DEMOCRATISING DEEP LEARNING IN MICROBIAL METABOLITES RESEARCH

DEMOCRATISING DEEP LEARNING IN MICROBIAL METABOLITES RESEARCH

By KESHAV DIAL, HBSc, OCGC

A Thesis Submitted to the School of Graduate Studies in Partial Fulfilment of the Requirements for the Degree: Doctor of Philosophy

McMaster University Doctor of Philosophy (2023) Hamilton, Ontario (Biochemistry and Biomedical Sciences)
TITLE: Democratising Deep Learning in Microbial Metabolite Research. AUTHOR: Keshav Dial, Honours BSc (The University of Western Ontario), Ontario College Graduate Certificate (Seneca College). SUPERVISOR: Associate Professor Dr Nathan A. Magarvey.

Number of pages: 201.

Abstract

Deep learning models are dominating performance across a wide variety of tasks. From protein folding to computer vision to voice recognition, deep learning is changing the way we interact with data. The field of natural products, and more specifically genomic mining, has been slow to adapt to these new technological innovations. As we are in the midst of a data explosion, it is not for lack of training data. Instead, it is due to the lack of a blueprint demonstrating how to correctly integrate these models to maximise performance and inference. During my PhD, I showcase the use of large language models across a variety of data domains to improve common workflows in the field of natural product drug discovery. I improved natural product scaffold comparison by representing molecules as sentences. I developed a series of deep learning models to replace archaic technologies and create a more scalable genomic mining pipeline decreasing running times by 8X. I integrated deep learning-based genomic and enzymatic inference into legacy tooling to improve the quality of short-read assemblies. I also demonstrate how intelligent querying of multi-omic datasets can be used to facilitate the gene signature prediction of encoded microbial metabolites. The models and workflows I developed are wide in scope with the hopes of blueprinting how these industry standard tools can be applied across the entirety of natural product drug discovery.

Acknowledgements

First I would like to thank my supervisor Nathan. You took a huge risk by hiring a student with no machine learning experience, no microbe experience and no background in chemistry. For all of the guidance and resources you have provided to help my growth, I thank you.

I'd like to thank my committee members, Andrew McArthur and Brian Golding. I would not have continued my graduate studies without your support. Thank you for making me a better scientist.

I'd like to thank my lab mates, especially Mathusan, Norman, Emily and Irina. You made every day at work an absolute joy.

Lastly, I wish to thank my family and friends for their encouragement during my graduate career. To Ambi, thank you for making me go back to school to pursue what I love. Mum and Dad, this work is as much yours as it is mine.

Table of Contents

Abstract iv

Acknowledgements v

Table of Contents vi

List of Tables and Figures xi

List of Abbreviations and Symbols xix

Declaration of Academic Achievement xxii

Chapter 1. Introduction 1

1.1 Natural Product Drug Discovery 1

1.2 Bioinformatics and the Genomic Mining Era 7

1.3 Natural Language Processing 12

1.4. Scope and nature of this work 19

1.4.1 Improving Molecular Comparison with Deep Learning 20

1.4.2 Improving Genomic Mining with Deep Learning 20

1.4.3 Improving Activity Prediction with Integrated Data and Statistics 21

1.4.4. Thesis Overview 22

1.5 Figures and Tables 23

1.6 Bibliography 24

Chapter 2: NP-BERT - An NLP approach to natural product comparison 32

2.1 Chapter Preface 32

2.2 Abstract 33

2.3 Introduction 33

2.3 Methodology 37

2.3.1 Model Architecture 37

2.3.2 Input/Outputs 38

2.3.3 Pretraining 38

2.3.4 Linked Task Fine-tuning 39

2.3.6 Quantisation, Optimisation and Acceleration 40

2.3.6 Experiments 40

2.3.6.4 Conserved Biosynthesis within Clusters 43

2.4 Results 44

2.4.1 Linked Task Fine-tuning 44

2.4.2 Comparison to NP Classifier 45

2.4.3 Natural Product Label Recovery using Embedding Distances and Fingerprint Dissimilarities 45

2.4.4 Natural Product Latent Space 46

2.4.5 Conserved Biosynthesis 46

2.5 Discussion 47

2.5.1 Finetuning LLM with Linked Task Training and Active Learning 47

2.5.2 Effectiveness as a new molecular comparison metric 49

2.5.3 Future Work 50

2.6 Figures and Tables 52

2.7 References 60

Chapter 3: Mining the Biosynthetic Universe using Large Language Models. 63

3.1 Chapter Preface 63

3.2 Abstract 64

3.3 Introduction 65

3.4 Methodology 68

3.4.1 Adenylation and Acyltransferase T5 68

3.4.2 PK Domain T5 Models 73

3.4.3 Enzyme-BERT 75

3.5 Results 87

3.5.1 Adenylation and Acyltransferase T5 87

3.5.2 Polyketide Domain T5 Models 88

3.5.3 Enzyme BERT 89
3.6 Discussion 92
3.6.1 Improving Scalability of Genomic Mining Using Transformers 92
3.6.2 Adenylation Domain Substrate Prediction with Transformers 94
3.6.3 Transformer Attention in Enzymology 96
3.6.3 Gene Cluster Comparison with IBIS 97
3.6.4 Future Work 100
3.7 Figures and Tables 101
3.7 Bibliography 118
Chapter 4: Deep Learning Guided De Novo Assembly of Bacterial Genomes 122
4.1 Chapter Preface 122
4.2 Abstract 123
4.3 Introduction 124
4.3 Methods 128
4.3.1 Bacterial T5 128
4.3.2 Network Graph Link Prediction with EnzymeBERT and RevGNN 131
4.3.3 Integration of Neural Network Inference into the Unicycler Framework 133
4.3.4 Experiments 136
4.4 Results 138
4.4.1 Bacterial T5 Training 138
4.4.2 EnzymeGNN Training 139
4.4.3 Unicycler Hyperparameter Optimisation 139
4.4.4 Comparison of different assembly strategies 140
4.4.5 Hybrid Assembly of Publicly Available Genomes 140
4.4.6 Chymostatin Biosynthetic Gene Cluster Discovery 141
4.5 Discussion 142
4.5.1 Deep Learning Enhanced Assembly 142

4.5.2 Rescue of Illumina Short-Read Assemblies with Targeted Sequencing 144
4.5.3 Future work 145
4.5 Figures and Tables 146
4.6 Bibliography 161
Chapter 5: Prediction of Metabolite Activity 165
5.1 Chapter Preface 165
5.2 Abstract 166
5.3. Introduction 166
5.4 Methodology 169
5.4.1 Mining the Human Gut Metatranscriptome for Biosynthetic Gene Clusters 169
5.4.2 Mining the Human Gut Metabolome for Microbial Metabolites 170
5.4.3 Linear Regression of Faecalibacterium prausnitzii compound and SSCAI 171
5.4.4 Mining the Human Gut Proteome for Bacteriocins 172
5.4.4.2 Mining Scardovia Bacteriocin Abundencies from Human Microbiome 173
5.4.5 Mango Processing of Samples 174
5.5.6 Supplementary Methods - Antimicrobial activity and determination of the minimum inhibitory concentration of Scardovia wiggsiae bacteriocin 175
5.5 Results 178
5.5.1 BGC Gene Signatures 178
5.5.1.2 Inflammation and Fatty Acid Oxidation 180
5.5.2 Metabolite Enriched Terms 181
5.5.3 Metabolite Relationship with Ulcerative Colitis 182
5.5.4. RiPP Transformer 182
5.5.5 Scardovia wiggsiae peptide relatedness 183
5.5.6. Peptide Enriched terms 183
5.6 Discussion 184
5.7 Figures and Tables 188

5.8 References 194

Chapter 6: Significance and future prospective 198

List of Tables and Figures

Chapter 1. Introduction **1**

Figure 1.1: The database of molecules from NPAtlas, binned by isolation publication year. The plot was created using Tableau Desktop. The amount of molecules isolated over the decades has continued to trend upward.....23

Chapter 2: NP-BERT - An NLP approach to natural product comparison **32**

Table 2.1 Accuracy of different fingerprinting techniques, measured using conserved labels of nearest neighbours for each molecule in the validation dataset (n = 12,534). The Euclidean distance of NPBERT embeddings performed best at the label recovery task.52

Figure 2.1 Example of Natural Product Classification Recovery with FCFP6 versus NP-BERT. (A) Non-redundant FCFP6 with Jaccard dissimilarities found an incorrect nearest neighbour. All classifications were not recovered. (B) NP-BERT with Euclidean distances was able to successfully find a derivative of the query molecule with all classifications recovered. 53

Figure 2.2: The natural product latent space of NP-BERT demonstrated with a dataset of molecules (n=49,523) projected to two dimensions and plotted with UMAP and data shader. Molecules are colourised by the predicted pathway labels. The molecules are separated into clusters of distinct pathways with minimal overlap. 54

Figure 2.3: Upon inspection of one of the “polyketide” HDBSCAN clusters, it consisted only of tetracyclines. An internal dataset of crude extracts was used to visualise the co-occurrence of the metabolites within fermentations. To create the network graph, each tetracycline was represented as a node and upon co-occurrence with another metabolite, an edge was drawn. Many of the tetracyclines within the HDBSCAN cluster are co-expressed. 55

Figure 2.4 Upon inspection of one of the "amino acids and peptides” HDBSCAN clusters, it consisted only of thiazoles. An internal dataset of crude extracts was used to visualise the co-occurrence of the metabolites within fermentations. To create the network graph, each thiazole was represented as a node and upon co-occurrence with another metabolite, an edge was drawn. Many of the thiazoles within the HDBSCAN cluster are co-expressed. 56

Figure 2.5 Upon inspection of one of the “polyketide” HDBSCAN clusters, it consisted mainly of natural statins. An internal dataset of crude extracts was used to visualise the co-occurrence of the metabolites within fermentations. To create the network graph, each

metabolite was represented as a node and upon co-occurrence with another metabolite, an edge was drawn. Many of the molecules within the HDBSCAN cluster are co-expressed. 57

Supplementary Figure S2.1(A) Huggingface's implementation of the RoBERTa For Sequence Classification used the emissions from a base model (size = length of the input sequence x hidden size) and passes it through two linear transformation layers. Because RoBERTa and T5 lack a pooling layer, the first layer of the classification acts as such changing the matrix to a square using a linear transformation with an additive bias (size = hidden size x hidden size). After a Tan-H activation, the embedding is passed through the dense layer. This is another linear transformation layer but it reshapes the matrix to become a flat logit vector (size = the number of classes). A softmax function normalises the logits into probabilities. Any inconsistencies are measured typically with cross-entropy loss and back-propagated through the network. (B) In multi-task learning, the three heads generate logits in parallel. Biases are only shared within the transformer. (C) In linked-task learning, the pooling layer's square output is passed in succession, giving the proceeding classification heads more biases to utilise. 58

Supplementary Figure S2.2 A training loss curve across all training epochs for each of the labels. With Linked-Task Learning, the loss for pathway and superclass prediction reaches loss values below 1.0 within 200 steps. Class labelling takes much longer but does eventually reach a similar value. 59

Chapter 3: Mining the Biosynthetic Universe using Large Language Models. 63

Figure 3.1: The entire IBIS pipeline is broken into 8 steps: (1) Open Reading Frame (ORF) Prediction - Pyrodigal is used to find all likely protein sequences per contig (2) Domain Annotation - Individual protein sequences are annotated with domain-level annotations through use of EnzymeBERT, a token classification head and two graph-based algorithms: Community-Based Polishing and Greedy Grouping (3) Whole Protein Annotation - Each protein sequence is annotated with a known biosynthetic functionality in the form of EC# or specific gene class (e.g. vanA) (4) Proximity-based BGC Boundary Calling - Based on the biosynthetic families associated with the annotations, the proteins are merged into BGCs using greedy grouping and a rule set. (5) Domain Substrate and Functionality Prediction - Any adenylation, acyltransferase or polyketide domains are further processed by the other T5 models for substrate prediction and determining whether or not the domain is functional. (6) Molecule Unit Prediction -

Using BEAR molecular units can be predicted from the peptide annotations within each putative BGC.....101
Figure 3.2 Example of Structure Prediction using BEAR and IBIS. The rich library of annotations provided by the IBIS pipeline facilitates the prediction of the molecular units produced by a biosynthetic gene cluster. The visualisation was created using the Natural Product Toolkit web application.....102
Table 3.1 A table summarising the Adenylation T5 model's performance across the different classification tasks as outputted from each individual classification head. With the exception of LogP and the substrate sequence classification heads, each task achieves over 85% accuracy in the validation set.....103
Table 3.2 A table summarising the Acyltransferase T5 model's performance across the different classification tasks as outputted from each individual classification head. The Acyltransferase T5 model performed well across all tasks and both datasets, with the exception of the CH3 prediction. 104
Figure 3.3: Radar plots of validation accuracy of the different substrate prediction strategies using embeddings from the model. (A) The standard sequence classification predicted labels. This uses a linear feed-forward layer to fit the embeddings to logits of the classes trained on. (B) An ensemble model trained using Microsoft's Explainable Boosted Machine (EBM) implementation. It used the predicted properties as input. (C) The FAISS nearest neighbours strategy. Across substrates, the nearest neighbours strategy outperformed or shared similar performance as the other two strategies. 105
Figure 3.4 Charts showing the changing pooled loss, functional accuracy and clade accuracy for the different Polyketide T5 Models. Some models were stopped early due to increased validation loss. Thiolation domains were not trained on clade accuracy. 108
Table 3.3 Performance of the Nearest Neighbour lookup strategy with the polyketide domain models. F1 Scores were above 0.7 for all models in determining whether or not a polyketide domain was functional. 109
Table 3.4 The performance of EnzymeBERT in predicting enzyme commission numbers, protein families and protein domains. Protein families and enzyme commission number labels were assigned using the nearest neighbour semantic search strategy. Protein domains were assigned using the token classification head. 110
Figure 3.5: The combined attention map for different positions along the Acetyltransferase conserved domain (length of 48 amino acids). The global alignment of the 535 sequences, spans 131 indices. (A) When embedding the first residue in each sequence

(i=1), residues within the first seven global positions are mainly attended to. (B) When embedding the 15th residue (i=15), attention is more widespread, with attention mainly spanning from the 10th position to the 40th position. (C) Attention heads for the 30th residue (i=30) and (D) at the 48th residue (i=48), displayed local attention. The attention of residues at a global index position smaller than 52 was negligible. (E) The sequence logo of the acetyltransferase HMM's alignment showed a few gaps. The gaps in the global alignment resulted in exaggerated local attention windows.....111

Figure 3.6 Radar plots summarising the chemical validity of BGC comparison methods using a dataset of BGCs with experimentally validated metabolites (n=441). (A) Triplets were derived from the FCFP6 featurization of molecules and dissimilarity was calculated using the Jaccard Index. (B) Triplets were derived using the NP-BERT embeddings of the molecules and the Euclidean distance was calculated. In blue, BGCs are treated as sets using features derived from IBIS; dissimilarity is calculated using the Jaccard Index. In orange, BGCs are treated as the averaged vector of EnzymeBERT embeddings. When a triplet relationship is maintained (the distance of a positive example to the anchor is less than the distance of a negative example to the anchor), a point is awarded. The accuracy is categorised by the superclass of the encoded metabolites. 112

Figure 3.7 The internal dataset of biosynthetic gene clusters was embedded using EnzymeBERT, (n = 296,216) projected to two dimensions (trustworthiness = 0.999), and plotted using UMAP and datashader. Gene clusters were also projected to 128 dimensions (trustworthiness = 0.999) and clustered using HDBSCAN (silhouette score = 0.402). A total of 231,772 BGCs were clustered into 12,495 gene cluster families (GCFs). 113

Figure 3.8 Example GCF matching the Curamycin A gene cluster. The original producer of Curamycin A is Streptomyces cyaneus. The GCF spans: Streptomyces katrae s3, Streptomyces lavendulae NRRL B-2774, Streptomyces viridosporus DSM 40243, Streptomyces sp. NRRL S-87, Streptomyces sp. AG109 G2-1, Streptomyces sp. RU-71, Streptomyces sp. NWU-339, Streptomyces sp. TRM-SA0054. (A) The taxonomic network graph showed no conservation on a species level, with the only matches being Streptomyces strains without a designated species. (B) The reference Curamycin A gene cluster was nearly identical to the reference BGC with the exception of a different EC number. (C) The Curamycin A metabolite was detected in the metabolomics analysis. 114

Figure 3.9 Example GCF matching the erythromycin gene cluster. The original producer of Erythromycin is Saccharopolyspora erythraea NRRL 2338. The GCF spans multiple genera including: Saccharopolyspora erythraea NRRL 2338, Saccharopolyspora erythraea DSM 41009, Saccharopolyspora erythraea DSM 40517, Saccharomonospora paurometabolica YIM 90007, Aeromicrobium erythreum AR18, Micromonospora rosaria DSM 803, Streptomyces noursei ATCC 11455, Streptomyces yunnanensis CGMCC 43555, Streptomyces sp. MG1, Streptomyces sp. NRRL F5193, Streptomyces sp. NWU49, and Streptomyces sp. CB02120-2. (A) The taxonomic network graph showed the GCF is conserved mainly between two genera Streptomyces and Saccharopolyspora. (B) There are minor changes between the Streptomyces representative versus the Saccharopolyspora reference BGC in the first polyketide synthase. (C) Erythromycin G was detected in the alternative Saccharopolyspora erythraea strain in the metabolomics analysis. 115

Figure 3.10 Example GCF matching the Polymyxin gene cluster. The original producer of Polymyxin is Paenibacillus polymyxa. The GCF spans: Paenibacillus forsythiae T98, Paenibacillus aquistagni Strain 11, Paenibacillus alvei A6-6I-X, Paenibacillus sp. IHBB 10380, Paenibacillus sp. UNC217MF, Paenibacillus sp. ST-S, Paenibacillus sp. C16COL, Paenibacillus alvei DSM 29, Paenibacillus pinihumi DSM 23905, Brevibacillus brevis X23, Brevibacillus sp. BC25, Brevibacillus brevis DZQ7, Brevibacillus brevis ATCC 35690, Brevibacillus brevis GZDF31, Brevibacillus sp. NRRL B-41110 and Brevibacillus brevis NBRC 100599-47. (A) All members of this GCF were mined from the family Paenibacillaceae, so all edges have some weight. There is only a light separation between Paenibacillus and Brevibacillus mined BGCs. (B) The differences between the reference BGC versus the representative from Paenibacillus alvei are minor; two adenylation domains have different substrates predicted in the first two NRPSs, possibly due to error. (C) Polymyxin B was confirmed to be produced by Paenibacillus alvei in the metabolomics analysis. 116

Supplementary Figure S3.1: Joint histogram and density plots for different combinations of BGC and molecular metrics. Histograms are on the outside of the upper and right axis. Density plots are plotted within. 117

Chapter 4: Deep Learning Guided De Novo Assembly of Bacterial Genomes 122

Figure 4.1 A stacked barplot showing the distribution of bacterial RefSeq genomes sequenced with a single platform, categorised by the submission year. There are thousands of

legacy RefSeq genomes assembled with exclusively short-read Illumina sequencing technology. To compile the figure, RefSeq metadata was downloaded from the NCBI and manually cleaned using Tableau's Prep Builder. The cleaned dataset was plotted using Tableau Desktop. 146
Figure 4.2 A stacked barplot showing the distribution of contig N50s for bacterial RefSeq genomes sequenced with a single platform. The majority of low N50 bacterial RefSeq genomes are those sequenced with exclusively Illumina short-read technology. To compile the figure, RefSeq metadata was downloaded from the NCBI and manually cleaned using Tableau's Prep Builder. The cleaned dataset was plotted using Tableau Desktop. 147
Figure 4.3 Training and validation loss curves for span masked language modelling of the BacterialT5 model. After 7 epochs, the loss had plateaued. A training session was started with a new dataset of genomes but was stopped early after no change loss was observed. 148
Figure 4.4 Validation accuracy and validation loss curves for the next sentence prediction tasks. The NSP on open reading frame data plateaued within 3 epochs. The NSP on intergenic regions began to gain loss at the 14th epoch and was stopped early. 149
Figure 4.5 Validation Accuracy and Loss of the RevGNN model with 40 and 20 Layers. The maximum distance between NSP ORFs is 10,000 BP. 150
Table 4.1 Bayesian Hyperparameter Optimization result from sweeps. The most important factor for the FuzzyAlign scores was the minimum edge probability to declare whether or not a RevGNN predicted link was true. The higher the minimum link probability increased the accuracy of the bridge qualities and thereby increased the overall score. RevGNN bridges introduce the least amount of misassemblies; increasing the weight of the bridges and decreasing the cut-off would allow for more RevGNN bridging. As BacterialT5 bridging introduced the most misassemblies, it would be beneficial to decrease the weights and increase the cutoffs, as quantified by the correlation scores. 151
Figure 4.6 A sankey graph showing the relationship between maximising for Mean Segment Count, N50, Open Reading Frame Count and FuzzyAlign. Maximising for metrics other than FuzzyAlign resulted in more misalignments.....152
Figure 4.7 Bayesian optimisation of the bridge parameters to maximise the FuzzyAlign score. The maximum FuzzyAlign score observed was 0.424. The optimisation was stopped after 783 iterations. 153

Table 4.2 Averaged assembly statistics of an in-silico Illumina MiSeq sequenced dataset of (n=521) Biosynthetic Gene Clusters with at least 10x coverage calculated using QUAST.....154
Figure 4.8 Bandage Plots for various assembly methods on DSM 2224. (A) The optimal SPADES assembly was created using a k-mer of 127 (B) The unicycler hybrid assembly used PacBio reads and Illumina short-reads alone. (C) NALA pipeline with only Illumina short-reads (D) NALA pipeline with PacBio long reads and Illumina short-reads. 155
Table 4.3 QUAST assembly statistics of the hybrid reassemblies using the deep-learning guided assembly pipeline versus the original publicly available genome. 156
Table 4.4 Gene Cluster Assembly Statistics Generated Using PRISM for Publicly Available Genomes versus New Genome. 157
Figure 4.9 The recovered NRP-PK Hybrid biosynthetic cluster from DSM 2224 using the deep learning guided assembly. The fragmented gene clusters from the hybrid assembly alone are highlighted by their corresponding colours.....158
Figure 4.10 The Chymostatin BGC discovered within Streptomyces orinoci DSM 40571 (contig 3 at 180,440 to 306,289). Using IBIS-LLMs, the adenylation domain substrates were predicted. Phenylalanine and Valine were both correctly predicted by the Adenylation T5 model. 159
Table 4.5 Chymostatin biosynthetic gene cluster found in Streptomyces mobaraensis and related peptide sequences discovered in Streptomyces orinoci DSM 40571 using EnzymeBERT euclidean distances. The only gene not found was the transcriptional regulator cstA. The relative sizes of the gene clusters were conserved with cstB-G spanning 13,820bp in the source genome versus 13,667 in DSM 40571.....160

