are optimized to deal with either small molecules $< 500 \text{ mw}(Da)$ or large molecules $> 10000 \text{ mw}(Da)$ but not medium size molecules as in case of macro-cycles with a molecular weight ranging between > 500 and $< 10000 (Da)$. Thus, there is a need to engineer a set of new descriptors that are tailored specifically to describe constrained peptides (macro-cycles). Doing such not only will help building precise QSAR models with more predictive power but it will indeed help the scientific community to push the process of early discovery of novel NRPs forward.

BIBLIOGRAPHY

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.
- Awale, M. and Reymond, J.-L. (2014). Atom Pair 2D-Fingerprints Perceive 3D-Molecular Shape and Pharmacophores for Very Fast Virtual Screening of ZINC and GDB-17. *Journal of Chemical Information and Modeling*, 54(7):1892–1907.
- Bae, K., Mallick, B. K., and Elsik, C. G. (2005). Prediction of protein interdomain linker regions by a hidden Markov model. *Bioinformatics*, 21(10):2264–2270.
- Baltz, R. H. (2006). Molecular engineering approaches to peptide, polyketide and other antibiotics. *Nature biotechnology*, 24(12):1533–1540.
- Baltz, R. H., Brian, P., Miao, V., and Wrigley, S. K. (2006). Combinatorial biosynthesis of lipopeptide antibiotics in *Streptomyces roseosporus*. *Journal of Industrial Microbiology and Biotechnology*, 33(2):66–74.
- Bartlett, J. G., Gilbert, D. N., and Spellberg, B. (2013). Seven Ways to Preserve the Miracle of Antibiotics. *Clinical Infectious Diseases*, 56(10):1445–1450.
- Beer, R., Herbst, K., Ignatiadis, N., Kats, I., Adlung, L., Meyer, H., Niopek, D., Christiansen, T., Georgi, F., Kurzawa, N., Meichsner, J., Rabe, S., Riedel, A., Sachs, J., Schessner, J., Schmidt, F., Walch, P., Niopek, K., Heinemann, T., Eils, R., and Di Ventura, B. (2014). Creating functional engineered variants of the single-module non-ribosomal peptide synthetase IndC by T domain exchange. *Mol. BioSyst.*, 10(7):1709–1718.
- Bhaskara, R. M., de Brevern, A. G., and Srinivasan, N. (2013). Understanding the role of domain–domain linkers in the spatial orientation of domains in multi-domain proteins. *Journal of Biomolecular Structure and Dynamics*, 31(12):1467–1480.
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.
- Bozhüyük, K. A. J., Fleischhacker, F., Linck, A., Wesche, F., Tietze, A., Niesert, C.-P., and Bode, H. B. (2017). De novo design and engineering of non-ribosomal peptide synthetases. *Nature Chemistry*, 10(3):275–281.

- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.
- Caboche, S., Pupin, M., Leclere, V., Fontaine, A., Jacques, P., and Kucherov, G. (2007). NORINE: a database of nonribosomal peptides. *Nucleic Acids Research*, 36(Database):D326–D331.
- Calcott, M. J., Owen, J. G., Lamont, I. L., and Ackerley, D. F. (2014). Biosynthesis of novel pyoverdines by domain substitution in a nonribosomal peptide synthetase of *Pseudomonas aeruginosa*. *Applied and Environmental Microbiology*, 80(18):5723–5731.
- Carhart, R. E., Smith, D. H., and Venkataraghavan, R. (1985). Atom pairs as molecular features in structure-activity studies: definition and applications. *Journal of Chemical Information and Modeling*, 25(2):64–73.
- Chang, Q., Wang, W., Regev-Yochay, G., Lipsitch, M., and Hanage, W. P. (2015). Antibiotics in agriculture and the risk to human health: how worried should we be? *Evolutionary applications*, 8(3):240–7.
- Chemical Computing group (2010). Molecular operating environment. <https://www.chemcomp.com/>. [Online; accessed 17-may-2019].
- Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., Dearden, J., Gramatica, P., Martin, Y. C., Todeschini, R., Consonni, V., Kuz'min, V. E., Cramer, R., Benigni, R., Yang, C., Rathman, J., Terfloth, L., Gasteiger, J., Richard, A., and Tropsha, A. (2014). QSAR modeling: where have you been? Where are you going to? *Journal of medicinal chemistry*, 57(12):4977–5010.
- Chollet, F. et al. (2015). Keras. <https://keras.io>.
- Cimermancic, P., Medema, M., Claesen, J., Kurita, K., Wieland Brown, L., Mavrommatis, K., Pati, A., Godfrey, P., Koehrsen, M., Clardy, J., Birren, B., Takano, E., Sali, A., Lington, R., and Fischbach, M. (2014). Insights into Secondary Metabolism from a Global Analysis of Prokaryotic Biosynthetic Gene Clusters. *Cell*, 158(2):412–421.
- Coëffet-Le Gal, M. F., Thurston, L., Rich, P., Miao, V., and Baltz, R. H. (2006). Complementation of daptomycin dptA and dptD deletion mutations in trans and production of hybrid lipopeptide antibiotics. *Microbiology*, 152(10):2993–3001.
- Crüsemann, M., Kohlhaas, C., and Piel, J. (2013). Evolution-guided engineering of nonribosomal peptide synthetase adenylation domains. *Chemical Science*, 4(3):1041.
- Dejong, C. A., Chen, G. M., Li, H., Johnston, C. W., Edwards, M. R., Rees, P. N., Skinnider, M. A., Webster, A. L. H., and Magarvey, N. A. (2016). Polyketide and nonribosomal peptide retrobiosynthesis and global gene cluster matching. *Nature Chemical Biology*, 12(12):1007–1014.
- Doekel, S., Coëffet-Le Gal, M.-F., Gu, J.-Q., Chu, M., Baltz, R. H., and Brian, P. (2008). Non-ribosomal peptide synthetase module fusions to produce derivatives of daptomycin in *Streptomyces roseosporus*. *Microbiology*, 154(9):2872–2880.
- Duffy, F. J., Verniere, M., Devocelle, M., Bernard, E., Shields, D. C., and Chubb, A. J. (2011). CycloPs: Generating Virtual Libraries of Cyclized and Constrained Peptides Including Nonnatural Amino Acids. *Journal of Chemical Information and Modeling*, 51(4):829–836.

- Durant, J. L., Leland, B. A., Henry, D. R., and Nourse, J. G. (2002). Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280.
- Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461.
- Editor ChemAxon (2011). Chemaxon - standardizer. <https://chemaxon.com/>. [Online; accessed 17-may-2019].
- Editor Daylight (2012). Smarts - a language for describing molecular patterns. <https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html>. [Online; accessed 17-may-2019].
- Editor Daylight (2013). Smirks - a reaction transform language. <https://www.daylight.com/dayhtml/doc/theory/theory.smirks.html>. [Online; accessed 17-may-2019].
- Editor Drug-bank (2010). Drugbank. <https://www.drugbank.ca/drugs/DB00027>. [Online; accessed 17-may-2019].
- Editor RDKit (2013). RDKit: Open-source cheminformatics. <http://www.rdkit.org>. [Online; accessed 17-may-2019].
- Farag, S., Bleich, R. M., Shank, E. A., Isayev, O., Bowers, A. A., and Tropsha, A. (2019). Inter-Modular Linkers play a crucial role in governing the biosynthesis of non-ribosomal peptides. *Bioinformatics*.
- Felnagle, E. A., Jackson, E. E., Chan, Y. A., Podevels, A. M., Berti, A. D., McMahon, M. D., Thomas, M. G., M, A., Berti, A. D., McMahon, M. D., Thomas, M. G., and Podevels, A. M. (2008). Production of Medically Relevant Natural Products Nonribosomal Peptide Synthetases Involved in the Production of Medically Relevant Natural Products. *Molecular Pharmaceutics*, 5(2):191–211.
- Filice, G. A., Nyman, J. A., Lexau, C., Lees, C. H., Bockstedt, L. A., Como-Sabetti, K., Leshner, L. J., and Lynfield, R. (2010). Excess Costs and Utilization Associated with Methicillin Resistance for Patients with Staphylococcus aureus Infection. *Infection Control & Hospital Epidemiology*, 31(4):365–373.