Chapter 5: Prediction of Metabolite Activity 165

Figure 5.1 Visualization of the MANGO statistical pipeline for a single biosynthetic gene cluster. (A) Using the least squares approach, individual biosynthetic ORFs are fit to a linear regression model, where each human gene is the dependent variable and the normalized quantity of ORF transcript is the independent variable. (B) The significance of each model is combined using Fisher's combined probability test creating p-values representative of the relationship between the whole BGC to the human genes. (C) Due to the high number of comparisons being made, these values are corrected using the Holm-Sidak Post-hoc test. (D) A level of significance is chosen and the corrected p-values are assessed. Genes that pass are considered a part of the gene signature. (E) The

PANTHER Enrichment web API is used to perform a gene ontology enrichment analysis. It will report the most enriched GO terms and a level of significance for a provided list of genes. 188
Figure 5.2: Structure of (A) F. Prau compound and (B) Butyrate. 189
Figure 5.3 The Faecalibacterium prausnitzii compound's abundance across different patient subgroups. While it is mostly found in healthy patients, there are cases when the metabolite is also found in large amounts in ulcerative colitis patients.....190
Figure 5.4 The RiPP Classification head was able to hit near-perfect performance within the first 100 training steps when trained alone. When trained with the Propeptide trimming head, the performance of this task was degraded. This degradation was evident even when pretrained to near-perfect performance before being returned with the trimming head. 191
Figure 5.5 The Propeptide trimming head struggled to reach past 65% accuracy when trained alone. When combined with the RiPP classification head, the shared information brought near-perfect performance within 100 steps.....192
Table 5.1 The Scardovia peptide aligned with PRANK against its nearest neighbours according to the ProtT5 model fine-tuned for RiPP compounds. Percentage identities and alignments were visualised with the EBI's MView web tool. 193

Chapter 6: Significance and future prospective **198**

List of Abbreviations and Symbols

A	Adenylation
AT	Acyltransferase
Attention Head	Mechanism used by transformer to weigh the influence of different tokens when generating an embedding. Multiple attention heads are used by a transformer create robust embeddings.
BERT	Bidirectional Encoder Representations from Transformers; uses the encoder side of a transformer.
BGC	Biosynthetic Gene Cluster
Clustering	Grouping of unlabelled examples based on similarity. Typically a distance matrix is used to entity relatedness.
Convolution	A convolving operation to merge data between colocalized features; it allows connected nodes to share information similar to attention.
DH	Dehydrotase
Dimensionality Reduction	A mathematical operation to transform a high dimensional space to a lower one.
ECFP	Extended-Connectivity Fingerprint
ELMo	Embeddings from Language Model; uses a bidirectional LSTM.
Embedding	A numerical vector representation of an object; typically how a deep learning model will output an input sequence.
ER	Enoylreductase
FAISS	Facebook AI Similarity Search
FCFP	Functional-Class Fingerprints
Finetuning	Training the model with a complex task to refine the latent space of a model. Typically a supervised task.
GCF	Gene Cluster Family
GCN	Graph Convolution Network - A deep learning model that uses convolutions to perform learning on a network graph.
GNN	Graph Neural Network - A deep learning model structured for learning a network graph.
GPU	Graphical Processing Unit. GPU accelerated deep learning is often faster due the number of cores available versus conventional CPU processing. GPUs contain hundreds of small cores versus a CPU's typical 16.

HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise; an effective density based clustering algorithm.
HF	Hugging Face; company that makes many of the transformer based frameworks.
IBIS	Intergrated Biosynthetic Informatics Suite
kNN	k-nearest neighbours search. Algorithm for returning the closest points equal to the number defined as k.
KR	Ketoreductase
KS	Ketosynthase
LLM	Large language Models (e.g. BERT, T5, transformers)
LSH	Locality Sensitive Hashing
MANGO	Microbiome Associated Natural-product Gene Ontology
MinHash	Min-wise independent permutations locality sensitive hashing scheme; a data sketch structure used for approximating Jaccard Indices
MLM	Masked Language Modelling. Words are randomly masked and a model is taught to “fill in the blanks”.
Multi-omics	An analysis in which the data sets are of multiple “omes” (ex. Proteomics, metabolomics, genomics, transcriptomics)
NALA	Neural-network-guided Arrangement and Linkage Assembly
Nearest Neighbours	Points closest to a query defined by a distance metric.
NLP	Natural Language Processing
NP	Natural Products
NP-BERT	Natural Products BERT
NRP	Non-Ribsomal Peptide
NRPS	Non-Ribosomal Peptide Synthase
PK	Polyketide
PKS	Polyketide Synthase
Pretraining	Training a model with a preliminary task to prime the latent space of a model. Typically a self-supervised task.
PRISM	PRediction Informatics for Secondary Metabolomes; the genomic mining tool designed by the Magarvey Lab.

RiPP	Ribosomally synthesized and post-translationally modified peptides
RoBERTa	Robustly Optimized BERT Pretraining Approach
Self-attention	Using attention heads, the transformer attends to tokens within a sequence.
SELFIES	Self-Referencing Embedded Strings
Semantic Search	Searching using meaning rather than literal matches (ex. Lexical search). Typically refers to using embeddings representations rather than the physical sequence.
Sequence classification	A fine tuning task for classifying the entire sequence of tokens with a single label.
SMILES	Simplified molecular-input line-entry system. A string representation of the graph structure of a molecule.
Span MLM	Span-masked language modelling. Spans of text are randomly enumerated and masked; the model is taught to return the number followed by the masked span of text.
T5	Text-to-Text Transfer Transformer
Token classification	A fine tuning task for classifying each individual token within a sequence with a label.
Tokenizer	A tool used to convert strings into tokens. Typically, it breaks sentences into words. Words are then numerically represented for input into a model.
Transformers	An encoder-decoder architecture, typically used for language modelling.
ULMFit	Universal Language Model Fine-tuning
UMAP	Uniform Manifold Approximation and Projection for Dimension Reduction

Declaration of Academic Achievement

This thesis is formatted as a sandwich thesis, with specific details relating to each contribution described in the corresponding chapter preface.

Chapter 1. Introduction

The cross-discipline research presented in this thesis requires a comprehensive understanding of natural products, bioinformatics and natural language processing. The importance of natural products in the field of medicine as well as the traditional techniques for their discovery are presented in Section 1.1. The impact of bioinformatics since its conception and its role in creating the exponential growth of biological data is presented in Section 1.2. The innovations in natural language processing resulting in revolutionary technologies such as ChatGPT and potential applications in natural product research are presented in Section 1.3.

1.1 Natural Product Drug Discovery

Natural products can be characterised as organic molecules synthesised by living organisms. Primary metabolites are the molecules essential for life such as DNA, RNA and proteins. Secondary metabolites are the molecules used by microbes to interact with their environment. Microbial secondary metabolites exhibit a wide variety of bioactivities. To chelate and accelerate iron absorption they use siderophores such as enterobactin.[1] Through the release of quorum-sensing molecules, such as acyl-homoserine lactone, bacteria can communicate with each other.[2] With the release of antifungal and antibacterial agents, neighbouring microbes can be killed and disrupted.

The myriad of specialised activities the metabolites exhibit has made them favourable drug candidates.[3]

Microbial natural products have played a pivotal role in the development of modern medicine. The first tuberculosis treatment was the aminoglycoside Streptomycin, isolated from *Streptomyces griseus* in 1943. [4] The discovery of the immunomodulating non-ribosomal peptide cyclosporine, isolated from the fungus *Tolypocladium inflatum*, enabled organ transplantation.[5] Avermectin was a macrocylic lactone isolated from *Streptomyces avermitilis* in 1974; its derivative Ivermectin is an effective treatment for river blindness and other parasitic infections. [6] Doxorubicin isolated from *Streptomyces peucetius*, is commonly used in the treatment of cancer. These drugs are still considered essential medicines by the World Health Organization, exemplifying the clinical importance of microbial natural products today.[7]

The field of natural products research has existed for over 90 years.[3] Rapid integration of advancements in biotechnology and algorithmics has led to a consistent rise in isolation rates of novel scaffolds with diverse chemistries (Figure 1.1). Initially, the protocol for secondary metabolite discovery was rooted in metabolomics and bioactivity screening. The process typically consisted of: (1) Growing a microbe, (2) Fractionating its extracts (3) Assaying for activity, and (4) Isolating the active compound.[8] From the

1950s to the 1970s bioactivity screening generated many antibiotics used today in a period referred to as “the golden age of antibiotic discovery”.[9]

Towards the mid-1980s to early 1990s, there were various innovations in analytical chemistry techniques making novel isolation easier; these included: electrospray ionization (ESI), matrix-assisted laser desorption/ionization (MALDI), the application of Two- Dimensional Nuclear Magnetic Resonance Spectrometry (2D NMR) to biological molecules, and the wide adoption of liquid chromatography paired to mass spectrometry (LCMS).[10-17] The integration of the high-performance instrumentation resulted in a large jump in the yearly rate of novel molecule isolation (from 1991’s 188 novel scaffolds per year to 1995’s 644 novel scaffolds per year).

As molecular biology and DNA sequencing technologies evolved, a deeper understanding of the genetic basis for the biosynthetic process was acquired. In 1991, a landmark paper was published describing erythromycin biosynthesis.[18] All genes responsible for the biosynthesis of the polyketide chain were found within a single operon. Each gene was composed of repeating modules. Each module participated in a different step of the elongation process similar to an assembly line. The genomic organisation of the operon shared collinearity with the required biochemical ordering. This allowed for the prediction of the linear molecular scaffold based solely on the module organisation. Further exploration of other secondary metabolites showed this

genomic organisation was a common pattern in microbial genomes.[19-24] The term biosynthetic gene cluster was used to describe these operons. Beyond synthesis, clusters also contained genes related to metabolite resistance, regulation and decoration (tailoring enzymes). [20, 25-37]

Biosynthetic gene clusters encode many different secondary metabolite families; these include but are not limited to non-ribosomal peptides (NRP), ribosomally encoded post-translationally modified peptides (RiPP), terpenes, polyketides, nucleosides and aminoglycosides.[38] Many of these families are synthesised through separate enzymes including Type II and Type III Polyketide synthesis, and a minority of NRPs.[39-42] Type I polyketides and the majority of non-ribosomal peptides are synthesised by the multi-module megasynthase structure found in the erythromycin gene cluster (via PKS and NRPS respectively).[43, 44] Because megasynthases have a modular order that directly reflects the step-wise synthesis of the metabolite, it is possible to predict the linear chain with high accuracy.[45-48] The PKS and NRPS megasynthases share many similarities, sometimes resulting in a hybridisation of the two families in the final molecular product. [49, 50]

In an NRPS megasynthase, amino acid chain elongation is performed through a cycle of reactions facilitated by separate functional domains.[51] The adenylation (A) domain is used to select and activate a substrate amino acid. These amino acids can

include non-proteogenic amino acids resulting in highly variable peptide chains. The binding pocket sequence of the A domain conveys the substrate specificity. In 1999, the code for the substrate specificity was partially deciphered but many substrates still lack a reference sequence.[52] After activation, the amino acid is tethered to the assembly line for decoration or further elongation using the thiolation (T) domain or peptidyl/acyl carrier protein (PCP or ACP). Examples of substrate decoration include O- or N-methylation. The tethered amino acid is then linked to an upstream monomer using the condensation (C) domain. When elongation is completed, a thioesterase (TE) domain is used for the esterification of the chain. The peptide chain is typically freed using hydration or another nucleophile with possible macrocyclisation. The domains facilitating each of the biochemical steps are highly conserved and easily recognised through sequence homology.[46, 48]

The Type I PKS performs chain elongation using a similar cycle of reactions also facilitated by conserved domains.[53] In PKSs, the adenylation domains are replaced with acyltransferase (AT) domains. The substrate specificity is also dictated by binding pocket sequences but instead of amino acids, the domain selects for the CoA-bound organic acids (e.g. malonyl-CoA). Ketosynthase (KS) domains replace condensation domains, facilitating the C-C bond to tether an extender unit to the polyketide chain. The ketone chain is highly reactive and can be further reduced; first by a ketoreductase (KR) domain - resulting in a β-hydroxyacyl, again by a dehydrotase (DH) domain - resulting in α,β-

enoyl, and possibly further using an enolreductase (ER) domain - resulting in a saturated acyl. The system is simple but yields great diversity, with one group estimating the linear polyketide space as large as 800 million.[54]

The biosynthesis of RiPPs also shares conserved protein domains. RiPPs start as a precursor peptide comprised of a conserved leader motif and core peptide.[55, 56] Following translation, surrounding enzymes will recognise the leader motif and begin modifying the core peptide. Many different modifications can be made resulting in drastic structural changes. Examples of modifying enzymes include dehydratases and methyltransferases; macrocyclization and the formation of disulfides, thioethers, sulfoxides from cysteine residues are also very common. The complex structures of RiPPs result in a wide variety of activities including antibacterial, antifungal and anticancer.[57, 58]

Biosynthetic gene clusters are a demonstration of nature's tendency to reuse enzymes in various combinations resulting in wildly different molecular scaffolds. The simplicity of the mechanism enabled the prediction of encoded chemistries. Predicted natural products from genomic information removed the serendipity of chemical novelty typically associated with bioactivity screening alone.[59, 60] Instead of focusing on microbes isolated from under-explored niches with the hopes of finding novel chemistry, candidates could be selected based on their encoded biosynthetic potential.[61-64] Using

the model of the biosynthetic gene cluster, the workflow for natural product discovery underwent a large change. As bioinformatic techniques increased in accuracy and genomic sequencing decreased in cost, a field of pre-emptive genomic mining emerged.

1.2 Bioinformatics and the Genomic Mining Era

Genomic mining is a computational technique where biosynthetic gene clusters are detected within a genome. The field of genomic mining is closely tied to bioinformatics. A pattern that history has shown time and time again is as bioinformatic tools increase in accuracy and scalability so do the tools in genomic mining.

The birth of bioinformatics begins with the software COMPROTEIN in 1962.[65, 66] It generated a peptide's primary structure using Edman peptide sequencing data. This was one of the first examples of computational tooling creating a data explosion and facilitating the scientific inquiry of completely new research questions. As more protein sequences were discovered, there grew a need to explore the evolutionary relationships between sequences. There were many different attempts to standardise the comparison, but it was only in 1970 when Saul B. Needleman and Christian D. Wunsch solved this problem by creating a dynamic programming algorithm to align pairs of sequences.[67] While alignment algorithms remain important even today, they have a scalability issue requiring $O(n^2)$ comparisons between each pair of sequences. To address scaling issues, rapid sequence algorithms began to emerge in the mid-1980s and early 1990s. In 1985

FASTA was released, dramatically decreasing the amount of time required to compare a protein sequence to a database.[68, 69] In 1990, the BLAST algorithm was released; it remains a staple in sequence comparison even today. [70]

The era of genomics begins in 1995 when the first complete bacterial genome was sequenced *(Haemophilus influenzae)*.[71] As sequencing projects grew, another data explosion occurs with the annotation of the incoming peptide sequences becoming unmanageable. Protein libraries grew at a rapid rate, with the SWISS-PROT library containing ~81,000 protein sequences by 1999.[72] In 1997, a library called Pfam was released to rapidly annotate and classify incoming peptide sequences into emerging families.[73] Pfam used HMMER's profile Hidden Markov Models (pHMMs) to determine whether or not a sequence belonged to a protein family.[74] Each protein family was represented as a Markov chain. The sequence homology of a family was modelled as a transition matrix; the likelihood of each residue belonging to a position in the chain was determined using a seed sequence alignment.[75] The probability of an incoming sequence belonging to the modelled family was determined using the transition matrix. By modelling each family separately, pHMMs can personalise scoring for substitutions, insertions and deletions in a way BLAST cannot. Each pHMM could report probabilities, along with the start and end coordinates for matching regions. Pfam showed pHMMs were extremely useful for the scalable determination of protein motifs and

domains *en masse*. Libraries such as PANTHER, TIGRFAM, and SMART using pHMMs emerged soon after.[76-78]

In 2005, the first next-generation sequencing (NGS) platform was released called 454 sequencing; it was much faster and cheaper than conventional Sanger sequencing for interrogating microbial genomes.[79] By 2006, the Genomes OnLine Database (GOLD) reported ~250 completed microbial genomes, with ~700 other projects in progress.[80, 81] With an unprecedented amount of microbial genomic information available, natural product scientists release the first genomic mining pipelines. The proprietary DECIPHER was created in 2002 by Ecopia Biosciences.[82] They utilised public datasets such as GenBank, to create an internal library of gene clusters. Using BLAST, they were able to rapidly query microbial genomes for biosynthetic genes and identify encoded biosynthetic potential. In 2006, de Jong et al. released BAGEL, a web server for determining putative bacteriocins.[83] It used Pfam HMMs to determine motifs of interest from predicted open-reading frames. Using tools like these, candidates for fermentation could be selected with a greater likelihood of success.

Pfam and pHMMs continue to play a large part in the genomic mining tools of today. The two dominant genomic mining tools, PRISM and antiSMASH, both rely on pHMMs selected specifically from the Pfam library as well as in-house pHMMs specifically created for biosynthetic gene clusters.[46, 48] Genomic mining tools have

evolved from identifying bacteriocins to identifying many different types of BGCs including aminoglycosides, nucleosides, β-lactams, alkaloids, and lincosamides. Bioinformatic tools such as BLAST were also integrated into the pipelines, with PRISM still using BLAST to identify key residue differences in adenylation domain binding pockets. Beyond predicting linear scaffolds, PRISM has modelled out tailoring reactions allowing the accurate prediction of complex scaffold structures. The influence of smarter candidate selection using genomic mining was immediately evident. With genomic mining tools in hand, natural product research experienced a second renaissance with rates of novel molecule discovery reaching as high as 1,389 in 2015. With the enzymology of more and more gene clusters being modelled into the genomic mining pipelines, the scalability of the current workflow has come into question. PRISM 4 boasts a library of 1772 pHMMs while antiSMASH 6.0 has 354 pHMMs; in both pipelines, all pHMMs must be run against each protein sequence individually.[84, 85] As more BGCs are discovered, these libraries will continue to grow as will their running times.

Just as advances in biotechnology and bioinformatics created data explosions with DNA and proteins, genomic mining created a data explosion with gene clusters. 30 years after the discovery of the first biosynthetic gene cluster, the Integrated Microbial Genomes' Atlas of Biosynthetic Gene Clusters (IMG-ABC) now reports 411,475.[86] The

same questions that arose when protein datasets grew exponentially are now being asked of gene clusters.

As Pfam was invented to quickly annotate incoming sequences with known protein families, there now exists tools to annotate incoming BGCs with known secondary metabolites. MultiGeneBLAST used the BLAST algorithm to map open reading frames from putative BGCs to gene clusters with solved metabolites.[87] There were caveats to this approach including the inability to recognise enzymes that are dissimilar in sequence homology but identical in function. To mitigate false negatives, more advanced pipelines moved away from the amino acid sequence for comparison; they instead characterised a protein as a sequence of predicted biosynthetic domains. Internal tools such as GARLIC (Global Alignment for natuRaL-products chemInformatiCs) and MLST used the Needleman-Wunch global alignment and BLAST algorithms respectively to perform comparisons of gene clusters represented as sequences of functional domains.[88] All three of these algorithms can score incoming gene clusters with a percentage of known identity. Unfortunately, as libraries of known BGCs continue to grow, alignment-based methods are not scalable.

Other frameworks have emerged to programmatically deduce gene cluster families (GCFs). BiG-SCAPE annotates all open-reading frames within BGCs with Pfam domains and creates similarity networks using a combination of pairwise metrics including the

Jaccard Index and affinity propagation.[89] While the strategy works proficiently on small datasets, it has a quadratic runtime complexity that does not allow application beyond a few tens of thousands of BGCs; this is not feasible when querying against today's wealth of BGCs (a combined total of over one million). To improve on this, BiG-SLICE was released.[90] It uses a Pfam feature matrix to represent BGCs as vectors. The use of vectors allows for the use of near-linear clustering algorithms such as K-Means and BIRCH to build GCFs.[91, 92] There are limitations to relying on Pfam for featurisation; while polyketides and a minority of NRPs have many features, the majority of NRPs, terpene and RiPPs have very few. In addition, Pfam does not take into account the individual residue differences in RiPP propeptide sequences and adenylation domain sequences, both of which are important determinants of the final natural product's chemical structure.

1.3 Natural Language Processing

Many of the problems plaguing the field of genomic mining are also encountered in natural language processing (NLP). In proteins, conserved functional domains are determined in a continuous protein sequence using pHMMs. In language processing, a similar problem exists called part-of-speech tagging.

A part of speech (PoS) is a category of words which share similar grammatical properties; in the English language, examples of a PoS include noun, verb, adjective etc.

In the 1970s a massive dataset of documents called the Lancaster-Oslo-Bergen Corpus was prepared in a machine-readable format. The corpus was made of 500 two-thousand-word texts written in British English. [93] One of the tasks required to process the massive dataset included PoS tagging. Just as SWISS-PROT was struggling to keep up with annotations of incoming peptide sequences, computational linguists were struggling to annotate a corpus of this size. To solve this, they modelled each PoS with HMMs, similarly to how Pfam modelled each protein family.[94] Since then, the field of NLP has continued to innovate and make massive strides. While Pfam still uses pHMMs to annotate proteins today, in NLP PoS tagging tasks are accomplished by a variety of other techniques highly scalable techniques. One of the most popular methods involves combining an artificial neural network (ANN) with a conditional random field (CRF). [95-98]

Artificial neural networks are mathematical models inspired by biological neural networks of animal brains.[99] Typically ANNs are structured with a minimum of three layers made up of individual artificial neurons: (1) An input layer which processes the raw input data such as tokenised words. (2) A hidden layer which can be made up of many layers in the case of Deep Neural Networks (DNNs); these layers are where the majority of input processing occurs. (3) An output layer, which will compute the final values such as vector representations of the input (embeddings). The output for an artificial neuron is typically calculated through a combination of weighted biases and

activation functions being applied to the neuron's input. While the input data for a typical ANN is assumed to be independent of each other, a special type of ANN called a recurrent neural network (RNN) has neurons connected in cycles, allowing the input of sequence data.[100] A specialised RNN called the Long-Short Term Memory (LSTM-RNN) has found much success in NLP.[101] Each unit is comprised of a cell, an input gate, an output gate and a forget gate. The cell acts as the unit's memory while the model moves across long sequences. The forget gate is used to prune unneeded information from the model's state. It has been used in handwriting recognition, speech recognition and even playing video games.[102-104]

The weights within an ANN are tuned using a training task. Depending on the training task, different weights will be optimised. For example, if an ANN is trained on classifying sentences as negative sentiment versus positive sentiment, it will have a different set of weights than the same ANN except trained on classifying sentences as spam versus non-spam. While weights may differ between tasks, some residual understanding may be transferable. In 2018, FastAI released a training method called Universal Language Model Fine-tuning (ULMFit).[105] It involved first training a language model with a simple pertaining task and then fine-tuning the model with other initiatives later. The concept of pre-training with one task to help an ANN perform well at another is called transfer learning. In subsequent frameworks, large language models (LLMs) utilised transfer learning through a ULMFit-like training regimen.

In 2017, Google released a language model framework called Embeddings from Language Models (ELMo). [106] It used an architecture comprised of a stacked layer of bidirectional Long Short-Term Memory Recurrent Neural Networks (LSTM-RNNs). It was first pre-trained on 10 epochs on the 1B Word Benchmark; afterwards, it was fine-tuned for individual language modelling tasks including question answering, sentiment analysis and named entity recognition (NER). NER is a task very similar to PoS; instead of calling parts of speech within a sentence, entities are recognised and classified into categories. Categorisation of the words is highly dependent on context; for example, “apple” could be classified as a company or a fruit depending on the context). To perform NER with ELMo, embeddings from the LSTM-RNN were passed to a CRF layer. The architecture set a new state-of-the-art score of 92.22% accuracy.

Following the release of this paper, bioinformatic tools started to migrate to LSTM-RNNs. In 2019, SignalP released version 5.0, in which they moved to a combination of convolution neural networks (CNNs), LSTMs and conditional random fields (CRFs); their performance reached an all-time high.[107] Also in 2019, Merck published a BGC caller using the ELMo architecture with random forests (RFs) on genomes. [108]They represented BGCs as ordered sequences of Pfam annotations; the model achieved greater performance at phenotyping BGCs than AntiSMASH rule-based approach. In 2020, Magarvey Laboratories released DeepRiPP, an LSTM-RNN CRF

architecture, trained with ULMFiT to recognise RiPPs. DeepRiPP outperformed previous alignment-based approaches.[58]

In 2019, Google released a model called Bidirectional Encoder Representations from Transformers or BERT.[109] Just as ELMo had set new state-of-the-art records across NLP tasks, BERT had pushed performance even further. BERT was based on the transformer architecture released in 2017.[110] The transformer can be broken down into two separate components: an encoder and a decoder. The encoder is used for converting the input tokens into an embedded space. The decoder is used to convert the embeddings into the target language. The transformer can be used to translate a sentence from one language to another; it could also be used to complete text-to-text pretraining tasks such as predicting the masked parts of a sentence as demonstrated with Google's T5 model. [111]

The encoder module of a transformer can be broken down into a stack of encoder units. [110] The first unit is provided with a tokenised input of the text, while subsequent encoder units are fed the output of the preceding encoder unit. Upon entering a unit, the input vector is passed through a series of transformation layers. The first layer is a self-attention layer - this layer is made up of multiple attention heads. Each attention head will calculate an embedded matrix for each input token. The matrix is calculated using all tokens found within the input; this imbues the embedding with context. For example, in

the sentence “The quick brown fox jumped over the lazy dog”, the embedding for “fox” would be calculated as a summed weighted vector using the words: the, quick, brown, fox, jumped, over, the, lazy, dog. It is through the self-attention heads, that the transformer can learn relationships between tokens. Because there are multiple self-attention heads, each head can learn separate functions. Some attention heads have been shown to specialise in certain types of syntactic relations. Beyond the self-attention layer, there is a normalisation layer and a feed-forward layer, both of which also contain weights.

The decoder module is also made of stacked units, with each output being passed to the proceeding unit. For decoding to take place, the embedded output from the final encoder unit is split into a series of vectors and passed to each of the decoder units. The self-attention heads in decoder units function differently than encoders with each token vector being calculated from attention limited to only preceding tokens. There is also an additional “encoder-decoder attention” layer which functions essentially like the self-attention heads from the encoder side with the exception of some minor changes in how the query, key and value matrices are derived. The final embedding from the decoder module is passed through a linear layer to calculate logits and then a softmax layer to calculate probabilities. Each probability is linked to a word in the transformer’s vocabulary. The highest scoring probability is what the token’s embedding will be translated to.

The BERT language model used only the encoding side of the transformer architecture. It was trained similarly to ULMFit, with pre-training followed by fine-tuning tasks. The developers pre-trained with a combination of two tasks: (1) A masked language modelling task - this involved masking tokens randomly and using the output embeddings' probabilities to recover the token and (2) a next sentence prediction task - this involved embedding two sentences and using a classifier to determine whether or not the sentences belonged next to each other in the original document. Both tasks have training data where ground truth labels can be easily manufactured from a large corpus of documents; as such they are considered unsupervised learning tasks. The unsupervised training was used to tune the 345 million parameters in BERT. The fine-tuning tasks were supervised and had substantially smaller datasets. Fine-tuning tasks included sentence pair classification, single sentence classification, question answering, and NER. Across the board, BERT was able to demonstrate high performance.[109]

The high performance of BERT along with the ease of training using large corpora and self-supervised tasks made it a favourable model to train with biological data. In 2020, ProtBert was released showing BERT could be trained on protein sequences and generate embeddings reflective of three-dimension structures.[112] In 2021, DNABert was released showing BERT could be trained on DNA sequences and could be used to find regulation domains in the human genome.[113] In 2022, SignalP migrated from the

LSTM architecture and moved to a BERT model, using a fine-tuned version of ProtBERT with a CRF for calling the boundaries of signal peptides within a propeptide.[114] As of yet, no genomic mining tools have adopted a transformer architecture into their pipeline.

1.4. Scope and nature of this work

Natural products research, and more broadly the field of genomic mining have entered unknown territory. Bacterial genomes are being sequenced cheaper and faster than ever before. Genomic mining tools have produced a wealth of information so large, we do not have tooling capable of making use of it. While other areas are adopting the technological innovations produced by tech giants like Google and Facebook to deal with the big data crisis, genomic mining suites continue to use outdated methods like HMMER and BLAST. The implementation of industry-standard techniques to solve big data problems will alleviate many of the computational burdens plaguing the field. NLP and transformers have moved deep learning from highly structured supervised learning to self-supervised and unsupervised learning. The copious amounts of publicly available genomic, peptide and molecular data, lend themselves to these tasks.

In my thesis, I constructed a series of tools to blueprint how deep learning systems and statistics can be used to leverage big public datasets and translate them into new inferences. I hypothesised that the output of the natural product isolation pipeline can be

greatly enhanced through the integration of machine learning and statistics. I aimed on improving the natural product discovery pipeline in three ways. First, improving the resolution of molecular comparison with natural language processing techniques. Second, improving the scalability and quality of genomic mining by using deep learning models. Third, improving activity prediction of encoded metabolites through the integration of multi-omics data.

1.4.1 Improving Molecular Comparison with Deep Learning

Conventional cheminformatics tooling compares molecules using substructure-based fingerprints. While the field has proposed deep learning approaches to improve molecular comparison, they continue to build on this paradigm and focus learning efforts on fingerprint-based substructures. In Chapter 2, I demonstrate a natural language processing approach to molecule comparison. Instead of using substructures, I treat molecules as sentences comprised of atom words. I showcase the failures of conventional approaches and how a large language model's resolution can mitigate them.

1.4.2 Improving Genomic Mining with Deep Learning

Conventional genomic mining tools are bottlenecked by the speed of pHMMs and BLAST. In addition, the structure of pHMM-based pipelines requires a laborious and meticulous approach to modelling new enzymology. In Chapter 3, I describe a novel

approach to genomic mining. The strategy is rooted in state-of-the-art natural language processing technology and provides scalability in terms of computation as well as curation. Using a suite of large language models, I replace all pHMM and BLAST-based tasks, thereby greatly increasing the speed of genomic mining. To speed up curation, I introduce a simple, vector-based method for adding new enzymology to the pipeline. I also demonstrate how a vector-based approach facilitates the rapid discovery of gene cluster families and gene cluster comparison.

Genomic mining pipelines have limited efficacy on highly fragmented genomes. RefSeq is plagued by unfinished assemblies comprised of short-read Illumina data. In Chapter 4, I propose a new strategy for improving the contig size of poor-quality genomes. I integrate multi-omics inference and graph deep learning into conventional assembly workflows. Using a graph convolution network trained on biosynthetic gene clusters and a transformer trained on bacterial genomes, I create more complete contigs and recover fragmented gene clusters.

1.4.3 Improving Activity Prediction with Integrated Data and Statistics

Encoded metabolite activity prediction has conventionally taken two approaches: (1) Focusing on the predicted molecular structure to utilise QSAR-based strategies or (2)

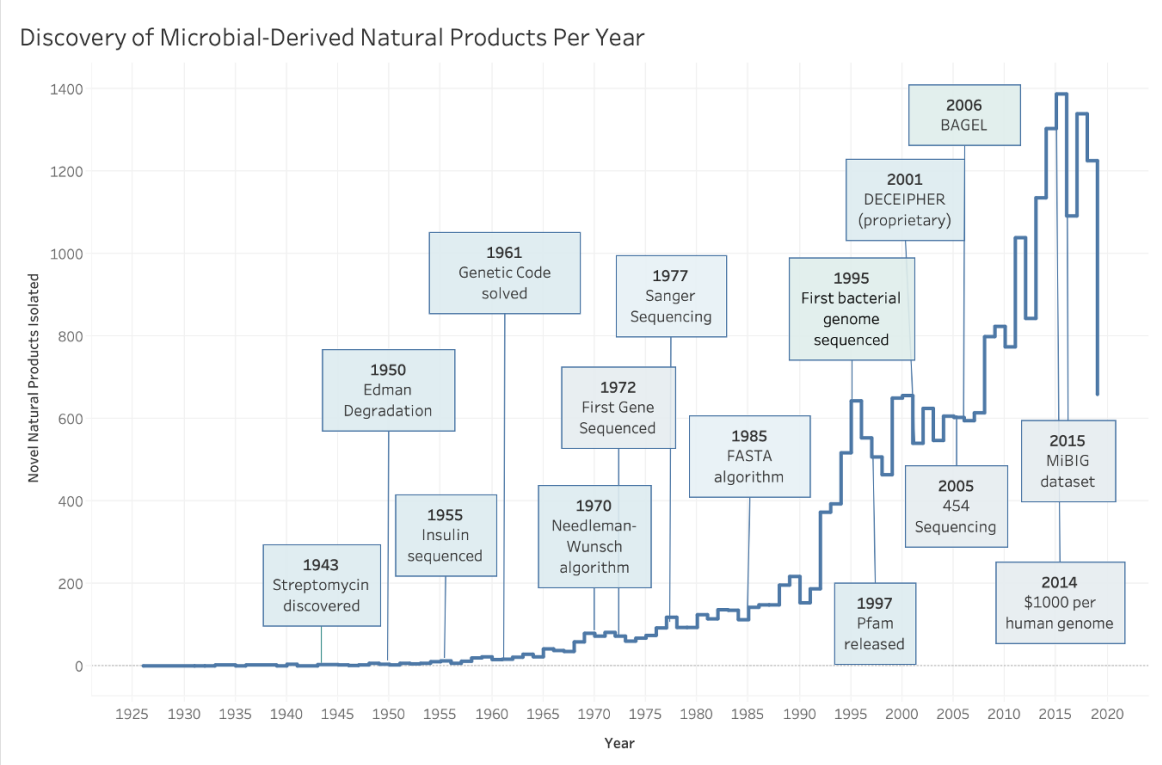
Focusing on conserved enzymology to project activity labels based on biosynthetic relatedness. In Chapter 5, I propose a third approach; by treating the human microbiome as an in-situ system for understanding microbial metabolism and host effect, the activity of an encoded metabolite can be elucidated. Using a simple statistics pipeline, I demonstrate that heuristic measures for a microbial metabolite can be used to generate a gene signature of its effect on the host. I show how the approach can be performed using three separate strategies rooted in microbial transcriptomics, microbial metabolomics and microbial proteomics.

1.4.4. Thesis Overview

With the wealth of publicly available data and the low barrier to entry for high-performance deep learning systems, artificial intelligence systems can reinvigorate natural product research once more. Using my unique combination of skills, I created scalable platforms capable of improving many of the pertinent steps used in data-driven natural product drug discovery.

1.5 Figures and Tables

Figure 1.1: The database of molecules from NPAtlas, binned by isolation publication year. The plot was created using Tableau Desktop. The amount of molecules isolated over the decades has continued to trend upward.



1.6 Bibliography

1. Raymond, K.N., E.A. Dertz, and S.S. Kim, *Enterobactin: An archetype for microbial iron transport.* Proceedings of the National Academy of Sciences, 2003. **100**(7): p. 3584-3588.
2. Coquant, G., J.-P. Grill, and P. Seksik, *Impact of N-Acyl-Homoserine Lactones, Quorum Sensing Molecules, on Gut Immunity.* Frontiers in immunology., 2020. **11**.
3. Cragg, G.M., D.J. Newman, and K.M. Snader, *Natural Products in Drug Discovery and Development.* Journal of Natural Products, 1997. **60**(1): p. 52-60.
4. Schatz, A., E. Bugle, and S.A. Waksman, *Streptomycin, a Substance Exhibiting Antibiotic Activity Against Gram-Positive and Gram-Negative Bacteria.∗†.* Proceedings of the Society for Experimental Biology and Medicine, 1944. **55**(1): p. 66-69.
5. Colombo, D. and E. Ammirati, *Cyclosporine in transplantation - a history of converging timelines.* Journal of Biological Regulators and Homeostatic Agents, 2011. **25**(4): p. 493-504.
6. Campbell, W.C., *History of avermectin and ivermectin, with notes on the history of other macrocyclic lactone antiparasitic agents.* Current Pharmaceutical Biotechnology, 2012. **13**(6): p. 853-865.
7. *WHO model list of essential medicines - 22nd list, 2021.*
8. Atanasov, A.G., et al., *Natural products in drug discovery: advances and opportunities.* Nature Reviews Drug Discovery, 2021. **20**(3): p. 200-216.
9. Aminov, R.I., *A Brief History of the Antibiotic Era: Lessons Learned and Challenges for the Future.* Frontiers in Microbiology, 2010. **1**: p. 134.
10. *Characterization of Alkaloids By Electrospray Mass Spectrometry: Natural Product Letters: Vol 1, No 2.*
11. *Evolution of Mass Spectrometers.* Lab Manager.
12. Karas, M., D. Bachmann, and F. Hillenkamp, *Influence of the wavelength in high-irradiance ultraviolet laser desorption mass spectrometry of organic molecules.* Analytical Chemistry, 1985. **57**(14): p. 2935-2939.
13. Fallon, A., R.F.G. Booth, and L.D. Bell, Applications of HPLC in Biochemistry. 1987: Elsevier. 355.
14. Albert, K., *Liquid chromatography-nuclear magnetic resonance spectroscopy.* Journal of Chromatography. A, 1999. **856**(1-2): p. 199-211.
15. Mesilaakso, M. and A. Niederhauser, *Nuclear Magnetic Resonance Spectroscopy in Analysis of Chemicals Related to the Chemical Weapons Convention*, in *Encyclopedia of Analytical Chemistry*. 2006, John Wiley & Sons, Ltd.
16. Marion, D., *An Introduction to Biological NMR Spectroscopy.* Molecular & Cellular Proteomics : MCP, 2013. **12**(11): p. 3006-3025.

17. de Koster, C.G. and P.J. Schoenmakers. *History of liquid chromatography—mass spectrometry couplings*. 2020. Elsevier.
18. Donadio, S., et al., *Modular Organization of Genes Required for Complex Polyketide Biosynthesis.* Science, 1991. **252**(5006): p. 675-679.
19. Aparicio, J.F., et al., *Organization of the biosynthetic gene cluster for rapamycin in Streptomyces hygroscopicus: analysis of the enzymatic domains in the modular polyketide synthase.* Gene, 1996. **169**(1): p. 9-16.
20. Paradkar, A.S., et al., *Molecular analysis of a beta-lactam resistance gene encoded within the cephamycin gene cluster of Streptomyces clavuligerus.* Journal of Bacteriology, 1996. **178**(21): p. 6266-6274.
21. Ruan, X., et al., *A second type-I PKS gene cluster isolated from Streptomyces hygroscopicus ATCC 29253, a rapamycin-producing strain.* Gene, 1997. **203**(1): p. 1-9.
22. Ichinose, K., et al., *The granaticin biosynthetic gene cluster of Streptomyces violaceoruber Tü22: sequence analysis and expression in a heterologous host.* Chemistry & Biology, 1998. **5**(11): p. 647-659.
23. Motamedi, H. and A. Shafiee, *The biosynthetic gene cluster for the macrolactone ring of the immunosuppressant FK506.* European Journal of Biochemistry, 1998. **256**(3): p. 528-534.
24. Pelludat, C., et al., *The yersiniabactin biosynthetic gene cluster of Yersinia enterocolitica: organization and siderophore-dependent regulation.* Journal of Bacteriology, 1998. **180**(3): p. 538-546.
25. Raibaud, A., et al., *Nucleotide sequence analysis reveals linked N-acetyl hydrolase, thioesterase, transport, and regulatory genes encoded by the bialaphos biosynthetic gene cluster of Streptomyces hygroscopicus.* Journal of Bacteriology, 1991. **173**(14): p. 4454-4463.
26. McGowan, S.J., et al., *Analysis of the carbapenem gene cluster of Erwinia carotovora: definition of the antibiotic biosynthetic genes and evidence for a novel beta-lactam resistance mechanism.* Molecular Microbiology, 1997. **26**(3): p. 545-556.
27. Onaka, H., et al., *Characterization of the biosynthetic gene cluster of rebeccamycin from Lechevalieria aerocolonigenes ATCC 39243.* Bioscience, Biotechnology, and Biochemistry, 2003. **67**(1): p. 127-138.
28. van Pée, K.H. and S. Unversucht, *Biological dehalogenation and halogenation reactions.* Chemosphere, 2003. **52**(2): p. 299-312.
29. Chang, Z., et al., *Biosynthetic pathway and gene cluster analysis of curacin A, an antitubulin natural product from the tropical marine cyanobacterium Lyngbya majuscula.* Journal of Natural Products, 2004. **67**(8): p. 1356-1367.
30. Sekurova, O.N., et al., *In vivo analysis of the regulatory genes in the nystatin biosynthetic gene cluster of Streptomyces noursei ATCC 11455 reveals their differential control over antibiotic biosynthesis.* Journal of Bacteriology, 2004. **186**(5): p. 1345-1354.

31. Felnagle, E.A., et al., *Identification of the biosynthetic gene cluster and an additional gene for resistance to the antituberculosis drug capreomycin.* Applied and Environmental Microbiology, 2007. **73**(13): p. 4162-4170.
32. Knirschová, R., et al., *Multiple regulatory genes in the salinomycin biosynthetic gene cluster of Streptomyces albus CCM 4719.* Folia Microbiologica, 2007. **52**(4): p. 359-365.
33. Luo, Y., et al., *Validation of the intact zwittermicin A biosynthetic gene cluster and discovery of a complementary resistance mechanism in Bacillus thuringiensis.* Antimicrobial Agents and Chemotherapy, 2011. **55**(9): p. 4161-4169.
34. Xie, Y., et al., *Identification of the biosynthetic gene cluster and regulatory cascade for the synergistic antibacterial antibiotics griseoviridin and viridogrisein in Streptomyces griseoviridis.* Chembiochem: A European Journal of Chemical Biology, 2012. **13**(18): p. 2745-2757.
35. Cai, W., et al., *The Biosynthesis of Capuramycin-type Antibiotics: IDENTIFICATION OF THE A-102395 BIOSYNTHETIC GENE CLUSTER, MECHANISM OF SELF-RESISTANCE, AND FORMATION OF URIDINE-5′-CARBOXAMIDE.* The Journal of Biological Chemistry, 2015. **290**(22): p. 13710-13724.
36. Thomy, D., et al., *The ADEP Biosynthetic Gene Cluster in Streptomyces hawaiiensis NRRL 15010 Reveals an Accessory clpP Gene as a Novel Antibiotic Resistance Factor.* Applied and Environmental Microbiology, 2019. **85**(20): p. e01292-19.
37. Kato, S., et al., *Induction of secondary metabolite production by hygromycin B and identification of the 1233A biosynthetic gene cluster with a self-resistance gene.* The Journal of Antibiotics, 2020. **73**(7): p. 475-479.
38. Walsh, C.T. and M.A. Fischbach, *Natural Products Version 2.0: Connecting Genes to Molecules.* Journal of the American Chemical Society, 2010. **132**(8): p. 2469-2493.
39. Shen, B., *Polyketide biosynthesis beyond the type I, II and III polyketide synthase paradigms.* Current Opinion in Chemical Biology, 2003. **7**(2): p. 285-295.
40. Katsuyama, Y. and Y. Ohnishi, *Type III polyketide synthases in microorganisms.* Methods in Enzymology, 2012. **515**: p. 359-377.
41. Wang, H., et al., *Atlas of nonribosomal peptide and polyketide biosynthetic pathways reveals common occurrence of nonmodular enzymes.* Proceedings of the National Academy of Sciences of the United States of America, 2014. **111**(25): p. 9259-9264.
42. Wang, J., et al., *Biosynthesis of aromatic polyketides in microorganisms using type II polyketide synthases.* Microbial Cell Factories, 2020. **19**(1): p. 110.
43. Ansari, M.Z., et al., *NRPS-PKS: a knowledge-based resource for analysis of NRPS/PKS megasynthases.* Nucleic Acids Research, 2004. **32**(Web Server issue): p. W405-413.