- Fourches, D., Muratov, E., and Tropsha, A. (2010). Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research. *Journal of chemical information and modeling*, 50(7):1189–204.
- George, R. A. and Heringa, J. (2002). An analysis of protein domain linkers: their classification and role in protein folding. *Protein Engineering, Design and Selection*, 15(11):871–879.
- Gokhale, R. S. and Khosla, C. (2000). Role of linkers in communication between protein modules. *Current opinion in chemical biology*, 4(1):22–7.
- Golkar, Z., Bagasra, O., and Pace, D. G. (2014). Bacteriophage therapy: a potential solution for the antibiotic resistance crisis. *The Journal of Infection in Developing Countries*, 8(02):129–36.

- Good, L. and Stach, J. E. M. (2011). Synthetic RNA silencing in bacteria - antimicrobial discovery and resistance breaking. *Frontiers in microbiology*, 2:185.
- Gould, I. M. and Bal, A. M. (2013). New antibiotic agents in the pipeline and how they can help overcome microbial resistance. *Virulence*, 4(2):185–91.
- Haft, D. H., Loftus, B. J., Richardson, D. L., Yang, F., Eisen, J. A., Paulsen, I., and White, O. (2001). TIGRFAMs: a protein family resource for the functional identification of proteins. *Nucleic Acids Research*, 29(1):41–43.
- Hearst, M., Dumais, S., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.
- Kaeberlein, T., Lewis, K., and Epstein, S. S. (2002). Isolating "uncultivable" microorganisms in pure culture in a simulated natural environment. *Science (New York, N.Y.)*, 296(5570):1127–9.
- Keller, U. and Schauwecker, F. (2003). Combinatorial Biosynthesis of Non-Ribosomal Peptides. *Combinatorial Chemistry & High Throughput Screening*, 6(6):527–540.
- Kennedy, D. (2013). Time to deal with antibiotics. *Science (New York, N.Y.)*, 342(6160):777.
- Klevens, R. M., Edwards, J. R., Richards, C. L., Horan, T. C., Gaynes, R. P., Pollock, D. A., and Cardo, D. M. (2007). Estimating Health Care-Associated Infections and Deaths in U.S. Hospitals, 2002. *Public Health Reports*, 122(2):160–166.
- Le, Q. V. and Mikolov, T. (2014). Distributed Representations of Sentences and Documents.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- Ling, L. L., Schneider, T., Peoples, A. J., Spoering, A. L., Engels, I., Conlon, B. P., Mueller, A., Schäberle, T. F., Hughes, D. E., Epstein, S., Jones, M., Lazarides, L., Steadman, V. A., Cohen, D. R., Felix, C. R., Fetterman, K. A., Millett, W. P., Nitti, A. G., Zullo, A. M., Chen, C., and Lewis, K. (2015). A new antibiotic kills pathogens without detectable resistance. *Nature*, 517(7535):455–459.
- Lott, J. S. and Lee, T. V. (2017). Revealing the Inter-Module Interactions of Multi-Modular Nonribosomal Peptide Synthetases. *Structure*, 25(5):693–695.
- Luyt, C.-E., Bréchet, N., Trouillet, J.-L., and Chastre, J. (2014). Antibiotic stewardship in the intensive care unit. *Critical care (London, England)*, 18(5):480.
- Martens, E. and Demain, A. L. (2017). The antibiotic resistance crisis, with a focus on the United States. *The Journal of Antibiotics*, 70(5):520–526.
- Mauldin, P. D., Salgado, C. D., Hansen, I. S., Durup, D. T., and Bosso, J. A. (2010). Attributable hospital cost and length of stay associated with health care-associated infections caused by antibiotic-resistant gram-negative bacteria. *Antimicrobial agents and chemotherapy*, 54(1):109–15.