44. Fischer, M. and M. Grininger, *Strategies in megasynthase engineering – fatty acid synthases (FAS) as model proteins.* Beilstein Journal of Organic Chemistry, 2017. **13**: p. 1204-1211.
45. Yadav, G., R.S. Gokhale, and D. Mohanty, *Towards Prediction of Metabolic Products of Polyketide Synthases: An In Silico Analysis.* PLOS Computational Biology, 2009. **5**(4): p. e1000351.
46. Medema, M.H., et al., *antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences.* Nucleic Acids Research, 2011. **39**(Web Server issue): p. W339-W346.
47. Röttig, M., et al., *NRPSpredictor2—a web server for predicting NRPS adenylation domain specificity.* Nucleic Acids Research, 2011. **39**(Web Server issue): p. W362-W367.
48. Skinnider, M.A., et al., *Genomes to natural products PRediction Informatics for Secondary Metabolomes (PRISM).* Nucleic Acids Research, 2015. **43**(20): p. 9645-9662.
49. Silakowski, B., B. Kunze, and R. Müller, *Multiple hybrid polyketide synthase/non-ribosomal peptide synthetase gene clusters in the myxobacterium Stigmatella aurantiaca.* Gene, 2001. **275**(2): p. 233-240.
50. Mizuno, C.M., et al., *A Hybrid NRPS-PKS Gene Cluster Related to the Bleomycin Family of Antitumor Antibiotics in Alteromonas macleodii Strains.* PLOS ONE, 2013. **8**(9): p. e76021.
51. Martínez-Núñez, M.A. and V.E.L.y. López, *Nonribosomal peptides synthetases and their applications in industry.* Sustainable Chemical Processes, 2016. **4**(1): p. 13.
52. Stachelhaus, T., H.D. Mootz, and M.A. Marahiel, *The specificity-conferring code of adenylation domains in nonribosomal peptide synthetases.* Chemistry & Biology, 1999. **6**(8): p. 493-505.
53. Nivina, A., et al., *Evolution and Diversity of Assembly-Line Polyketide Synthases.* Chemical Reviews, 2019. **119**(24): p. 12524-12547.
54. González-Lergier, J., L.J. Broadbelt, and V. Hatzimanikatis, *Theoretical Considerations and Computational Analysis of the Complexity in Polyketide Synthesis Pathways.* Journal of the American Chemical Society, 2005. **127**(27): p. 9930-9938.
55. Arnison, P.G., et al., *Ribosomally synthesized and post-translationally modified peptide natural products: overview and recommendations for a universal nomenclature.* Natural product reports, 2013. **30**(1): p. 108-160.
56. Rubin, G.M. and Y. Ding, *Recent advances in the biosynthesis of RiPPs from multicore-containing precursor peptides.* Journal of industrial microbiology & biotechnology, 2020. **47**(9-10): p. 659-674.
*57. Kodani, S. and K. Unno, How to harness biosynthetic gene clusters of lasso peptides. Journal of Industrial Microbiology and Biotechnology, 2020. **47**(9-10): p. 703-714.*

58. Merwin, N.J., et al., *DeepRiPP integrates multiomics data to automate discovery of novel ribosomally synthesized natural products.* Proceedings of the National Academy of Sciences, 2020. **117**(1): p. 371-380.
59. Bauman, K.D., et al., *Genome mining methods to discover bioactive natural products.* Natural Product Reports. **38**(11): p. 2100-2129.
60. Rosic, N., *Genome Mining as an Alternative Way for Screening the Marine Organisms for Their Potential to Produce UV-Absorbing Mycosporine-like Amino Acid.* Marine Drugs, 2022. **20**(8): p. 478.
61. Sanchez, L.M., et al., *Examining the Fish Microbiome: Vertebrate-Derived Bacteria as an Environmental Niche for the Discovery of Unique Marine Natural Products.* PLOS ONE, 2012. **7**(5): p. e35398.
62. Scherlach, K. and C. Hertweck, *Mining and unearthing hidden biosynthetic potential.* Nature Communications, 2021. **12**(1): p. 3864.
63. Rojas-Gätjens, D., et al., *Antibiotic-producing Micrococcales govern the microbiome that inhabits the fur of two- and three-toed sloths.* Environmental Microbiology, 2022. **24**(7): p. 3148-3163.
64. Götze, S., et al., *Ecological Niche-Inspired Genome Mining Leads to the Discovery of Crop-Protecting Nonribosomal Lipopeptides Featuring a Transient Amino Acid Building Block.* Journal of the American Chemical Society, 2023. **145**(4): p. 2342-2353.
65. *COMPROTEIN, a Computer Program to Aid Primary Protein Structure Determination.*
*66. Gauthier, J., et al., A brief history of bioinformatics. Briefings in Bioinformatics, 2019. **20**(6): p. 1981-1996.*
67. Needleman, S.B. and C.D. Wunsch, *A general method applicable to the search for similarities in the amino acid sequence of two proteins.* Journal of Molecular Biology, 1970. **48**(3): p. 443-453.
68. Lipman, D.J. and W.R. Pearson, *Rapid and Sensitive Protein Similarity Searches.* Science, 1985. **227**(4693): p. 1435-1441.
69. Pearson, W.R., *Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms.* Genomics, 1991. **11**(3): p. 635-650.
*70. Altschul, S.F., et al., Basic local alignment search tool. Journal of Molecular Biology, 1990. **215**(3): p. 403-410.*
71. Fleischmann, R.D., et al., *Whole-genome random sequencing and assembly of Haemophilus influenzae Rd.* Science (New York, N.Y.), 1995. **269**(5223): p. 496-512.
72. Bairoch, A. and R. Apweiler, *The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.* Nucleic Acids Research, 2000. **28**(1): p. 45-48.

73. Sonnhammer, E.L.L., S.R. Eddy, and R. Durbin, *Pfam: A comprehensive database of protein domain families based on seed alignments.* Proteins: Structure, Function, and Genetics, 1997. **28**(3): p. 405-420.
74. Mistry, J., et al., *Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions.* Nucleic Acids Research, 2013. **41**(12): p. e121.
75. Eddy, S.R., *Multiple Alignment Using Hidden Markov Models.*
76. Schultz, J.r., et al., *SMART, a simple modular architecture research tool: Identification of signaling domains.* Proceedings of the National Academy of Sciences, 1998. **95**(11): p. 5857-5864.
77. Haft, D.H., et al., *TIGRFAMs: a protein family resource for the functional identification of proteins.* Nucleic Acids Research, 2001. **29**(1): p. 41-43.
78. Thomas, P.D., et al., *PANTHER: A Library of Protein Families and Subfamilies Indexed by Function.* Genome Research, 2003. **13**(9): p. 2129-2141.
79. Patrick, K.L., *454 Life Sciences: Illuminating the future of genome sequencing and personalized medicine.* The Yale Journal of Biology and Medicine, 2007. **80**(4): p. 191-194.
80. Kyrpides, N.C., *Genomes OnLine Database (GOLD 1.0): a monitor of complete and ongoing genome projects world-wide.* Bioinformatics, 1999. **15**(9): p. 773-774.
81. Hadjithomas, M., et al., *IMG-ABC: A Knowledge Base To Fuel Discovery of Biosynthetic Gene Clusters and Novel Secondary Metabolites.* mBio, 2015. **6**(4): p. e00932-15.
82. Farnet, C.M., et al., *Method, system and knowledge repository for identifying a secondary metabolite from a microorganism*, O. World Intellectual Property, Editor. 2003, Ecopia Biosciences Inc.
83. de Jong, A., et al., *BAGEL: a web-based bacteriocin genome mining tool.* Nucleic Acids Research, 2006. **34**(Web Server issue): p. W273-W279.
84. Skinnider, M.A., et al., *Comprehensive prediction of secondary metabolite structure and biological activity from microbial genome sequences.* Nature Communications, 2020. **11**(1): p. 6058.
85. Blin, K., et al., *antiSMASH 6.0: improving cluster detection and comparison capabilities.* Nucleic Acids Research, 2021. **49**(W1): p. W29-W35.
86. Palaniappan, K., et al., *IMG-ABC v.5.0: an update to the IMG/Atlas of Biosynthetic Gene Clusters Knowledgebase.* Nucleic Acids Research, 2020. **48**(D1): p. D422-D430.
87. Medema, M.H., E. Takano, and R. Breitling, *Detecting sequence homology at the gene cluster level with MultiGeneBlast.* Molecular Biology and Evolution, 2013. **30**(5): p. 1218-1223.
88. Skinnider, M.A., et al., *Comparative analysis of chemical similarity methods for modular natural products with a hypothetical structure enumeration algorithm.* Journal of Cheminformatics, 2017. **9**: p. 46.

89. Navarro-Muñoz, J.C., et al., *A computational framework to explore large-scale biosynthetic diversity.* Nature chemical biology, 2020. **16**(1): p. 60-68.
90. Kautsar, S.A., et al., *BiG-SLiCE: A highly scalable tool maps the diversity of 1.2 million biosynthetic gene clusters.* GigaScience, 2021. **10**(1): p. giaa154.
91. Zhang, T., R. Ramakrishnan, and M. Livny. *BIRCH: an efficient data clustering method for very large databases*. 1996. Association for Computing Machinery.
92. Johnson, J., M. Douze, and H. Jégou, Billion-scale similarity search with GPUs. 2017, arXiv.
93. Johansson, S. and S. Johansson, *Lancaster-Oslo-Bergen corpus of modern English (LOB) : [tagged, horizontal format] / Stig Johansson.*
94. DeRose, S.J., *Grammatical category disambiguation by statistical optimization.* Computational Linguistics, 1988. **14**(1): p. 31-39.
95. Pandian, S.L. and T.V. Geetha. *CRF Models for Tamil Part of Speech Tagging and Chunking*. 2009. Springer.
96. Silfverberg, M., et al. *Part-of-Speech Tagging using Conditional Random Fields: Exploiting Sub-Label Dependencies for Improved Accuracy*. in *ACL 2014*. 2014. Association for Computational Linguistics.
97. Wang, P., et al., *Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network*. 2015, arXiv.
98. Chiche, A. and B. Yitagesu, *Part of speech tagging: a systematic review of deep learning and machine learning approaches.* Journal of Big Data, 2022. **9**(1): p. 10.
99. Rosenblatt, F., *The perceptron: A probabilistic model for information storage and organization in the brain.* Psychological Review, 1958. **65**: p. 386-408.
*100. Hopfield, J.J., Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America, 1982. **79**(8): p. 2554-2558.*
*101. Hochreiter, S. and J. Schmidhuber, Long Short-Term Memory. Neural Computation, 1997. **9**(8): p. 1735-1780.*
102. OpenAi, et al., *Dota 2 with Large Scale Deep Reinforcement Learning*. 2019, arXiv.
*103. Carbune, V., et al., Fast multi-language LSTM-based online handwriting recognition. International Journal on Document Analysis and Recognition (IJDAR), 2020. **23**(2): p. 89-102.*
104. Fasoli, A., et al., *4-bit Quantization of LSTM-based Speech Recognition Models*. 2021, arXiv.
105. Howard, J. and S. Ruder, *Universal Language Model Fine-tuning for Text Classification*. 2018, arXiv.
106. Peters, M.E., et al., *Deep contextualized word representations*. 2018, arXiv.
107. Almagro Armenteros, J.J., et al., *SignalP 5.0 improves signal peptide predictions using deep neural networks.* Nature Biotechnology, 2019. **37**(4): p. 420-423.

108. Hannigan, G.D., et al., *A deep learning genome-mining strategy for biosynthetic gene cluster prediction.* Nucleic Acids Research, 2019. **47**(18): p. e110.
109. Devlin, J., et al., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv:1810.04805 [cs], 2019.
110. Vaswani, A., et al., Attention Is All You Need. arXiv:1706.03762 [cs], 2017.
111. Raffel, C., et al., *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.* arXiv:1910.10683 [cs, stat], 2020.
112. Elnaggar, A., et al., *ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing.* arXiv:2007.06225 [cs, stat], 2021.
113. Ji, Y., et al., *DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome.* Bioinformatics, 2021. **37**(15): p. 2112-2120.
114. Teufel, F., et al., *SignalP 6.0 predicts all five types of signal peptides using protein language models.* Nature Biotechnology, 2022. **40**(7): p. 1023-1025.
115. *Rapid and Sensitive Protein Similarity Searches | Science.*

Chapter 2: NP-BERT - An NLP approach to natural product comparison

2.1 Chapter Preface

My research was initially focused on the comparison of biosynthetic gene clusters. I validated the comparisons using molecular labels curated by experts. The dataset of labels was very small and was not up to date with the latest dataset of experimentally verified gene clusters. As a heuristic, I moved to chemical similarities with our in-house GRAPE/GARLIC software. Unfortunately, GRAPE/GARLIC does not scale with large amounts of molecules. It also was unable to break down many of the new experimentally verified gene clusters' metabolites. Based on our tests with our LEMONS software, I moved to FCFP6 as an alternative. It too failed to capture many of the biosynthetic similarities that my gene cluster comparison software was picking up on. Recently, the NPClassifier dataset was released with a molecular classification tool. The NPClassifier tool was not intended to be used as a chemical similarity metric, but rather exclusively for label projection and classification. I developed NP-BERT to create a new metric for molecular comparison. It meets a need that every comparison tool in the field was unable to.

For this work, I developed all of the deep learning training frameworks for the customised linked-tasked learning regimen I propose. My framework utilises a variety of public tools in completely new ways. I also used public data to train the model. To

demonstrate biosynthetic conservation, I worked with Mathusan Gunabalasingam to map molecules to mass spectra using his in-house framework called MAPLE.

2.2 Abstract

Natural products are a rich source of medically relevant molecules. Characterised by their large size and structural complexity, methods for comparing natural products are more nuanced than their synthetic chemical counterparts. A variety of different software has been developed for the classification and comparison of natural products, many of which are based on molecular substructures derived from the Extended Connectivity Fingerprint. With the release of self-referencing strings (SELFIES), a natural language processing approach can be taken. In this work, we present NP-BERT - a large language model capable of representing natural products as vector representations for large-scale comparison and classification. We evaluate different fingerprints in comparison with NP-BERT embeddings using a natural product labelling recovery task. Using scalable clustering techniques we demonstrate the rapid discovery of biosynthetic analogues within the microbial metabolite space.

2.3 Introduction

Historically, natural products (NPs) have been a rich source of medicinal molecules. NPs continue to play a large role in drug discovery with 64.9% of FDA-

approved anti-cancer agents being NP derived since 1981.[1] Natural products are defined as molecules produced by living organisms. Because they are natural metabolites, NPs have been refined by the evolutionary process for biological processes.[2, 3] They display a wide variety of activities including antimicrobial properties and immunomodulation. [4, 5] For many years, publicly available datasets were sparse, but recent curation efforts have resulted in the democratisation of NP research. The Dictionary of Natural Products was once considered the only comprehensive source of chemical data on natural products, but now databases such as COCONUT and NPAtlas are freely accessible. [6-8] Activity data has also become more readily available with curation efforts such as NPASS focusing on natural product activity screenings. [9]

While data has become more accessible, cheminformatics techniques effective in the NP space are still lagging behind. Much of the tooling has been developed with synthetic chemistry libraries in mind. MACCS fingerprints are limited to 166 predefined substructures optimised for molecular classification. [10] With our LEMONS framework, we demonstrated that these substructures are not useful in NP comparison.[11] High throughput screening libraries display limited chemistry, possibly attributed to their cost-effective synthesis and attention paid to Lipinski's rule of 5; natural products deviate from this norm.[2, 3, 12] The large, 3D structures and complex chemistries of NPs cannot be captured by small predefined substructure-bound fingerprinting techniques. There have been efforts made to create fingerprints consisting of natural product substructures such

as the NC-MFP and GRAPE; as novel NPs are continually discovered, only time will tell how long hard-coded substructure-based techniques will remain effective. [11, 13]

To bypass the limitations of substructure-based fingerprinting techniques, hashed fingerprints dynamically generate chemical features.[14] The extended connectivity fingerprint, up to four bonds (ECFP4), has demonstrated the best performance in drug analogue recovery studies, while the Function Class Fingerprint, up to six bonds (FCFP6) has demonstrated the performance in NP derivative comparison. [11, 15] To perform fast comparisons using hashed fingerprints, substructures are combined through a folding process and the fingerprint is reduced to a typical 1024 bits. Using the Tanimoto Index to compare bit vectors, structural similarity can be measured.[16] Unfortunately, the effectiveness of bit vectors is lost as dimensionality is reduced. Using the data sketch MinHash, differences between highly sparse, high-dimensional vectors can be approximated. [17] This technique was demonstrated with the MinHash Fingerprint (MHFP) and LSHForests.[15, 18]

While ECFP featurization in combination with Tanimoto indexing has remained a staple in molecular classification and comparison, there are flaws. As demonstrated with our tool LEMONS, the biosynthetic changes commonly made by microbial organisms, are not relatively reflected by hashed fingerprinting techniques.[11] While our GRAPE/ GARLIC method was previously demonstrated as an effective method for NP

comparison, it is not scalable with larger datasets. More recently, scalable NP comparison has been demonstrated using deep learning technologies.[19] With the liberation of NP datasets, training large deep-learning models is more feasible. In 2021, a massive dataset of 73,607 NPs with expert knowledge-based labels was released. Molecules were annotated with Pathway (specialised metabolism), Superclass (chemotaxonomic information) and Class (structural) labels. In combination with the dataset, a deep learning-based classifier was released called NPClassifier.[20] It used counted ECFP4 fingerprints as an input vector for a feed-forward neural network. While limited to the hashed substructures, the model successfully annotates the majority of incoming molecules with detailed NP information. As with all classification tasks, labelling fails when a class is too small for learning.

There have been efforts to move away from the hashed fingerprint featurization of molecules and instead focus directly on the graph string language SMILES. A regex-based technique was developed to tokenise SMILES directly into words capable of being learned by an LSTM RNN.[21] Byte-pair encoding (BPE) has also been applied to making SMILES a learnable language for large language models.[22] Recently, a new dialect of the SMILES molecular language was released, called SELFIES. It allows for a simplification of the SMILES molecular graph string, into human-readable units reflective of the molecule's structure. The authors also demonstrated a learned latent space using SELFIES over SMILES can contain more valid structures. [23] The SELFIES

dialect contains words less abstract than the hashed molecular radii of conventional fingerprints, but are still generalisable enough to facilitate the LLM's propensity to learn a chemically relevant latent space.

In this work, we propose a new vector-based method for comparing natural products. We present NP-BERT - an LLM trained with the SELFIES dialect for representing natural products. We demonstrate its superior ability to perform natural product classification recovery over common fingerprinting techniques (i.e. ECFP6, FCFP6 and Kekota-Roth). We visualise the latent space of NPs encoded by NP-BERT. We also perform clustering of a large dataset of NPs and demonstrate an intra-cluster conserved biosynthesis using metabolomics.

2.3 Methodology

2.3.1 Model Architecture

The NP-BERT model was based on the standard RoBERTa architecture. Using the Hugging Face library, a RoBERTa model was instantiated with a hidden size of 768 and a maximum sequence length of 2048; the size limit was implemented due to memory constraints. [24, 25] To train the model to perform classification, a custom linked-task architecture was designed. (Supplementary Figure S2.1) The custom classification head differs from the standard sequence classification head by separating the pooling layer and

dense layers. Output from the pooling layer can be passed in a hierarchical manner onto proceeding finer-task classification heads. A separate classification head was created for each tier in the ontology of natural product classification (pathway, superclass and class). The three custom classification heads were stacked so the pooling output of the pathway classification head would be directly passed to the pooling layer of the superclass classification head followed by the class classification head.

2.3.2 Input/Outputs

To utilise an LLM, molecules must be represented as sentences. Molecules were first represented as SMILES strings and then converted to Self-Referencing Embedded Strings (SELFIES). SELFIES are easily tokenizable. The NP-BERT tokenizer was comprised of a combination of custom components. A custom SELFIES pretokenizer and decoder were designed using the “selfies” python package and the “tokenizers" package by Hugging Face (HF). A chirality normaliser was designed using RDKit for removing the chirality of a SMILES string before converting it to the SELFIES format. [26] Word-level tokenisation was used to split the SELFIES string. A class token was appended to every molecule. The vocabulary was generated using the NP-Classifier curated dataset of 73,607 annotated small molecules and Zinc15’s 308,035 biologics. [27]

2.3.3 Pretraining

A standard masked-language modelling (MLM) task was used to pre-train NP-BERT. HF's implementation of MLM within their "transformers" package was used for coordinating masked inputs and labelling during training ("RoBERTAForMaskedLM" in combination with "DataCollatorForLanguageModeling"). The model was trained on both datasets in succession. First NP-BERT was trained on the Zinc15 Biologics dataset, for 38,639 global steps across 10 epochs with a batch size of 16. The model was then trained on the NPClassifier dataset for 7,719 global steps across 10 epochs with a batch size of 16. Both datasets were broken into training, testing and validation datasets using a 64-20-16 split. To minimise memory consumption and training time, the Stage 2 Optimiser with Parameter offloading from Microsoft's DeepSpeed package was used. [28] Precision was limited to 16-bit. Learning rates were automatically determined using PyTorch Lightning. [29]

2.3.4 Linked Task Fine-tuning

NP-BERT was fine-tuned using the multi-labelled NPClassifier dataset. During collation, each molecule was passed with each of the three levels of classification labels: class, superclass and pathway. For the loss calculation, each custom classification head calculated a separate loss based on its corresponding label. Cross-entropy losses across the class, superclass and pathway classification heads were summed together and passed to the optimiser. PyTorch Lightning's implementation of the Distributed Data-Parallel

(DDP) optimiser was used to distribute training across multiple GPUs. Learning rates were automatically determined using PyTorch Lightning. Using the "BAAL" python package, a custom active learning training procedure was used. [30] For the first epoch, the most informative 1000 samples were used for training. For every proceeding epoch, 10,000 examples were sampled from the training data but only the most informative 2000 were added to the rolling dataset. The active learning sampling continued until all sentences are integrated. The most informative samples were determined using the "Bayesian active learning by disagreement" or BALD heuristic. [31] The model was trained for a total of 60 epochs. Training results are reported in Section 2.4.1.

2.3.6 Quantisation, Optimisation and Acceleration

To maximise performance during inference, NP-BERT was optimised with Microsoft's ONNX package.[32] Models were quantised and exported with dynamic axes. The ONNXRuntime (ORT) framework was used for inference downstream. A custom transformers' pipeline was made to allow for the hierarchical prediction of the custom classification heads. Modifications were also made to allow for the usage of ORT alongside PyTorch and TensorFlow.

2.3.6 Experiments

2.3.6.1 Comparison to NPClassifier

To compare performance to NPClassifier, a dataset of failed classifications was created. Any molecule within the NPClassifier dataset that NPClassifier was unable to classify correctly was pooled. For pathway, superclass, and class a total of 338, 709 and 1176 compounds were incorrectly classified respectively. Accuracy on this dataset was calculated. Results are reported in Section 2.4.2.

2.3.6.2 Natural Product Label Recovery using Embedding Distances versus Fingerprint Dissimilarities

There are other popular methods for comparing molecules including the extended connectivity fingerprint (ECFP), functional class fingerprint (FCFP) and Klekota-Roth fingerprints.[14, 33] ECFP and FCFP fingerprint a molecule by sampling the surroundings of each atom and converting them into discrete features using a hashing function. The size of the surroundings to be sampled is quantified by an “atom diameter”. For example, ECFP6 uses an atom diameter of 6 and sampled substructures will have a maximum width of 6 bonds. ECFP differs from FCFP in the way that certain substructures are treated. FCFP abstract substructures into functional groups based on their roles as a pharmacophore. Klekota-Roth fingerprints are based on the presence or absence of certain substructures. Similar to the FCFP abstraction, the substructures in Klekota-Roth were selected based on their relevance to biological activity. To compare

the resolution of NP-BERT embeddings, these three fingerprinting techniques were used to find similar molecules. Molecules were then evaluated based on shared natural product classifications.

Non-folded FCFP6, ECFP6, and Klekota-Roth fingerprints for each molecule in the validation dataset were calculated. FCFP6 and ECFP6 used the implementations found in RDKit. Klekota-Roth was implemented using the SMARTS provided in the original paper. RDKit's substructure search was used to determine whether or not a query Kloka-Roth substructure was found in the molecule. For each molecule, its fingerprints were converted to the data sketch MinHash. The MinHashes were used to create an LSHForest and the nearest neighbour for each molecule was determined per fingerprint. Points were awarded when the nearest neighbours shared mutual labels.

To evaluate NP-BERT in a similar context, every molecule in the validation dataset was tokenised and embedded by the base NP-BERT model. Each classification token embedding was passed in succession through the hierarchical classification heads. The embeddings at each tier of classification were stored in separate vector datasets. The framework "Faiss" was used to create a flat L2 Index across the vector datasets. [34] The flat L2 Index allows for a highly optimised nearest neighbour search. The nearest molecule at each classification level was found using the nearest neighbour technique optimised on Euclidean distances. Results are reported in Section 2.4.3.

2.3.6.3 Natural Product Latent Space

To showcase the LLM's understanding of microbial natural products, the latent space of NPBERT was plotted using an internal dataset of 49,523 molecules. Each molecule was embedded, and the different levels of molecular classifications were predicted. The embeddings were projected to two dimensions using the RAPIDS ML GPU implementation of UMAP (Uniform Manifold Approximation and Projection).[35, 36] Settings for UMAP were set to a minimum of two neighbours and a minimum euclidean distance of 0.1. The molecules were plotted using the official "umap" plotting library. To determine scaffold families within the plot, a second projection with 128 dimensions was clustered using the RAPIDS ML GPU implementation of HDBSCAN. [37] Clustering was optimised using the RAPIDS ML implementation of the silhouette score.[38] Results are reported in Section 2.4.4.

2.3.6.4 Conserved Biosynthesis within Clusters

The unsupervised clustering of the NP-BERT embeddings resulted in clusters of highly related molecules. Many highly related molecules can be synthesised by a single biosynthetic gene cluster. Incomplete biosynthesis and modifications can result in highly similar structures. [39] Typically in a crude microbial extract, a parent scaffold and its derivatives are found. To investigate if the members of a single group (determined using

HDBSCAN) shared the biosynthetic pathway, the contents of crude extracts were analysed for the presence of intra-group derivatives.

We have an internal fermentation library of microbial crude extracts. Extracts are processed using tandem liquid chromatography-mass spectrometry (LC-MS). An in-house pipeline called MAPLE (MetAbolomics Peaks Logic Engine) is used to confidently mass-match spectral peaks to known molecules using structural information. To assess if clustered NP-BERT molecules co-occur in metabolism, HDBSCAN clusters were cross-referenced with the database of extracts. To visualise co-occurrence, network graphs were created for three HDBSCAN clusters. Each metabolite within the cluster was represented as a node. When two metabolites from a single cluster were detected in the same extract, an edge was drawn. For repeated instances of co-occurrence, the edge weight was increased. Network graphs were visualised using Gelphi.[40] Results are reported in Section 2.4.5.

2.4 Results

2.4.1 Linked Task Fine-tuning

After 60 epochs, the validation accuracy for the model in prediction pathways was 99.08%, superclasses 98.03%, and classes 89.39%. The test accuracy of the model in

predicting pathways was 99.18%, superclasses 98.26% and classes 94.19%. Loss changes over across the 60 epochs can be found in Supplementary Figure S2.2.

2.4.2 Comparison to NP Classifier

Across the entire dataset, NPClassifier is able to achieve pathway, superclass and class accuracies of 91.71%, 89.33%, and 86.36% respectively. NPClassifier incorrectly predicted the pathway, superclass, and class for 338, 709 and 1176 compounds respectively. Of the 338 failed pathway classifications, NP-BERT was able to correctly predict 238 (70.4%). Of the 709 failed superclass classifications, NP-BERT was able to correctly predict 522 (73.6%). Of the 1176 failed superclass classifications, NP-BERT was able to correctly predict 694 (59.0%).

2.4.3 Natural Product Label Recovery using Embedding Distances and Fingerprint Dissimilarities

Natural product labels were recovered most comprehensively with NP-BERT's nearest neighbours. Across all hierarchies, it outperformed ECFP6, FCFP6 and Klekota-Roth. While Klekota-Roth was able to recover more pathway labels than ECFP6 and FCFP6, it performed the worst in terms of superclass and class recovery. FCFP6 slightly outperformed ECFP6 across all recovery tasks. All results are summarised in Table 2.1.

An example of an incorrect nearest neighbour calculated from ECFP6 dissimilarities versus NP-BERT embeddings is found in Figure 2.1.

2.4.4 Natural Product Latent Space

The plotted natural product latent space of the 49,523 molecules can be seen in Figure 2.2. Using HDBSCAN 25,512 molecules were clustered into 1,066 scaffold groups (Silhouette score = 0.635). Many of the molecules were not classified because a minimum size constraint of 10 was applied.

2.4.5 Conserved Biosynthesis

Three of the groups discovered by HDBSCAN were chosen for further explanation: tetracyclines, thiazoles and statins.

2.4.5.1. Tetracyclines

The group consisted exclusively of tetracyclines and tailored derivatives including Dehydrochlortetracycline, 7-Bromo-6-demethyltetracycline, Bromotetracycline, Anhydrotetracycline, and Amicycline. After cross-referencing with the dataset of processed fermentations, all detected molecules from this group were found to co-occur in microbial metabolism (Figure 2.3).

2.4.5.2 Thiazoles

The group consists mainly of thiazoles including 14-Hydroxycystothiazole, 14,15 Dihydroxycystothiozole, Cystothiazole-A, Melithiazol-G, Melithiazol-I and Myxothiazol. After cross-referencing with the dataset of processed fermentations, all detected molecules from this group were found to co-occur in microbial metabolism (Figure 2.4).

2.3.6.3 Statins

Members of this group were all polyketides derived, with many being natural statins including Monocolin, Mevastatin, 6β-Hydroxymethylsimvastatin, Eptastatin, and Lovastatin. After cross-referencing with the dataset of processed fermentations, all detected molecules from this group were found to co-occur in microbial metabolism (Figure 2.5).

2.5 Discussion

2.5.1 Finetuning LLM with Linked Task Training and Active Learning

As shown in Section 2.4.1 and Section 2.4.2, NP-BERT demonstrates a superior classification ability over NPClassifier. Its accuracy is higher overall across the dataset and it does not share the same weaknesses. It was explained that NPClassifier was unable

to learn some of the rarer labels because there was not a sufficient population for the class. With the transformers architecture, combined with the SELFIES tokenizer and using active learning strategies, we have demonstrated the rarer classes are indeed learnable.

There are many differences between the two approaches, any of which could be contributing factors to superior performance. In terms of input, SELFIES is a more verbose language than ECFP hashes; its lack of abstraction pairs well with the transformers' architecture. With SELFIES, the transformer's attention heads are given the freedom to find the most important atoms of the molecule when representing it in the latent space. In addition, the use of BALD to find the most useful training examples during fine-tuning would result less overfitting for overrepresented classes.

One of the techniques pioneered in this work is linked task learning. Hierarchical classification is understudied and only a few frameworks have been designed to solve this problem.[41] Hierarchical classification is a specialised case of multi-label, where an item is given multiple labels but the labels are organised in a hierarchical tree (e.g. Fruit ➡ Apple ➡ Granny Smith). The most common solutions ignore the hierarchy and instead treat the problem as a flat multi-label classification; this is the approach taken by NPClassifier.[20, 42] Other approaches typically involve traversing a tree of sub-classifiers per parent class. This is extremely slow when the hierarchy is very wide.[43]

There are a few novel solutions such as the Deep Hierarchical Classification system developed by the Alibaba Group [42, 44] The Deep Hierarchical Classification system concatenates embeddings from each classification level in succession. This results in an extremely long representation for the final classification layer and does not scale with very deep hierarchies. The linked task learning methodology proposed in this work simplifies this approach by linking hidden layers of classifiers avoiding concatenation completely. NP-BERT's success demonstrates that the approach is viable and effective. It can be used in other areas of bioinformatics where hierarchies are common (ex. taxonomies, gene ontologies, Enzyme Commission (EC) Numbers, Medical Subject Heading (MeSH) headings).

2.5.2 Effectiveness as a new molecular comparison metric

As demonstrated before with the LEMONS tool, traditional fingerprinting methods are not effective at comparing NPs. As shown in Figure 2.1, traditional fingerprinting techniques pay too much weight to small changes in the molecule to effectively score biosynthetic modifications. In this case, FCFP6 could not discern the similarity between a polyketide dimer and a related monomer; instead, it scored the similarity near identically to a terpenoid.

The latent space of NPBERT is tuned more effectively for representing NPs in quadrants related to their biosynthesis. In Figure 2.2, there is a clear separation of molecules belonging to different biosynthetic pathways. In Section 2.4.5, it was demonstrated that the unsupervised clustering of molecules can create groups of molecules truly related on a biosynthetic level. We demonstrated three separate groups of highly related molecules were co-expressed in microbial fermentations. The resolving power of NPBERT's embeddings can be used to effectively score the biosynthetic similarity of molecules.

2.5.3 Future Work

We demonstrated LLMs can understand NPs with a high degree of resolution if presented with the data in the correct context. The effectiveness of the latent space in representing structures with high regard for their biosynthesis opens the potential for activity prediction. NPClassifier showed that much of the superclass and class level annotation can be used as a heuristic for finding favourable anti-malarial candidates. [20] In addition, ChemBERTA was used to perform activity prediction across a variety of different datasets. [22] NP-BERT's embeddings can be repositioned and fine-tuned in a similar manner to find potential activities tied to NP scaffold families. Further exploration should be made in understanding what molecular features are attended to in high-performing attention heads. Possible future work could explore which chemical moieties

are critical in predicting NP classes and if the information is capable of being extrapolated to activity for pharmacophore discovery.

2.6 Figures and Tables

Table 2.1 Accuracy of different fingerprinting techniques, measured using conserved labels of nearest neighbours for each molecule in the validation dataset (n = 12,534). The Euclidean distance of NPBERT embeddings performed best at the label recovery task.

Fingerprinting Technique	Pathway	Super Class	Class
	Nearest Neighbour Accuracy		
ECFP6	0.857	0.730	0.602
FCFP6	0.868	0.763	0.626
Klekota-Roth	0.880	0.690	0.554
NPBERT	**0.947**	**0.880**	**0.742**

Figure 2.1 Example of Natural Product Classification Recovery with FCFP6 versus NP-BERT. (A) Non-redundant FCFP6 with Jaccard dissimilarities found an incorrect nearest neighbour. All classifications were not recovered. (B) NP-BERT with Euclidean distances was able to successfully find a derivative of the query molecule with all classifications recovered.

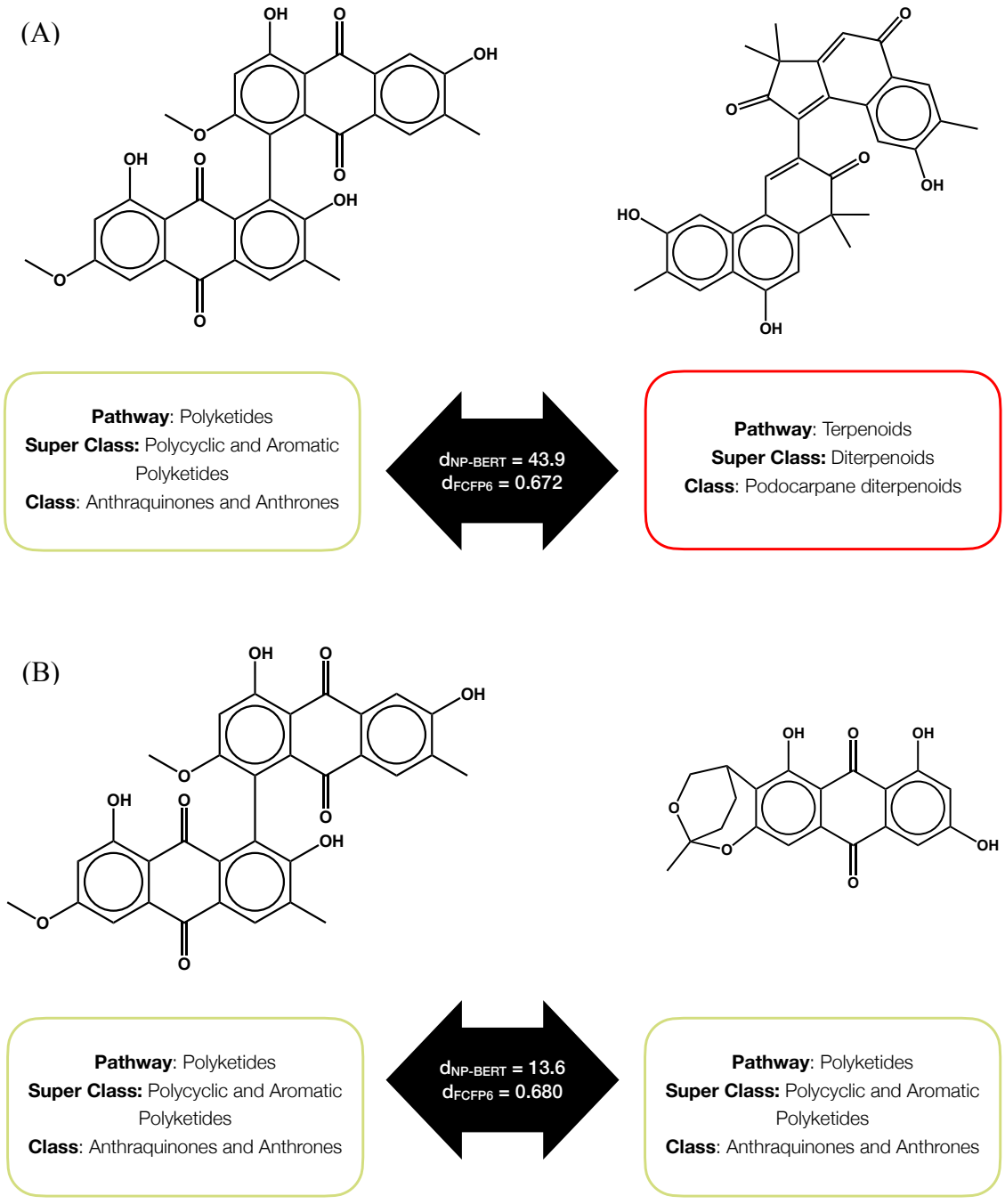


Figure 2.2: The natural product latent space of NP-BERT demonstrated with a dataset of molecules (n=49,523) projected to two dimensions and plotted with UMAP and data shader. Molecules are colourised by the predicted pathway labels. The molecules are separated into clusters of distinct pathways with minimal overlap.

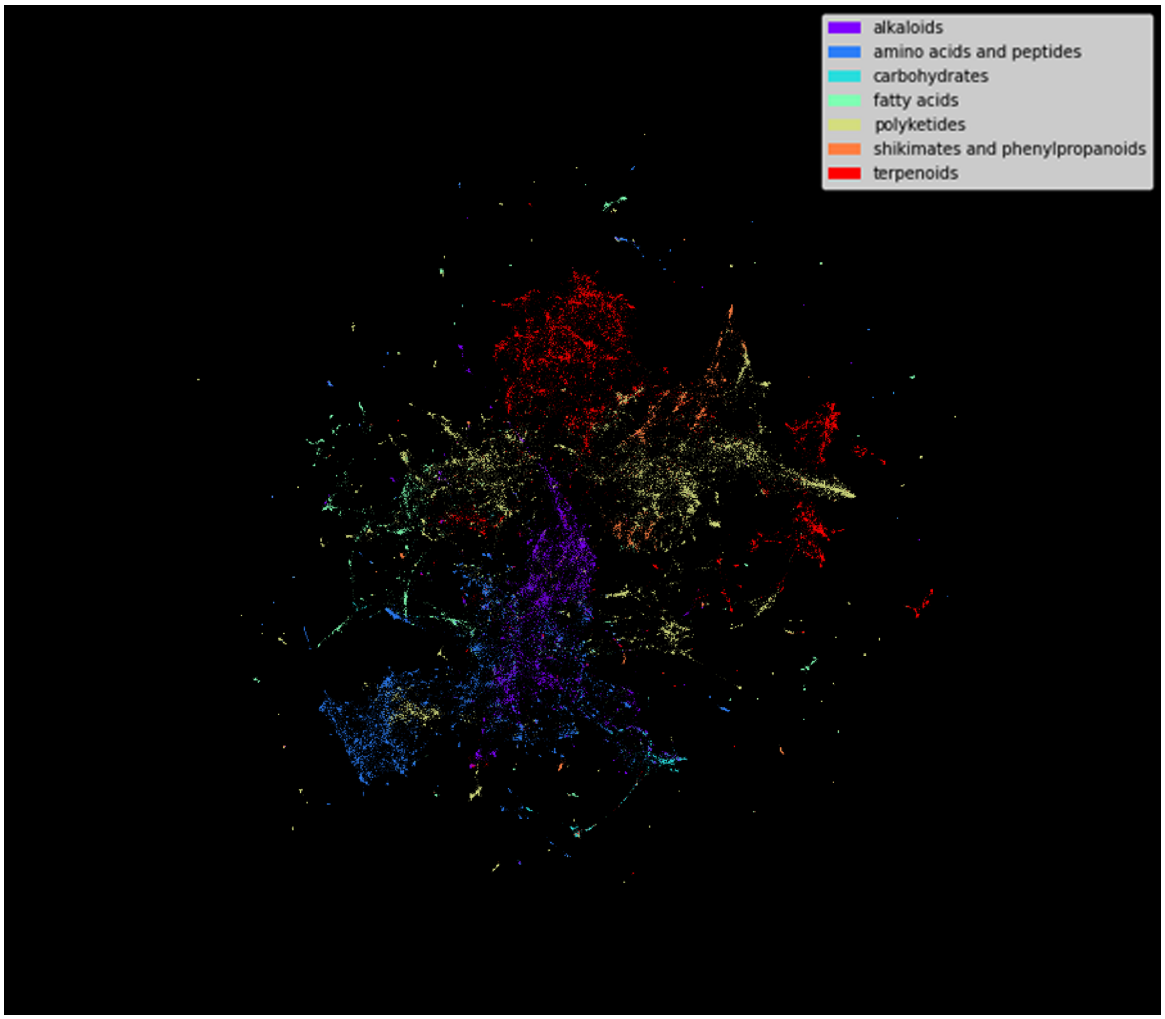


Figure 2.3: Upon inspection of one of the "polyketide" HDBSCAN clusters, it consisted only of tetracyclines. An internal dataset of crude extracts was used to visualise the co-occurrence of the metabolites within fermentations. To create the network graph, each tetracycline was represented as a node and upon co-occurrence with another metabolite, an edge was drawn. Many of the tetracyclines within the HDBSCAN cluster are co-expressed.