- Medema, M. H., Kottmann, R., Yilmaz, P., Cummings, M., Biggins, J. B., Blin, K., De Bruijn, I., Chooi, Y. H., Claesen, J., Coates, R. C., Cruz-Morales, P., Duddela, S., Düsterhus, S., Edwards, D. J., Fewer, D. P., Garg, N., Geiger, C., Gomez-Escribano, J. P., Greule, A., Hadjithomas, M., Haines, A. S., Helfrich, E. J. N., Hillwig, M. L., Ishida, K., Jones, A. C., Jones, C. S.,

- Jungmann, K., Kegler, C., Kim, H. U., Kötter, P., Krug, D., Masschelein, J., Melnik, A. V., Mantovani, S. M., Monroe, E. A., Moore, M., Moss, N., Nützmänn, H.-W. W., Pan, G., Pati, A., Petras, D., Reen, F. J., Rosconi, F., Rui, Z., Tian, Z., Tobias, N. J., Tsunematsu, Y., Wiemann, P., Wyckoff, E., Yan, X., Yim, G., Yu, F., Xie, Y., Aigle, B., Apel, A. K., Balibar, C. J., Balskus, E. P., Barona-Gómez, F., Bechthold, A., Bode, H. B., Borriss, R., Brady, S. F., Brakhage, A. A., Caffrey, P., Cheng, Y.-Q. Q., Clardy, J., Cox, R. J., De Mot, R., Donadio, S., Donia, M. S., Van Der Donk, W. A., Dorrestein, P. C., Doyle, S., Driessen, A. J. M., Ehling-Schulz, M., Entian, K.-D. D., Fischbach, M. A., Gerwick, L., Gerwick, W. H., Gross, H., Gust, B., Hertweck, C., Höfte, M., Jensen, S. E., Ju, J., Katz, L., Kaysser, L., Klassen, J. L., Keller, N. P., Kormanec, J., Kuipers, O. P., Kuzuyama, T., Kyrpides, N. C., Kwon, H.-J. J., Lautru, S., Lavigne, R., Lee, C. Y., Linquan, B., Liu, X., Liu, W., Luzhetskyy, A., Mahmud, T., Mast, Y., Méndez, C., Metsä-Ketelä, M., Micklefield, J., Mitchell, D. A., Moore, B. S., Moreira, L. M., Müller, R., Neilan, B. A., Nett, M., Nielsen, J., O’Gara, F., Oikawa, H., Osbourn, A., Osburne, M. S., Ostash, B., Payne, S. M., Pernodet, J.-L. L., Petricek, M., Piel, J., Ploux, O., Raaijmakers, J. M., Salas, J. A., Schmitt, E. K., Scott, B., Seipke, R. F., Shen, B., Sherman, D. H., Sivonen, K., Smanski, M. J., Sosio, M., Stegmann, E., Süßmuth, R. D., Tahlan, K., Thomas, C. M., Tang, Y., Truman, A. W., Viaud, M., Walton, J. D., Walsh, C. T., Weber, T., Van Wezel, G. P., Wilkinson, B., Willey, J. M., Wohlleben, W., Wright, G. D., Ziemert, N., Zhang, C., Zotchev, S. B., Breitling, R., Takano, E., and Glöckner, F. O. (2015). Minimum Information about a Biosynthetic Gene cluster. *Nature Chemical Biology*, 11(9):625–631.
- Meyer, S., Kehr, J.-C., Mainz, A., Dehm, D., Petras, D., Süßmuth, R., and Dittmann, E. (2016). Biochemical Dissection of the Natural Diversification of Microcystin Provides Lessons for Synthetic Biology of NRPS. *Cell Chemical Biology*, 23(4):462–471.
- Michael, C. A., Dominey-Howes, D., and Labbate, M. (2014). The antimicrobial resistance crisis: causes, consequences, and management. *Frontiers in public health*, 2:145.
- Miller, B. R., Sundlov, J. A., Drake, E. J., Makin, T. A., and Gulick, A. M. (2014). Analysis of the linker region joining the adenylation and carrier protein domains of the modular nonribosomal peptide synthetases. *Proteins*, 82(10):2691–702.
- Miyazaki, S., Kuroda, Y., and Yokoyama, S. (2002). Characterization and prediction of linker sequences of multi-domain proteins by a neural network. *Journal of Structural and Functional Genomics*, 2(1):37–51.