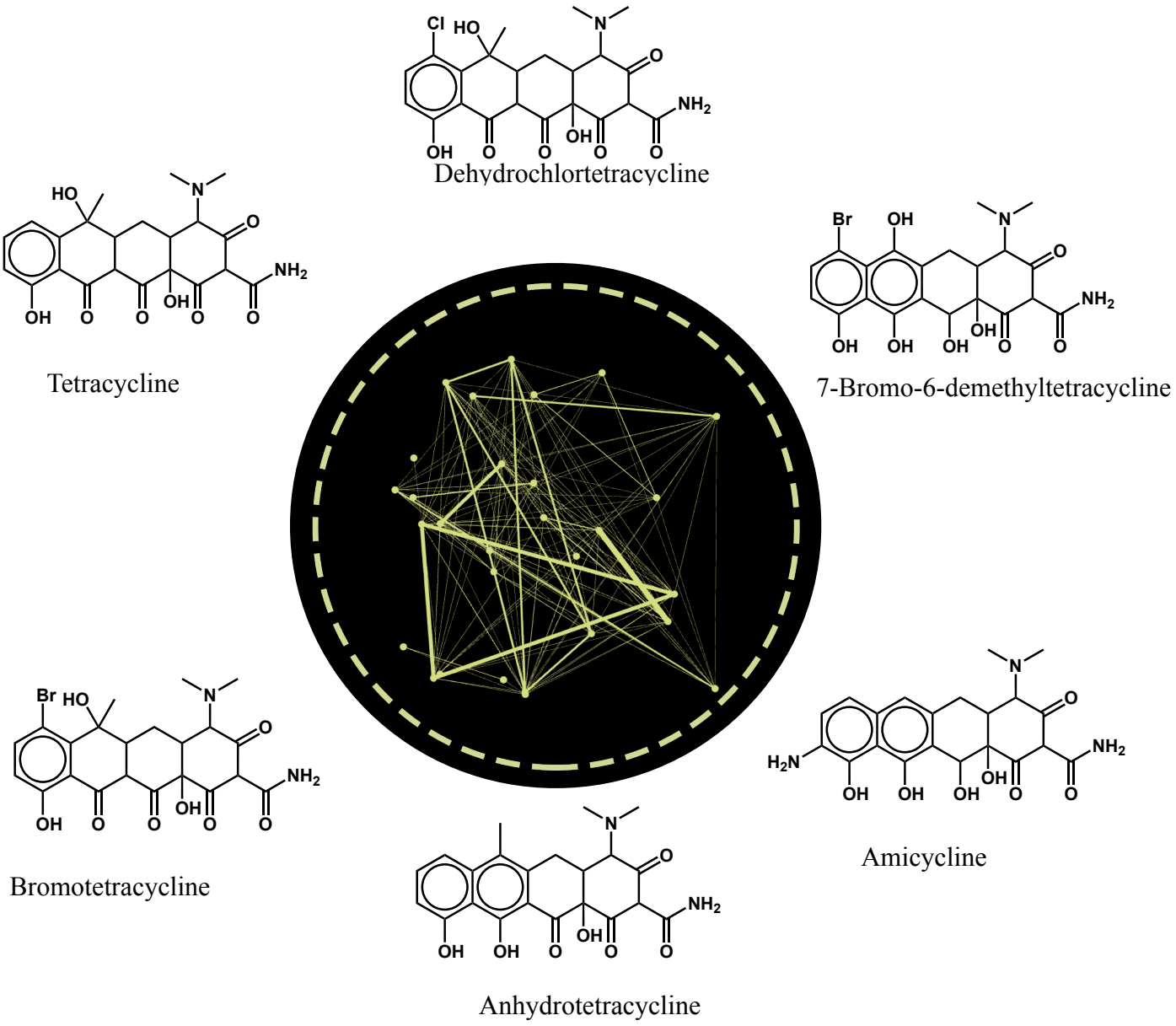
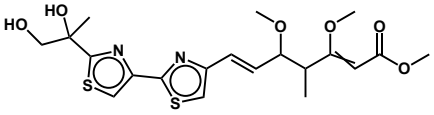
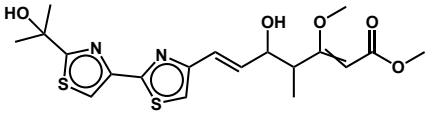


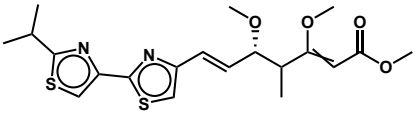
Figure 2.4 Upon inspection of one of the "amino acids and peptides" HDBSCAN clusters, it consisted only of thiazoles. An internal dataset of crude extracts was used to visualise the co-occurrence of the metabolites within fermentations. To create the network graph, each thiazole was represented as a node and upon co-occurrence with another metabolite, an edge was drawn. Many of the thiazoles within the HDBSCAN cluster are co-expressed.



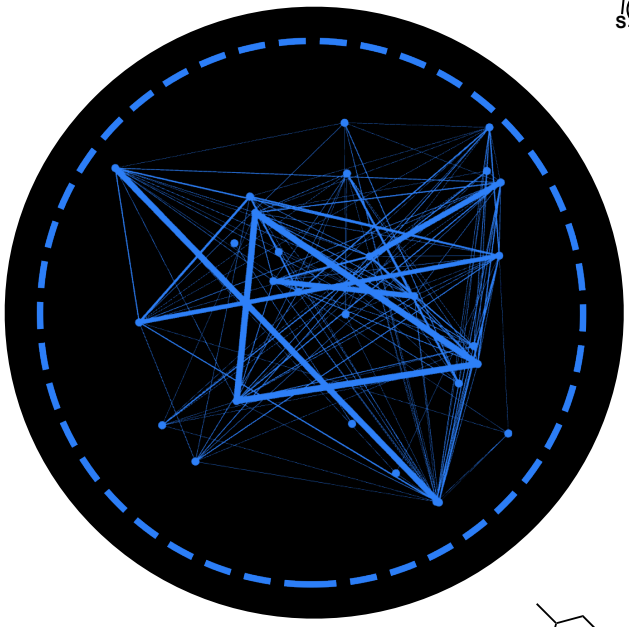
14,15 Dihydroxycystothiozole

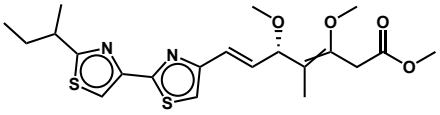


14-Hydroxycystothiazole

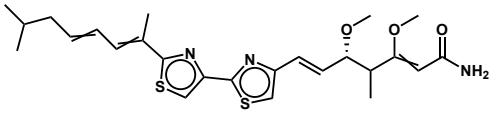


Cystothiazole-A

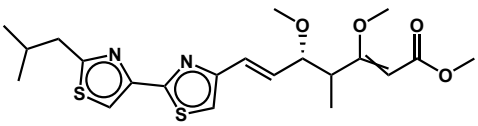




Melithiazol-G

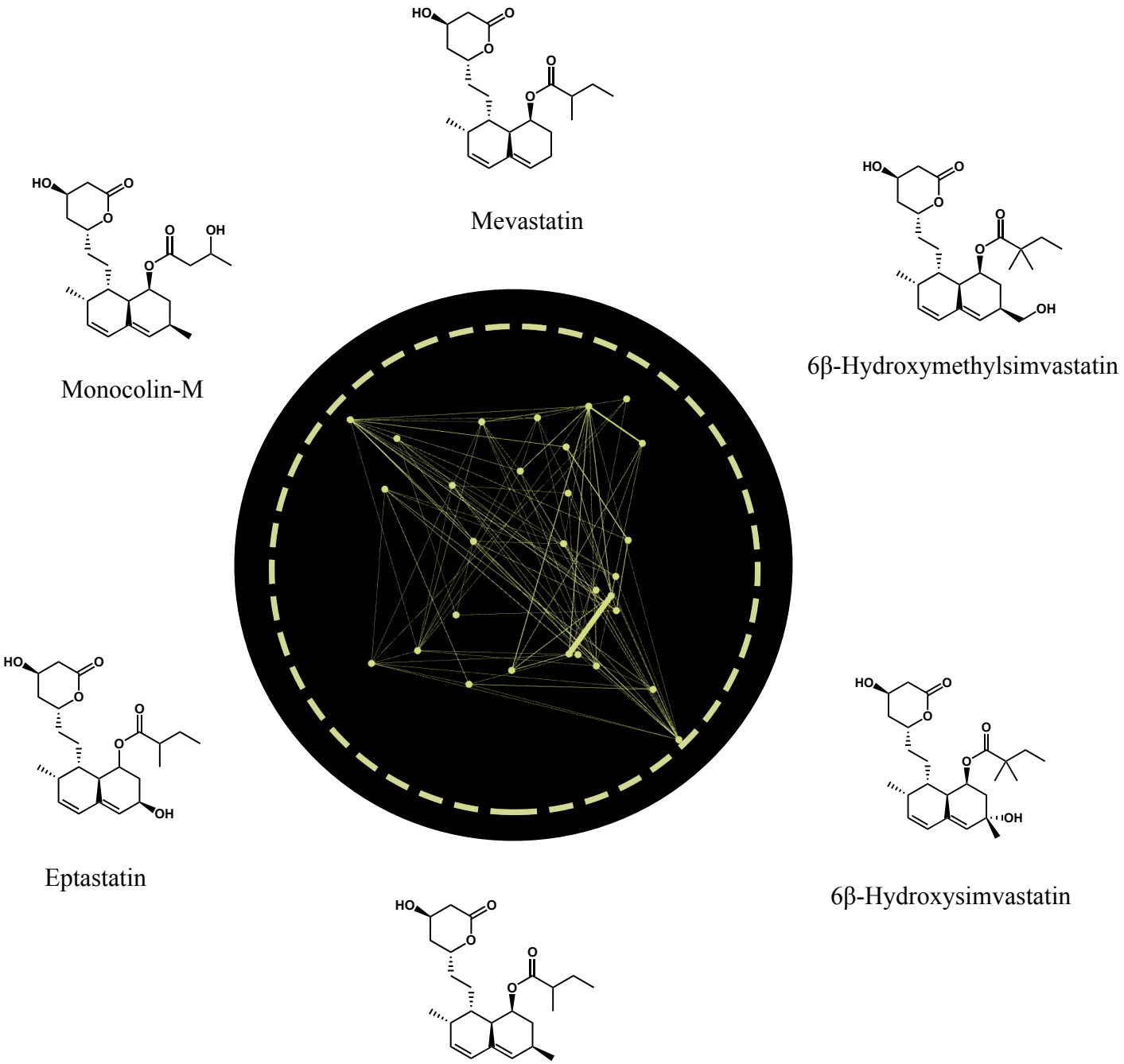


Myxothiazo

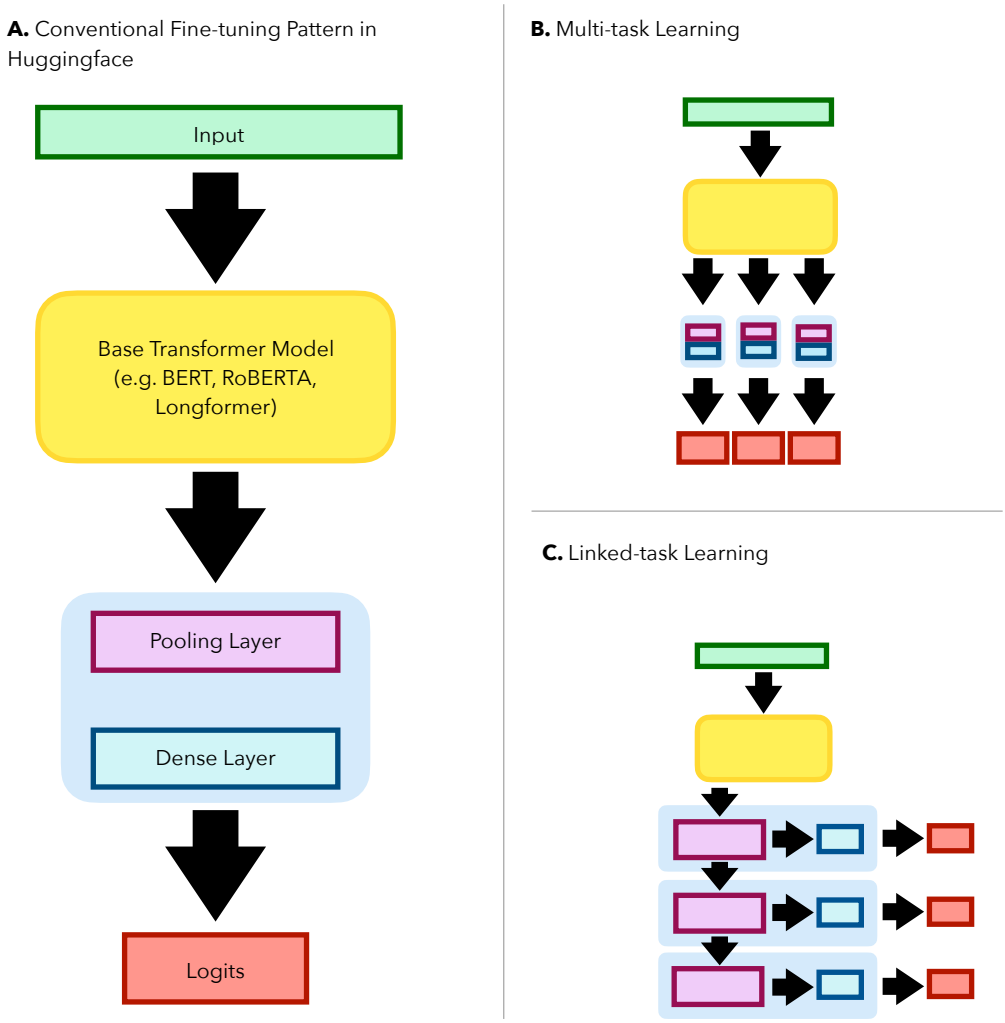


Melithiazol-I

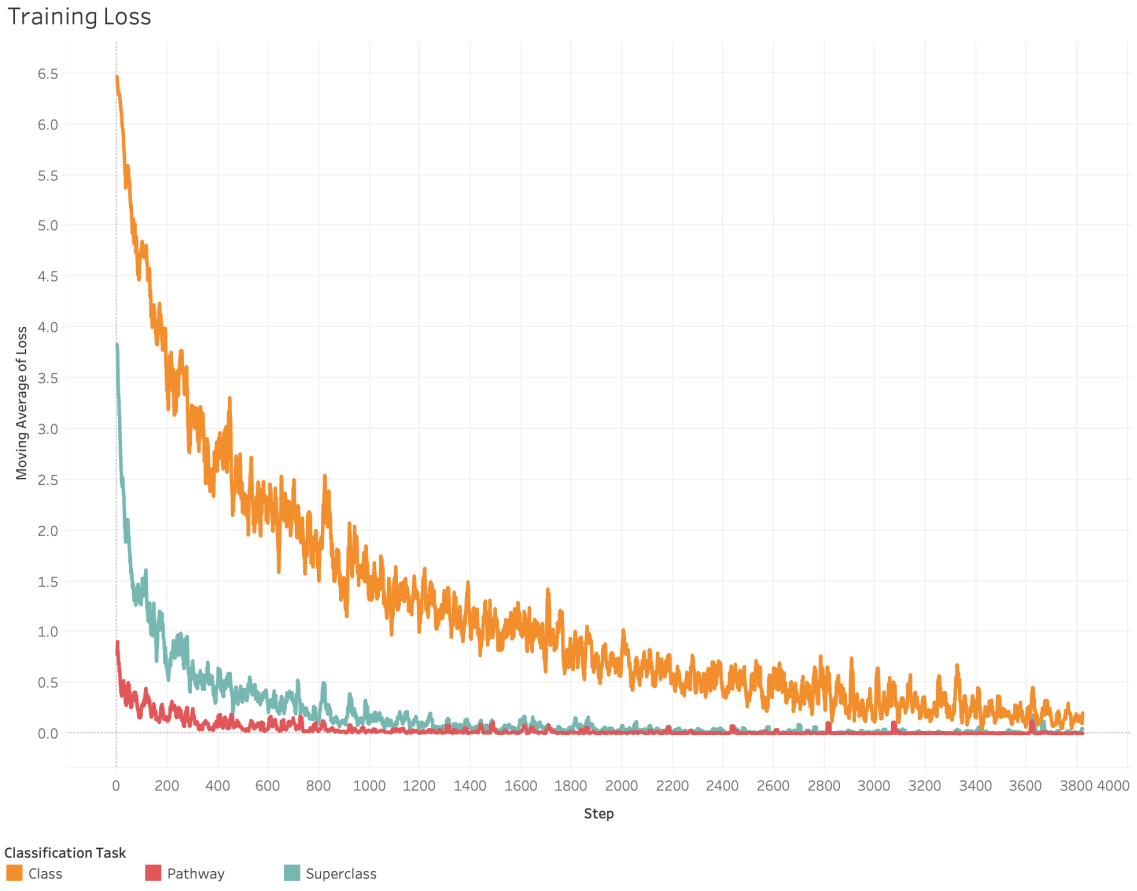
Figure 2.5 Upon inspection of one of the "polyketide" HDBSCAN clusters, it consisted mainly of natural statins. An internal dataset of crude extracts was used to visualise the co-occurrence of the metabolites within fermentations. To create the network graph, each metabolite was represented as a node and upon co-occurrence with another metabolite, an edge was drawn. Many of the molecules within the HDBSCAN cluster are co-expressed.



Supplementary Figure S2.1(A) Huggingface's implementation of the RoBERTa For Sequence Classification used the emissions from a base model (size = length of the input sequence x hidden size) and passes it through two linear transformation layers. Because RoBERTa and T5 lack a pooling layer, the first layer of the classification acts as such changing the matrix to a square using a linear transformation with an additive bias (size = hidden size x hidden size). After a Tan-H activation, the embedding is passed through the dense layer. This is another linear transformation layer but it reshapes the matrix to become a flat logit vector (size = the number of classes). A softmax function normalises the logits into probabilities. Any inconsistencies are measured typically with cross-entropy loss and back-propagated through the network. (B) In multi-task learning, the three heads generate logits in parallel. Biases are only shared within the transformer. (C) In linked-task learning, the pooling layer's square output is passed in succession, giving the proceeding classification heads more biases to utilise.



Supplementary Figure S2.2 A training loss curve across all training epochs for each of the labels. With Linked-Task Learning, the loss for pathway and superclass prediction reaches loss values below 1.0 within 200 steps. Class labelling takes much longer but does eventually reach a similar value.



2.7 References

1. Newman, D.J. and G.M. Cragg, Natural Products as Sources of New Drugs over the Nearly Four Decades from 01/1981 to 09/2019. Journal of Natural Products, 2020. **83**(3): p. 770-803.
2. Stratton, C.F., D.J. Newman, and D.S. Tan, Cheminformatic comparison of approved drugs from natural product versus synthetic origins. Bioorganic & medicinal chemistry letters, 2015. **25**(21): p. 4802-4807.
3. Atanasov, A.G., et al., Natural products in drug discovery: advances and opportunities. Nature Reviews Drug Discovery, 2021. **20**(3): p. 200-216.
4. Jantan, I., W. Ahmad, and S.N.A. Bukhari, Plant-derived immunomodulators: an insight on their preclinical evaluation and clinical trials. Frontiers in Plant Science, 2015. **6**.
5. Schneider, Y.K., Bacterial Natural Product Drug Discovery for New Antibiotics: Strategies for Tackling the Problem of Antibiotic Resistance by Efficient Bioprospecting. Antibiotics, 2021. **10**(7): p. 842.
6. Buckingham, J., Dictionary of Natural Products. 1993: CRC Press. 1306.
7. Sorokina, M., et al., COCONUT online: Collection of Open Natural Products database. Journal of Cheminformatics, 2021. **13**(1): p. 2.
8. van Santen, J.A., et al., The Natural Products Atlas 2.0: a database of microbially-derived natural products. Nucleic Acids Research, 2022. **50**(D1): p. D1317-D1323.
9. Zhao, H., et al., NPASS database update 2023: quantitative natural product activity and species source database for biomedical research. Nucleic Acids Research, 2023. **51**(D1): p. D621-D628.
10. Durant, J.L., et al., *Reoptimization of MDL keys for use in drug discovery.* Journal of Chemical Information and Computer Sciences, 2002. **42**(6): p. 1273-1280.
11. Skinnider, M.A., et al., Comparative analysis of chemical similarity methods for modular natural products with a hypothetical structure enumeration algorithm. Journal of Cheminformatics, 2017. **9**: p. 46.
12. Doak, B.C. and J. Kihlberg, Drug discovery beyond the rule of 5 - Opportunities and challenges. Expert Opinion on Drug Discovery, 2017. **12**(2): p. 115-119.
13. Seo, M., et al., Development of Natural Compound Molecular Fingerprint (NC-MFP) with the Dictionary of Natural Products (DNP) for natural product-based drug development. Journal of Cheminformatics, 2020. **12**: p. 6.
14. Rogers, D. and M. Hahn, *Extended-Connectivity Fingerprints.* Journal of Chemical Information and Modeling, 2010. **50**(5): p. 742-754.
15. Probst, D. and J.-L. Reymond, A probabilistic molecular fingerprint for big data settings. Journal of Cheminformatics, 2018. **10**(1): p. 66.

16. Bajusz, D., A. Rácz, and K. Héberger, Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? Journal of Cheminformatics, 2015. **7**(1): p. 20.
17. Broder, A.Z. On the resemblance and containment of documents. in Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171). 1997.
18. Bawa, M., T. Condie, and P. Ganesan. LSH forest: self-tuning indexes for similarity search. in the 14th international conference. 2005. ACM Press.
19. Menke, J., J. Massa, and O. Koch, Natural product scores and fingerprints extracted from artificial neural networks. Computational and Structural Biotechnology Journal, 2021. **19**: p. 4593-4602.
20. Kim, H.W., et al., NPClassifier: A Deep Neural Network-Based Structural Classification Tool for Natural Products. Journal of Natural Products, 2021. **84**(11): p. 2795-2807.
21. Schwaller, P., et al., "Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models †Electronic supplementary information (ESI) available: Time-split test set and example predictions, together with attention weights, confidence and token probabilities. See DOI: 10.1039/c8sc02339e. Chemical Science, 2018. **9**(28): p. 6091-6098.
22. Chithrananda, S., G. Grand, and B. Ramsundar, ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. arXiv:2010.09885 [physics, q-bio], 2020.
23. Krenn, M., et al., Self-Referencing Embedded Strings (SELFIES): A 100% robust molecular string representation. Machine Learning: Science and Technology, 2020. **1**(4): p. 045024.
24. Liu, Y., et al., RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs], 2019.
25. Wolf, T., et al., HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs], 2020.
26. Landrum, G., et al., rdkit/rdkit: 2020_03_1 (Q1 2020) Release. 2020, Zenodo.
27. Sterling, T. and J.J. Irwin, *ZINC 15 – Ligand Discovery for Everyone.* Journal of Chemical Information and Modeling, 2015. **55**(11): p. 2324-2337.
28. Rasley, J., et al. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. 2020. Association for Computing Machinery.
29. Falcon, W., *Pytorch lightning.* GitHub. Note: https://github. com/PyTorchLightning/pytorch-lightning, 2019. **3**: p. 6.
30. Atighehchian, P., F. Branchaud-Charron, and A. Lacoste, Bayesian active learning for production, a systematic study and a reusable library. 2020, arXiv.
31. Houlsby, N., et al., Bayesian Active Learning for Classification and Preference Learning. 2011, arXiv.

32. onnx/onnx: Open standard for machine learning interoperability.
33. Klekota, J. and F.P. Roth, Chemical substructures that enrich for biological activity. Bioinformatics, 2008. **24**(21): p. 2518-2525.
34. Johnson, J., M. Douze, and H. Jégou, *Billion-scale similarity search with GPUs*. 2017, arXiv.
35. McInnes, L., J. Healy, and J. Melville, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [cs, stat], 2018.
36. Nolet, C.J., et al., Bringing UMAP Closer to the Speed of Light with GPU Acceleration. 2021, arXiv.
37. McInnes, L., J. Healy, and S. Astels, hdbscan: Hierarchical density based clustering. The Journal of Open Source Software, 2017. **2**.
38. Rousseeuw, P.J., Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 1987. **20**: p. 53-65.
39. Mo, X., et al., Identification of nocamycin biosynthetic gene cluster from Saccharothrix syringae NRRL B-16468 and generation of new nocamycin derivatives by manipulating gene cluster. Microbial Cell Factories, 2017. **16**(1): p. 100.
40. Bastian, M., S. Heymann, and M. Jacomy, *Gephi: An Open Source Software for Exploring and Manipulating Networks.* Proceedings of the International AAAI Conference on Web and Social Media, 2009. **3**(1): p. 361-362.
41. Silla, C.N. and A.A. Freitas, A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery, 2011. **22**(1-2): p. 31-72.
42. Gao, D., et al., Deep Hierarchical Classification for Category Prediction in E-commerce System. arXiv:2005.06692 [cs], 2020.
43. Silla Jr, C.N. and A.A. Freitas. A Global-Model Naive Bayes Approach to the Hierarchical Prediction of Protein Functions. in 2009 Ninth IEEE International Conference on Data Mining. 2009.
44. An, G., et al., Hierarchical deep learning models using transfer learning for disease detection and classification based on small number of medical images. Scientific Reports, 2021. **11**(1): p. 4250.

Chapter 3: Mining the Biosynthetic Universe using Large Language Models.

3.1 Chapter Preface

It was a collaborative effort to revamp the entire genomic mining pipeline. Norman Spencer and Mathusan Gunabalasingam curated datasets of biosynthetic sequences analogous to the models found in PRISM and antiSMASH. I developed a framework capable of training on multiple datasets for multiple tasks. Using their datasets, I created a ProtBERT model capable of annotating peptide sequences with biosynthetic domains and protein families. Mathusan Gunabalasingam replicated my work but also included an enzyme commission number dataset. Norman Spencer and Mathusan Gunabalasingam developed cut-offs for the annotation library to increase accuracy.

In the past, we have tried to use transformers for adenylation domain substrate prediction but with little success. I developed a new strategy rooted in multi-task learning. I worked with Victor Blaga, Norman Spencer and Mathusan Gunabalasingam to create datasets for adenylation and acyltransferase substrates with additional features to facilitate a multi-task approach. Mathusan Gunabalasingam also hand-curated the functional and non-functional polyketide domain datasets. Using my custom deep learning framework, I trained models capable of predicting the substrates and functionalities with high degrees

of success. Mathusan Gunabalasingam created the look-up approach with Milvus to minimise classification time and increase accuracy.

For gene cluster comparison, I developed a feature-based approach using the enumerated feature sets from InterPro and calculated distances using the MinHash. Norman Spencer built off this work, creating a more biosynthetic-centric approach to featurization using information from all of the models along with additional calculated features such as module numbers. I also developed the vector-based approach for representing BGCs. To validate the gene cluster families, Mathusan Gunabalasingam found the matching metabolites within the mass spectral data. While much of this platform is based on my technical innovations, its refinement and success would not have been possible without the joint efforts of Norman Spencer and Mathusan Gunabalasingam.

3.2 Abstract

Genomic mining pipelines, such as AntiSMASH and PRISM, have been extremely successful at detecting encoded metabolites in microbial genomes. They have created an explosion of data with IMG-ABC now containing over 400,000 biosynthetic gene clusters (BGCs). BLAST and profile Hidden Markov Models are the core technologies of BGC mining and comparison. Unfortunately, the current paradigm for using these tools does not scale, with running times now ranging from hours to days. In

this work, we present an alternative deep learning-based pipeline to microbial genomic mining. An integrated biosynthetic informatics suite (IBIS) of large language models is used to predict encoded chemistry, discover gene cluster families, and map out the microbial biosynthetic space. Our pipeline is optimised for scalability and boasts a nearly 8x speed gain over conventional methods.

3.3 Introduction

Microbial metabolites are a rich source of medically relevant molecules. Microbial secondary metabolites are encoded in genomic islands called biosynthetic gene clusters (BGCs). The genes found in the BGCs can be involved in the synthesis, regulation and resistance of the encoded molecule. [1, 2] The spatial proximity of the genes with shared functionality provides a unique case to facilitate the automated detection of BGCs. PRISM and AntiSMASH are two genomic mining pipelines developed for the rapid discovery of BGCs in a bacterial genome.[3, 4] Both software use profile Hidden Markov Models (pHMMs) for processing protein sequences. Each pHMM is created using a sequence alignment of a conserved protein family. By encoding sequence conservation as a Markov process and scoring amino acid transition using a sequence alignment, a single HMM can be used to find an enzymatic domain on a per residue level of resolution.[5]

With every newly discovered BGC, new pHMMs are needed to capture the novel enzymology. When new pHMMs are then added to the pipeline, the running time of the genomic mining pipelines is further extended. While the first version of PRISM contained 479 HMMs, PRISM 4 now boasts a library of 1772 pHMMs; antiSMASH 6.0 has 354 pHMMs. In both pipelines, all pHMMs must be run against each peptide sequence individually. PRISM's average running time is now 58.8min while a de-novo AntiSMASH run takes several hours.[6, 7] As the biosynthetic enzymatic space continues to grow, the HMM-based pipelines will become increasingly slower. The current paradigm is not scalable.

Genomic mining technology has enabled the rapid discovery of BGCs. Advances in high-throughput sequencing have procured a wealth of publicly available bacterial genomes. The Integrated Microbial Genomes' Atlas of Biosynthetic Gene Clusters (IMG-ABC) now reports 411,475. [8] Our internal database boasts 649,291 BGCs. Rapidly comparing and classifying the libraries of gene clusters has also become a scaling problem. BiG-SCAPE was developed to generate gene cluster families (GCFs) but it is only scalable to tens of thousands of BGCs. [9, 10] BiG-SLICE was developed to take a vector-based approach to represent BGCs, but it again relies on a library of pHMMs for featurization. Annotating the BGCs with the pHMMs makes up over 90% of the running time.[10] As BGC libraries continue to grow, more efficient methods for organising and traversing the biosynthetic space are needed.

The advent of transformers facilitates a new approach to biosynthetic peptide annotation and GCF elucidation. Common natural language processing (NLP) tasks such as Named Entity Recognition and Sentence Classification based on sentiment are analogous to the residue-based domain annotation and whole protein classification tasks BGC mining tools perform.[11] With a single transformer trained to replace the library of HMMs, each peptide sequence would only need to be processed once. While transformer-based pipelines were once slow in running time, frameworks such as ONNX have been designed to speed up inference through quantisation and approximation.[12]

Beyond pHMMs, transformers can be used as drop-in replacements for sequence database querying techniques such as BLAST.[13] BLAST is commonly used to find highly related peptide sequences in a database by looking for sequence conservation. Unfortunately, BLAST is often the most computationally intensive part of genomic mining pipelines. Slight amino acid changes, as are often the case with adenylation domains and polyketide domains, can drastically increase running times. [14] As an alternative to sequence alignment-based strategies such as BLAST, embeddings from transformers can be used. Transformers trained on protein sequences have demonstrated the ability to predict structural class, protein function and source domain of life using their latent space. [15] Vector databases, such as Milvus, have been created to enable vector searches on the scale of billions. Through the quantisation of the one-dimensional

vectors, embeddings can be indexed and readily queried for nearest neighbours. [16, 17] By utilising the vector embeddings of peptide sequence instead of the sequence alignment, rapid high-resolution querying can be achieved.

In this work, we present a scalable pipeline to genomic mining using IBIS - An Integrated Biosynthetic Informatics Suite of large language models (Figure 3.1). The IBIS of LLMs currently consists of three main components: (1) AdenylationT5 and AcyltransferaseT5 - two transformers trained for the substrate property prediction of binding pockets, (2) PK Domain T5 - five transformers trained to determine whether or not polyketide domains are functional and (3) EnzymeBERT - A high-speed transformer infused with biochemical knowledge and trained for the classification of protein families and domains. We demonstrate basic structure prediction using peptide annotations and unsupervised GCF calling using topic modelling techniques.[18]

3.4 Methodology

3.4.1 Adenylation and Acyltransferase T5

In current gene cluster mining tools, substrate prediction is performed using sequence alignment-based techniques. Small changes in the residues of adenylation and acyltransferase domains will result in different binding substrates.[19, 20] PRISM uses BLASTP and prediCAT uses MAFFT-generated trees to the nearest neighbour based on

the residues.[4, 21] While both methods are effective, alignment-based techniques are not scalable. To address these issues IBIS introduces a transformer-based approach to substrate prediction. Two separate LLMs were trained to predict the discrete substrate label as well as the potential chemical properties of predicted substrates.

3.4.1.1 Model Architecture

The publicly available peptide-based transformer, ProtT5, was able to discrete separate amino acids by biophysical features, proteins by structural class and proteins by their native kingdom of life.[15] Because of the demonstrated high resolution, the encoder side of the ProtT5 model was selected as the base LLM for substrate prediction. In order to maximise inference and learning, a multi-task training regimen was designed. Both models were trained to predict the substrate as well as a series of biochemical features. Custom sequence classification heads were created to fine-tune each model across separate tasks. The Adenylation T5 model was trained to predict the substrate family, the aromaticity, the number of hydrogen bond acceptors, the number of hydrogen bond donors, the partition coefficient (LogP), and the topological polar surface area (TPSA). The Acyltransferase T5 model was trained to predict the substate family, the CH2 count, the CH3 count, the number of hydrogen bond acceptors, the number of hydrogen bond donors, LogP, and TPSA.

3.4.1.2 Input/Outputs

The ProtT5 tokenizer was used to treat each protein sequence as a sentence and every amino acid as a word. For each task, a separate class token was appended to the sequence (7 class tokens for the Acyltransferase T5 and 6 class tokens for the Adenylation T5).

3.4.1.3 Multi-task Finetuning

A hand-curated dataset of adenylation and acyltransferase domains was prepared from Magarvey Laboratories' internal library of biosynthetic gene clusters with experimentally verified metabolites. There was a total of 2,283 adenylation domains with 83 different substrates, but only 34 of the substrates had classes with over 4 examples. The feature engineering process for the additional biochemical properties was performed using RDKit.[22] To convert the continuous features into discrete classes, the values were roughly grouped using the Jenks natural breaks optimisation and then manually tailored by expert chemists.[23] Using this process, every adenylation domain was given six separate class labels and the acyltransferase domains were given seven. The datasets were stratified by their substrate label and split into training, validation and testing using the ratio 64:16:20.

To perform multi-task learning, separate sequence classification heads were used for each task. Labels were passed to their corresponding classification heads and the loss was calculated using cross-entropy. To combine the losses, a random loss weighting was used; this algorithm randomly weights the losses between the different tasks before summing them together. The final weighted loss was passed to the DeepSpeed Stage 3 optimiser with parameter offloading.[24, 25] Due to memory constraints on the in-house GPUs, the model was trained for 30 epochs with batches of only a single sequence. The learning rate was automatically determined by PyTorch Lightning.[26] Both models were evaluated using the Sci-kit learn library against test and validation tests.[27] Performance in terms of accuracy, balanced accuracy, F1 score, Hamming Loss, Zero One Loss, Hinge Loss, Jaccard Score, and Mathew's Correlation Coefficient are reported in Section 3.4.1.

3.4.1.4 Optimisation and Accelerated Inference

Both models were optimised and exported for accelerated inference using ONNX.[12] The dense layers of the individual classification heads were exported separately to allow for full embedding output before being converted into individual label logits. Custom pipelines were built for inference with the ORT framework and Hugging Face's transformers package.[28, 29]

3.4.1.5 Post-Processing

Success on sequence classification tasks is limited to the number of representative samples available for a given class. 49 of the 83 substrates for the adenylation domain dataset were quite rare (< 4 sequences available). The rare substrates do not have a sufficient number of peptide sequences for classification training. In addition, when a new substrate is discovered, adding it to the already trained classification head would require retraining the model. To bypass these limitations, two alternative strategies for sequence classification were devised. Results from the standard classification head and two alternative strategies are reported in Section 3.4.2.

3.4.1.5.1 Vector Databases

Using the software Milvus, two vector databases were created; one for solved adenylation domains and another for solved acyltransferase domains.[17] Each database was comprised of the embedding class token for the substrate label; this is exported by the dense layer of the custom classification head. Both databases used a flat inverted file index, optimised for euclidean distance searches. Using the two vector databases, substrate labels were predicted through k-nearest neighbour (kNN) look-ups.[16]

3.4.1.5.2 Explainable Boosted Machine

The Explainable Boosted Machine (EBM) is a statistical model used for classification and interpretability at Microsoft Research.[30] It has shown similar performance to XGBoost across a variety of tasks. It also reveals the decision-making process when a prediction is made. An EBM model was trained as an ensemble model; it used the discrete features predicted using the six Adenylation T5 classification heads to predict the adenylation domain substrate.

3.4.2 PK Domain T5 Models

Polyketide domains are readily detected using pHMMs. Some of the enzymatic domains are non-functional due to minor residue changes in their sequence. The non-functional domains are a source of error when predicting the potential structure of a biosynthetic gene cluster. To address this issue, multiple ProtT5 models were fine-tuned to define whether or not polyketide domains were functional.

3.4.2.1 Model Architecture

There are 5 models for the Polyketide Domain T5 group: Dehydrotase (DH), EnoylReductase (ER), Ketosynthase (KS), Ketoreductase (KR), and Thiolation (T). Each model uses the ProtT5 encoder as its base with two sequence classification heads for two separate tasks: (1) Classifying whether or not the domain is functional, and (2) The evolutionary clade of the source microbe.

3.4.2.2 Input/Outputs

The ProtT5 tokenizer was used to treat each protein sequence as a sentence and every amino acid as a word. As before for each task, a separate class token was appended to the sequence (two class tokens).

3.4.2.3 Fine-Tuning

To train the 5 different PK Domain T5 models, separate datasets were hand-curated. For each PK domain model, peptide sequences were collected from literature sources and split into functional versus non-functional classes. Peptide sequences were also aligned and split into evolutionary clades as a secondary classification. Similar to the Adenylation and Acyltransferase T5 models, each of the PK Domain T5 models was trained using multi-task learning. The only exception was the thiolation domain model; it was not trained on clade classification. All losses were pooled using random loss weighting. Each model was trained across multiple GPUs using DeepSpeed's Stage 3 Optimiser with parameter offloading for a total of 15 epochs. [24, 31] Training results can be found in Section 3.5.2.1.

3.4.2.4 Accelerated Inference

The highest-performing models according to substrate prediction accuracy were exported for accelerated inference using ONNX. Only the substrate classification head was exported, as the evolutionary clade task was provided solely to assist in the learning process. Custom pipelines were built for inference with the ORT framework and Hugging Face's transformers package. [28, 29]

3.4.2.5 Post-Processing

Separate vector databases were developed for each PK Domain model. Each database is made of the embedded appended class token used for the functional classification task. Each database used a flat inverted file index, optimised for euclidean distance searches. Functional classification labels were assigned using the nearest neighbours. Results can be found in Section 3.5.2.2

3.4.3 Enzyme-BERT

EnzymeBERT is an LLM trained to comprehensively annotate a peptide sequence in a single pass. AntiSMASH and PRISM use pHMMs to annotate peptide sequences with biosynthetic enzyme families, biosynthetic domains and general genes of interest. Annotations include polyketide domains, NRPS domains, terpene synthases, regulators, resistance genes etc.[3, 4] The annotations can be separated into two groups: (1) peptide region annotations (domains) and (2) whole peptide sequence annotations (enzyme

families and genes of interest). EnzymeBERT is trained to perform both tasks using two separate strategies. The strategy for whole peptide sequence annotation was conceived as a vector database lookup of annotated peptide sequence embeddings. For peptide region annotations, a token classification strategy was developed.

3.4.3.1 Model Architecture

The base architecture selected was the pre-trained ProtBERT model released in 2020. It was pre-trained on UniRef100 and demonstrated high performance on fine-tuned protein classification tasks. [15] ProtBERT is also very fast to run upon quantisation and optimisation in comparison to ProtT5. A total of six classification heads were created to fine-tune the model's latent space using different annotation tasks. Four sequence classification heads were created for the prediction of different levels of Enzyme Commission (EC) numbers.[32] A separate sequence classification head was created for the prediction of protein families relevant to biosynthesis but not covered by an EC number (will be referred to as protein families). A single token classification head was created to predict the boundaries of different functional domains within a protein sequence such as dehydratase domains, adenylation domains etc. (will be referred to as biosynthetic domains).

3.4.3.2 Input/Outputs

The ProtBERT tokenizer was used to treat each protein sequence as a sentence and every amino acid as a word.

3.4.3.3 Fine-Tuning

Enzyme BERT was fine-tuned in two separate stages. The first stage imbued the base model with a fundamental understanding of enzymology using EC numbers. The second stage continued the training but introduced more nuanced tasks such as the identification of additional biosynthetic relevant protein families and the boundaries of biosynthetic domains. Results from the fine-tuning tasks are found in Section 3.5.3.1.

3.4.3.4.1 Task 1: Multi-Task Fine-tuning for EC# Classification (Four Tasks)

To infuse the ProtBERT model with biochemical knowledge, it was first trained on the prediction of EC numbers. A dataset of 316,601 protein sequences with EC numbers to the fourth level was curated. To balance the distribution of classes the dataset was split: 213,849 for training, 31,192 for validation, and 14,852 for testing. Using the scikit-learn package, individual class weights were calculated and used in the cross-entropy loss calculation.[27] Losses from each EC level's classification head were summed together and passed to a DeepSpeed Stage 3 Optimiser with parameter offloading. Precision was limited to 16 bits.[24] Gradients were accumulated after every 8 batches. Optimal

learning rates were automatically calculated using PyTorch Lightning.[26] The model was trained for a total of 100 epochs.

3.4.3.4.2 Task 2: Multi-Task Fine-tuning for Domain, Protein Family and EC# Classification (Six Tasks)

Two additional datasets were curated for the biosynthetic domain classification and protein family classification tasks. Profile Hidden Markov Models (pHMMs) were selected from the libraries of antiSMASH, PRISM, Pfam and InterPro on the basis of biosynthetic relevance.[3, 7, 33, 34] UniProt was mined using the pHMMs to create a dataset of annotated peptide sequences.[35] The annotations were condensed into non-redundant families based on overlapping peptide sequence matches. After merging and quality filtering, a total of 1,101 protein families and 123 biosynthetic domains were selected. Peptide sequences with biosynthetic domain annotations were converted into a dataset with a list of token-level labels (e.g. every amino acid was labelled with a corresponding domain). Peptide sequences with protein family annotations were converted into a dataset with a single sequence-level label (e.g. the entire peptide sequence was labelled with its corresponding family).

EnzymeBERT was fine-tuned using multi-task learning across six tasks: (1-4) a sequence classification task to predict the EC number classification at each level, (5) a sequence

classification task to predict the protein family and (6) a token classification task to predict the regions of a functional domain. The three datasets (EC number, protein family and function domain) were pooled together and each peptide sequence was assigned six labels for each of the tasks. When a sequence was missing its curated label for a given task, an "UNKNOWN" label was imputed. The total dataset was comprised of 603,538 protein sequences: 492,337 for training, 65,555 for testing, and 45,646 for validation. During the multi-task training, each label was passed to its designated classification head. Using the scikit-learn package, individual class weights were calculated and used in the cross-entropy loss calculation.[27] Cross entropy losses were summed across the heads and passed to a DeepSpeed Stage 3 optimiser with parameter offloading.[24, 25]

3.4.3.4 Quantisation, Optimisation and Acceleration

EnzymeBERT and the token classification head were quantised and optimised using ONNX.[12] Custom pipelines were built for inference with the ORT framework and HF's transformers package. [28, 29] The running time was calculated on 100 random batches of 32 ORFs.

3.4.3.6 Post-Processing

3.4.3.6.1 Vector Databases

Using the software Milvus, two vector databases were created (1) EC numbers and (2) Protein families.[17] Each database was comprised of the "[CLS]" token's embedding for every reference peptide sequence within the two datasets. Both databases used a flat inverted file index, optimised for euclidean distance searches. The corresponding EC numbers and protein family classifications were assigned using k-nearest neighbour (kNN) look-ups with the two vector databases. [16]

3.4.3.6.2 Token Classification Grouping

The token classification head annotates every amino acid with a separate label and probability. To merge token labels into discrete regions across the peptide sequence, a separate polishing algorithm was written. We represent every peptide sequence as a linear connected graph of amino acids. Every amino acid is represented as a node and is initially connected to neighbouring amino acids via a weighted edge of 1.0. Additional edges are drawn for surrounding residues within a 10-residue radius if they share the same token label; these edges are weighted with the common label's probability. The peptide sequence network graph is then subdivided into communities using the Louvain method. [36] A majority vote is taken to label the community of residues with a single label.

3.4.3.6.3 Gene Cluster Calling

The protein family and functional domain annotations provided by the EnzymeBERT model can be used to predict the boundaries of a BGC. For each BGC chemotype, a list of relevant annotations was curated. Using a greedy grouping algorithm, peptides were grouped together based on two conditions: (1) if they were within 10,000 base pairs upstream or downstream of one another and (2) shared peptide annotations within the same curated chemotype list. The 10,000 base pair cut-off was based on literature.[37] The start and stop locations for the final groups of peptide sequences were referred to as putative gene clusters.

3.2.3.6.4 Molecular Unit Prediction

Enzyme commission numbers can be directly connected to chemical reactions through a variety of different public databases. Leveraging genomic annotations to deduce the reaction space, molecular units of a biosynthetic gene cluster can be predicted. To develop a framework capable of this, all primary and secondary metabolism reactions from Kyoto Encyclopaedia of Genes and Genomes (KEGG), BRENDA, and MetaCyc were scraped. [38-40] In addition to publicly available reactions, hundreds of rare biosynthetic pathways were hand-curated. All reactions were converted to a Reaction SMILES format.[41] The annotations predicted by EnzymeBERT were linked to the reaction library.