- Mootz, H. D., Schwarzer, D., and Marahiel, M. A. (2000). Construction of hybrid peptide synthetases by module and domain fusions. *Proceedings of the National Academy of Sciences of the United States of America*, 97(11):5848–53.
- Nguyen, K. T., Ritz, D., Gu, J.-Q., Alexander, D., Chu, M., Miao, V., Brian, P., and Baltz, R. H. (2006). Combinatorial biosynthesis of novel antibiotics related to daptomycin. *Proceedings of the National Academy of Sciences of the United States of America*, 103(46):17462–7.
- Nichols, D., Cahoon, N., Trakhtenberg, E. M., Pham, L., Mehta, A., Belanger, A., Kanigan, T., Lewis, K., and Epstein, S. S. (2010). Use of ichip for high-throughput in situ cultivation of “uncultivable” microbial species. *Applied and environmental microbiology*, 76(8):2445–50.
- Nilakantan, R., Bauman, N., Dixon, J. S., and Venkataraghavan, R. (1987). Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *Journal of Chemical Information and Modeling*, 27(2):82–85.

- O'Boyle, N. M., Morley, C., and Hutchison, G. R. (2008). Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit. *Chemistry Central Journal*, 2(1):5.
- O'Boyle, N. M. and Sayle, R. A. (2016). Comparing structural fingerprints using a literature-based similarity benchmark. *Journal of Cheminformatics*, 8(1):36.
- Pang, T., Huang, L., Deng, Y., Wang, T., Chen, S., Gong, X., and Liu, W. (2017). Logistic regression analysis of conventional ultrasonography, strain elastosonography, and contrast-enhanced ultrasound characteristics for the differentiation of benign and malignant thyroid nodules. *PLOS ONE*, 12(12):e0188987.
- Patterson, T. J., Ngo, M., Aronov, P. A., Reznikova, T. V., Green, P. G., and Rice, R. H. (2003). Biological Activity of Inorganic Arsenic and Antimony Reflects Oxidation State in Cultured Human Keratinocytes. *Chemical Research in Toxicology*, 16(12):1624–1631.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Piddock, L. J. V. (2012). The crisis of no new antibiotics—what is the way forward? *The Lancet. Infectious diseases*, 12(3):249–53.
- Pupin, M., Esmaeel, Q., Flissi, A., Dufresne, Y., Jacques, P., and Leclère, V. (2016). Norine: A powerful resource for novel nonribosomal peptide discovery. *Synthetic and Systems Biotechnology*, 1(2):89–94.
- Reger, A. S., Carney, J. M., and Gulick, A. M. (2007). Biochemical and crystallographic analysis of substrate binding and conformational changes in acetyl-CoA synthetase. *Biochemistry*, 46(22):6536–46.
- Roberts, R., Hota, B., Ahmad, I., Scott II, R., Foster, S., Abbasi, F., Schabowski, S., Kampe, L., Ciavarella, G., Supino, M., Naples, J., Cordell, R., Levy, S., and Weinstein, R. (2009). Hospital and Societal Costs of Antimicrobial-Resistant Infections in a Chicago Teaching Hospital: Implications for Antibiotic Stewardship. *Clinical Infectious Diseases*, 49(8):1175–1184.
- Robinson, C. R. and Sauer, R. T. (1998). Optimizing the stability of single-chain proteins by linker length and composition mutagenesis. *Proceedings of the National Academy of Sciences of the United States of America*, 95(11):5929–34.
- Rogers, D. and Hahn, M. (2010). Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754.
- Spellberg, B., Blaser, M., Guidos, R. J., Boucher, H. W., Bradley, J. S., Eisenstein, B. I., Gerding, D., Lynfield, R., Reller, L. B., Rex, J., Schwartz, D., Septimus, E., Tenover, F. C., Gilbert, D. N., and (IDSA), I. D. S. o. A. (2011). Combating antimicrobial resistance: policy recommendations to save lives. *Clinical infectious diseases : an official publication of the Infectious Diseases Society of America*, 52 Suppl 5(Suppl 5):S397–428.
- Stevens, B. W., Joska, T. M., and Anderson, A. C. (2005). Progress toward re-engineering non-ribosomal peptide synthetase proteins: a potential new source of pharmacological agents. *Drug Development Research*, 66(1):9–18.

- Süssmuth, R. D. and Mainz, A. (2017). Nonribosomal Peptide Synthesis-Principles and Prospects. *Angewandte Chemie International Edition*, 56(14):3770–3821.
- Suyama, M. and Ohara, O. (2003). DomCut: prediction of inter-domain linker regions in amino acid sequences. *Bioinformatics (Oxford, England)*, 19(5):673–4.
- Tanaka, T., Kuroda, Y., and Yokoyama, S. (2003). Characteristics and prediction of domain linker sequences in multi-domain proteins. *Journal of Structural and Functional Genomics*, 4(2/3):79–85.
- Tarry, M. J., Haque, A. S., Bui, K. H., and Schmeing, T. M. (2017). X-Ray Crystallography and Electron Microscopy of Cross- and Multi-Module Nonribosomal Peptide Synthetase Proteins Reveal a Flexible Architecture. *Structure*, 25(5):783–793.e4.
- Todeschini, R. and Consonni, V. (2000). *Handbook of Molecular Descriptors. Methods and Principles in Medicinal Chemistry*. Wiley-VCH Verlag GmbH, Weinheim, Germany.
- Tropsha, A. (2010). Best Practices for QSAR Model Development, Validation, and Exploitation. *Molecular Informatics*, 29(6-7):476–488.
- Udwary, D. W., Merski, M., and Townsend, C. A. (2002). A Method for Prediction of the Locations of Linker Regions within Large Multifunctional Proteins, and Application to a Type I Polyketide Synthase. *Journal of Molecular Biology*, 323(3):585–598.
- van Rossum, G. (1995). Python tutorial.
- Ventola, C. L. (2015a). The antibiotic resistance crisis: part 1: causes and threats. *P & T : a peer-reviewed journal for formulary management*, 40(4):277–83.
- Ventola, C. L. (2015b). The antibiotic resistance crisis: part 2: management strategies and new agents. *P & T : a peer-reviewed journal for formulary management*, 40(5):344–52.
- Viswanathan, V. K. (2014). Off-label abuse of antibiotics by bacteria. *Gut microbes*, 5(1):3–4.
- Wall, L., Christiansen, T., and Orwant, J. (2000). *Programming Perl*. O'Reilly.
- Weber, T., Blin, K., Duddela, S., Krug, D., Kim, H. U., Brucoleri, R., Lee, S. Y., Fischbach, M. A., Muller, R., Wohlleben, W., Breitling, R., Takano, E., and Medema, M. H. (2015). antiSMASH 3.0—a comprehensive resource for the genome mining of biosynthetic gene clusters. *Nucleic Acids Research*, 43(W1):W237–43.
- Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling*, 28(1):31–36.
- Winn, M., Fyans, J. K., Zhuo, Y., and Micklefield, J. (2016). Recent advances in engineering nonribosomal peptide assembly lines. *Nat. Prod. Rep.*, 33(2):317–347.
- Wriggers, W., Chakravarty, S., and Jennings, P. A. (2005). Control of protein functional dynamics by peptide linkers. *Biopolymers*, 80(6):736–46.
- Wu, R., Reger, A. S., Lu, X., Gulick, A. M., and Dunaway-Mariano, D. (2009). The mechanism of domain alternation in the acyl-adenylate forming ligase superfamily member 4-chlorobenzoate: coenzyme A ligase. *Biochemistry*, 48(19):4115–25.

Young, D., Martin, T., Venkatapathy, R., and Harten, P. (2008). Are the Chemical Structures in Your QSAR Correct? *QSAR & Combinatorial Science*, 27(11-12):1337–1345.

Yu, D., Xu, F., Gage, D., and Zhan, J. (2013). Functional dissection and module swapping of fungal cyclooligomer depsipeptide synthetases. *Chemical Communications*, 49(55):6176.