Using EnzymeBERT, a biosynthetic gene cluster can be annotated with its entire reaction space. A separate framework called BEAR (manuscript in progress) can use the

reactions to perform in-silico biosynthesis. For some reactions, the substrates used were customised based on the results from the Adenylation and Acyltransferase T5 models. Results from the PK Domain T5 models were also used to indicate whether or to include the reaction space of a putative functional polyketide domain. Using the combination of BEAR and IBIS, the molecular unit space of an annotated BGC can be predicted. An example of erythromycin's predicted linear polyketide chain can be seen in Figure 3.2.

3.4.3.6 Experiments

3.4.3.6.1 Attention Map of Maltose Acetyltransferase Domain Sequences

The dataset for the domain-level annotations contained highly similar sequences. There was a concern the transformer may determine token classifications by globally attending to the entire sequence rather than locally attending to individual residues of interest; the former being a characteristic of whole sequence memorisation rather than learning the discrete patterns in enzymology. To address this concern, an attention map was generated of EnzymeBERT processing a dataset of acyltransferase domains.

To compute an attention map, attention weight distributions must be output from the transformer while it embeds a peptide sequence. The weight distribution is a multi-dimensional tensor. Weights are generated for every layer in the model (16 layers). Within each layer, every attention head will generate a separate weight distribution (12 heads per

layer). Each attention head will output a separate weight distribution for the individual amino acids in the input sequence (n represents the number of amino acids in the sequence). Each weight distribution corresponds to the contribution surrounding residues had to the final amino acid's embedding. The attention weight tensor is therefore a shape of (16, 12, n, n).

As a case study, a dataset of maltose acetyltransferase sequences was extracted from the InterPro repository for the HMM PF12464.[33] A total of 535 sequences were processed by the EnzymeBERT model. The attention head weight distribution was saved for each sequence. To represent the average attention across all sequences, attention tensors were averaged together using the InterPro global alignment as a guide. The combined EnzymeBERT attention maps (one per global alignment index) were plotted as heat maps. Each individual heat map showed the attention distribution across surrounding residues for a given global alignment position. The global alignment was plotted as a sequence logo using the python package "logomaker".[42] Results are shown in Section 3.5.3.2.

3.4.3.6.2 Running Time of EnzymeBERT with ONNX

To calculate the average running time of EnzymeBERT on an input peptide sequence, 100 batches of 32 peptide sequences randomly sampled from microbial genomes were run

using the ONNXRuntime framework on an NVIDIA Quadro RTX 5000. Results are reported in Section 3.5.3.3.

3.4.3.6.3 Preservation of chemical relationships within BGC comparisons

Two separate strategies were developed for comparing BGCs: (1) MinHash Indexing using hand-curated features derived from the different LLMs and (2) Vector Indexing using combined open reading frame embeddings. Using a dataset of BGCs with experimentally verified metabolites, gene cluster distances/dissimilarities were validated using a triplet relationship strategy. Triplets consist of an anchor (query BGC), a positive example (BGC with a highly related metabolite) and a negative example (BGC with a highly unrelated metabolite). The two chemical metrics used to deduce positive and negative BGCs were: (1) non-folded, enumerated FCFP6 with the Jaccard Index and (2) the Euclidean distance of NP-BERT embeddings. For the FCFP6 metric, negatives were calculated with a dissimilarity greater than 0.7 and positives with a dissimilarity less than 0.2; this generated a total of 1,504,290 triplets for a total of 441 unique anchors. For the NP-BERT metric, negatives were calculated with a distance greater than 50 and positives were calculated with a distance less than 10; this generated a total of 475,851 triplets for a total of 441 unique anchors. Each triplet was categorised by the NP-BERT predicted super class of the anchor's linked metabolite. In addition, Spearman correlations were

performed between both BGC approaches against both chemical comparison approaches. The performance of both strategies with both metrics is reported in Section 3.5.3.4.

3.4.3.6.3.1 MinHash Indexing of Biosynthetic Annotation Sets

MinHashes are data sketches of sets. To represent a biosynthetic gene cluster as an unordered set, features were engineered using the annotations from the IBIS of LLMs. Information regarding individual domains, enzymes, EC numbers, predicted substrates, and modules were all integrated into the sets. Duplicated information was captured using enumeration. To calculate metrics, the dissimilarities between the unordered sets were calculated using the Jaccard Index. In production, the unordered sets were converted to MinHash data sketches with 128 permutations and the xxhash hashing algorithm.[43] Nearest neighbours are found using the MinHashLSH implementation in the python library “datasketch".[44]

3.4.3.6.3.2 FAISS Indexing of Pooled Biosynthetic Embeddings

To represent a biosynthetic gene cluster as a single vector only EnzymeBERT was used. All peptide sequences within the BGC were embedded by EnzymeBERT and the embeddings were then averaged together to a single vector. Metrics were calculated using true Euclidean distances between averaged embeddings. In production, approximated

Euclidean distances are calculated using a flat L2 index created with FAISS. Nearest neighbours are calculated using the FAISS built-in index searching.[16]

3.4.3.6.4 BGC Universe

An internal dataset of over 649,291 BGCs was pruned for redundancy using 100% ORF identity. A total of 296,216 unique BGCs remained for visualisation. To represent BGCs as vectors for visualisation, the pooled biosynthetic embedding strategy was used. First, every ORF was embedded using EnzymeBERT. The embedded class tokens of every ORF within a BGC were averaged together to create a single embedding representative of the BGC. The BGC embeddings were projected to two dimensions using the RAPIDS GPU implementation of UMAP.[45] The projected embedding was validated using the trustworthiness metric proposed by Venna and Kasuki. [46] Gene clusters were plotted using UMAP's plotter tool and data shader.[47, 48] The IBIS-predicted chemotypes were used for colour selection. The BGC Universe visualisation can be found in Section 3.5.3.6.

3.4.3.6.5 Programmatic Deduction of Gene Cluster Families

Using topic modelling techniques similar to BERT-Topic, GCFs can be programmatically deduced.[18] To perform this analysis, the BGC embeddings were projected from 768 dimensions to 128 dimensions using RAPIDS UMAP.[45, 47] The

projected embedding was validated using the trustworthiness metric proposed by Venna and Kasuki. [46] Using RAPIDS GPU implementation of HDBSCAN, the dataset of 296,216 BGCs was clustered into putative GCFs.[49] Clustering was optimised using the silhouette score.[50]

GCFs related to experimentally verified metabolites were selected for further exploration. BGCs within the same GCF were plotted as network graphs using Gephi.[51] Each node represented a BGC; edges in the graph were drawn between all nodes with edge weights varying based on shared taxonomy (i.e. a BGC belonging to Streptomyces had heavier edges drawn between other Streptomyces BGCs versus BGCs from other genera). Molecules were confidently mass matched to microbial fermentation extracts using the in-house tool MAPLE (MetAbolomics Peaks Logic Engine).

3.5 Results

3.5.1 Adenylation and Acyltransferase T5

3.5.1.1 Adenylation T5 and Acyltransferase T5 Fine Tuning

For the acyltransferase T5 model, all properties were predicted with over 90% accuracy. For the Adenylation T5 model, all properties were predicted with over 80% accuracy, with the exceptions of LogP and the substrate. For both models performance

between validation and test datasets was similar. All metrics are reported in Table 3.1 and Table 3.2.

3.5.1.2 Adenylation T5 Substrate Prediction

The substrate classification head performed the worst of all three methods. The EBM method performed slightly better than the classification head. The FAISS nearest neighbour technique was the most accurate overall. The EBM technique achieved greater accuracy than the nearest neighbour lookup for a select few substrates (Pip, Piz, Hpg, Glu, Dbut, Dab, Arg and Sal). The accuracy of the different techniques for substrate prediction is summarised in Figure 3.3.

3.5.2 Polyketide Domain T5 Models

3.5.2.1 Polyketide Domain T5 Fine-tuning

Validation accuracies and losses across the different Polyketide Domain Models are shown in Figure 3.4. With the exception of the ketosynthase model, the T5 models were able to correctly predict whether or not a domain was functional or non-functional with over 90% accuracy. Clade prediction accuracy reached over 80% for enoylreductase and dehydratase domains. Models began to overfit quite quickly (at approximately 10 epochs).

3.5.2.2 Polyketide Domain Nearest Neighbour Lookup Performance

When each model's functional prediction was swapped to the nearest neighbour lookup protocol, ketosynthase functional prediction performance increased with an F1 Score reaching 0.71. All results are summarised in Table 3.3.

3.5.3 Enzyme BERT

3.5.3.1 EC# Prediction, Protein Family, and Enzymatic Domain Classification Metrics

Across all classification tasks (the four levels of EC numbers, protein family, and protein domain), EnzymeBERT was able to achieve F1 scores above 0.9. All classification metrics can be found in Table 3.4.

3.5.3.2 Attention Map of Token Classification Head

As EnzymeBERT embeds individual tokens of an input peptide sequence, its attention heads display local attention for each individual residue. The residues most attended to are those immediately surrounding the token of interest. Figure 3.5 displays the moving local attention window. The gapped global alignment of the motif's sequences resulted in local attention windows with an exaggerated width.

3.5.3.3 Running Time of EnzymeBERT with ONNX

Without any polishing steps, the average processing time of EnzymeBERT per ORF on the NVIDIA Quadro RTX 5000 was 0.099s. The average genome contains approximately 5000 proteins; extrapolating from this number the average running time should only take 8.25 minutes. Using the largest (*Sorangium cellulosum* strain So0157-2: 14,782,125 bp with 11,599 genes) and smallest genomes (*Candidatus Nasuia deltocephalinicola* strain NAS-ALF: 112,091 bp with 137 proteins) available on the NCBI as a reference, the estimated running time for processing a genome is between 20 minutes and 13.6 seconds

3.5.3.4 Gene Cluster Comparison with Vector Embeddings versus Biosynthetic Features

Triplet accuracies for both the vector-based approach and the feature-engineered sets approach to BGC comparison are shown in Figure 3.6. For the FCFP6 dissimilarity triplets, the feature-engineered sets approach outperformed the vector embeddings approach across every category with the exception of fatty acyls. The vector embeddings approach while not superior still performed well with the majority of categories averaging over 90% accuracy. For the NP-BERT distance triplets, the feature engineering approach outperformed the vector embeddings approach across every category with the exception

of fatty acyl and naphthalenes. The vector approach also performed well with the majority of categories still achieving an accuracy of over 90%, but there were more categories with poor performance (~40-55%) including phenolic acids, Beta lactams, fatty acids and conjugates. The feature-engineered approach's dissimilarities had Spearman correlation scores of 0.203 and 0.253 with the FCFP6 dissimilarity and NP-BERT distance respectively (p-values were equal to 0). The vector-based approach's distances had Spearman correlation scores of 0.065 and 0.202 with the FCFP6 dissimilarity and NP-BERT distance respectively (p-values were equal to 0). Joint histogram plots for all four comparisons can be found in Supplementary Figure S3.1.

3.5.3.6 BGC Universe Visualisation

The plotted UMAP projected biosynthetic gene clusters are shown in Figure 3.7. While there was no segmentation of chemotypes on a macro level, individual clusters of chemotype conservation were observed throughout the embedded space. A large overlap between Non-Ribosomal Peptides and Polyketides, as well as their hybrids, was observed. The trustworthiness of the 2D UMAP embedding was calculated as 0.999.

3.5.3.6 BGC Derivative Families

Of the dataset of 296,216 BGCs, only 213,007 were assigned to 12,495 GCFs using HDBSCAN; the remainder were unlabelled due to the strict clustering cut-off of 10

members. Three of the GCFs were selected for further exploration. In Figure 3.8, the group of Type II polyketide gene clusters resembling Curamycin A is shown. All BGCs were located in *Streptomyces* genomes, similar to the original producer of Curamycin A, *Streptomyces cyaneus*. Mass matching of metabolites within the crude extract from *Streptomyces viridosporus* DSM 40243 confirmed Curamycin A was produced. In Figure 3.9, the group of Type I polyketide gene clusters resembling Erythromycin is shown. Many of the gene clusters were mined from genomes of the Streptomyces and the original producer *Saccharopolyspora*. The exceptions were *Aeromicrobium erythreum*, *Micromonospora rosaria* and *Saccharomonospora paurometabolica*. A*eromicrobium erythreum* historically produces erythromycin. *Micromospora rosaria* historically produces rosaramicin, a similar molecule to erythromycin. In Figure 3.10, a group of Non-Ribosomal Peptide gene clusters resembling Polymyxin B is shown. Gene clusters of this group were mined exclusively from *Paenibacillus* and *Brevibacillus*. Mass matching of metabolites within the crude extract from *Paenibacillus alvei* DSM 29 confirmed Polymyxin B was produced.

3.6 Discussion

3.6.1 Improving Scalability of Genomic Mining Using Transformers

We have demonstrated a single fine-tuned transformer can effectively replace the thousands of pHMMs used in genomic mining. With EnzymeBERT we achieved an F1-

score over 0.9 across enzyme commission number prediction, protein family prediction and protein domain prediction tasks. The running time of EnzymeBERT on the average genome is estimated at 7.5 minutes; in comparison to PRISM's 58.8 minutes and antiSMASH's multiple hours, the use of EnzymeBERT will result in an approximately eightfold speed gain over conventional pipelines.[6, 7]

When a BGC's newly discovered enzymology is to be modelled as a pHMM it is a very time-consuming and nuanced process.[52] First, a sufficient amount of sequences must be gathered to create a sequence alignment capable of capturing the protein family's profile. After the pHMM is created, the cut-off score must be optimised as to not interfere with other related pHMM annotations in the library. This process has been simplified in the IBIS pipeline in two ways: (1) taking advantage of the transformer's high-resolution latent space and (2) moving classification from discrete classifiers to a semantic search.

By leveraging EnzymeBERT's latent space, the total number of representative sequences is more relaxed. When a peptide sequence is embedded by EnzymeBERT, its vector is imbued meaningfully with all of the biases trained into the transformer through the different fine-tuning tasks. The bias acquired through the fine-tuning process is retained in the transformer's memory; this effectively replaces the bias calculated through sequence alignment for a pHMM. The latent space of EnzymeBERT contains enough information that underpopulated classes of peptides are still meaningfully represented

together; this was demonstrated with the precision and recall of the Enzyme Commission number's predictions at the third and fourth levels. While the representatives at these levels are sparse, the F1 scores at the third and fourth levels are still above 0.98 and 0.96 respectively. The performance here is something of note as the state-of-art tool, DeepEC, demonstrates an overall F1 Score of 0.609. [53]

We uncoupled the classification label assignment from the classification heads; instead, classification is rooted in semantic search.[54, 55] Classification heads contain a finite number of labels limited to those in the training data; when using a vector database search, the labels are only limited to the vector database's reference sequences. We cannot easily add more classes to a classification head without retraining, but we can easily add more references to the vector database as new enzymology is discovered. The semantic search-like process is only possible because of the transformer's high-resolution latent space. Genomic mining can become more scalable with the use of transformers, not only through the reduced running time but also with simplicity in modelling biosynthetic enzymology into the system.

3.6.2 Adenylation Domain Substrate Prediction with Transformers

We also demonstrated that alignment-based strategies are not needed to achieve accurate adenylation domain predictions (Figure 3.3). While the classification head alone

had poor performance, the nearest neighbour strategy had over 80% accuracy for the majority of substrates. Semantic search bypasses the typical bottlenecks met when training conventional classifiers. Developing a comprehensive dataset with the minimum number of sequences to train a classification head is a difficult task. This issue was especially evident with rarer substrates such as piperazic acid.[56] Our classification head failed completely for piperazic acid adenylation domains (0% accuracy), but the semantic search was able to achieve an accuracy of 70%.

While there are over 500 known substrates for non-ribosomal peptide synthases, only 83 had adenylation domain sequences available.[21, 57] Semantic search is only applicable in substrate prediction while there are sequences available for the substrate of interest. To help identify matches for the remainder of substrates, the Adenylation T5 model also predicts substrate properties in addition to the substrate. The majority of properties are predicted with over 85% accuracy (Table 3.1). The predicted properties can be used as a signature for identifying putative NRPS substrates without reference sequences.

We attempted to blend predicted properties with classification head substrate predictions using an EBM model. The EBM model predicted Piperazic acid substrates with 100% accuracy. The model also achieved 100% accuracy across an additional 7 substrates. Unfortunately, the EBM classifier failed on a substantial number of other

substrates. While the EBM model did not achieve the same level of performance as the semantic search, it showcased the additional inference predicted properties can bring. Ideally, an ensemble approach will be taken in the future, where the substrates are first determined through semantic search and non-confident matches are corrected with the predicted properties using the EBM.

3.6.3 Transformer Attention in Enzymology

Profile HMMs are clearly defined by a Markov process scored with a multi-sequence alignment but transformers are currently not completely understood.[58] Most insight into the decision-making of a transformer is derived through attention-head weight outputs and attention-head masking. Profile HMMs are effective at recognising conserved motifs within protein sequences because of the sequential nature of a Markov process. [5] Transformers are not limited to neighbouring residues when learning from a sequence; through the use of global self-attention, transformers can weigh the influence of any token within the input sequence.[11]

When modelling an entire sequence, the ability to look at the entire input is very useful. The long-term memory of LSTMs facilitates similar performance gains.[59] Unfortunately, large deep learning models tend to overfit; there was a concern that the BERT model may memorise entire sequence patterns when predicting the domain labels

of individual tokens (e.g. predicting a thiolation domain because a condensation domain was seen). Ideally, during token classification, the transformer's attention heads would locally focus on adjacent residues similar to an HMM.

All concerns of memorisation were mitigated after plotting the averaged attention map of the Acetyltransferase domain (PF12464) in Figure 3.6. The figure showed attention weights shifting depending on the residue of interest being classified. The small attention windows demonstrated local attention rather than global. The pattern also suggests the architecture of the underlying transformer (BERT) can be swapped to a more memory-efficient model with minimal impact on performance; these include models using local windowed attention or sparse attention such as the Longformer or BigBird. [60-62] Currently our pipeline has to split megasynthases into windows before processing with EnzymeBERT due to memory constraints; the usage of optimised attention models would allow for larger enzymes to be processed at once, further decreasing our overall runtime.

3.6.3 Gene Cluster Comparison with IBIS

Rapidly comparing massive libraries of biosynthetic gene clusters is an active area of research. MultiGeneBLAST uses BLAST to map incoming peptides to those in known biosynthetic enzymes. [63] ClusterScout, BiG-SCAPE, BiG-SLICE use HMM libraries to

featurize biosynthetic gene clusters before applying various distance/dissimilarity metrics for comparison. [9, 64, 65] With transformers, a feature-based comparison is no longer required. In natural language processing, it is typical to average word embeddings together to generate a sentence embedding in an unsupervised manner.[66-69] In the same way, we can use peptide embeddings averaged together to generate a BGC embedding.

To compare the effectiveness of the unsupervised approach with a feature-based approach, we developed an in-house featurization algorithm based on the annotations predicted from the IBIS models. Features included predicted enzyme identities, substrates, and polyketide domain functionality. In addition, the protein domain annotations of thiotemplated gene clusters were further featurized to include explicit module information. With all of this information, the featurized comparison of BGCs was able to capture chemical relationships almost perfectly across all gene clusters with experimentally verified metabolites (Figure 3.6). We saw dips in performance for certain families including napthelenes and fatty acyls. Upon further investigation, gene clusters with irregular scores contained many hypothetic peptides with unannotated functionalities such as cercosporin, or contained exclusively general features found in many gene clusters such as 1-heptadecene; this exemplifies the inherent weakness of feature-based comparisons of BGCs. If no distinguishing features are modelled into the dataset, the comparison will fail. Even with extensive curation work, the comprehensiveness of the feature set will be biased toward gene clusters with experimentally verified metabolites.

The unsupervised method of gene cluster comparison achieves a similar level of success in capturing FCFP6-based relationships. It also manages to outperform the feature-based approach on the fatty acyls superclass. The EnzymeBERT embeddings inherently have resolving power, even if not explicitly declared. When vector comparisons were measured against the more stringent NP-BERT relationships, the unsupervised approach demonstrated more weaknesses. There were multiple potential sources of failure upon inspection. Some of the gene clusters were very large (>100,000 bp) and others spanned multiple contigs. The inclusion of peptides irrelevant to biosynthesis is more likely in BGCs of this nature. Another source failure was the lack of acyltransferase and adenylation domain resolution. As adenylation and acyltransferase domains with different substrates are not explicitly distinguished during fine-tuning, their embeddings will be generalised to their parent class; this can be corrected through an additional fine-tuning process.

We used the averaged vector embeddings to plot out the entire biosynthetic latent space using an internal dataset of over 200,000 non-redundant BGCs. While on a macro level, there was no distinct separation of BGCs, the majority of the BGCs were separated into small homogenous clusters with a conserved chemotype. We demonstrated that three of the GCFs discovered through HDBSCAN shared enzymology and encoded chemical similarity (Figure 3.8, Figure 3.9, Figure 3.10). We also showed there is metabolomic data

to support the predicted relatedness of the encoded molecules. While the unsupervised approach to gene cluster prediction is not perfect, it is still able to capture the chemical similarity of encoded metabolites with a high degree of accuracy.

3.6.4 Future Work

The IBIS pipeline demonstrates the optimisations transformers can bring to the field of genomic mining. The high resolution of a transformer's latent space can be utilised to replace pHMMs and BLAST in protein sequence homology workflows. With model quantisation and GPU-based inference, running times can be drastically reduced in comparison to conventional pipelines. Beyond improving the computational scalability of genomic pipelines, we demonstrate a simple method for adding new enzymology to the pipeline using semantic search. The plasticity of the transformer’s latent space places a great deal of importance on the training curriculum. In future work, we hope to bring the latent spaces of NPBERT and EnzymeBERT closer together with multi-domain refinement. A CLIP-like training regimen would further refine encoded chemistry comparisons.[70]

3.7 Figures and Tables

Figure 3.1: The entire IBIS pipeline is broken into 8 steps: (1) Open Reading Frame (ORF) Prediction - Pyrodigal is used to find all likely protein sequences per contig (2) Domain Annotation - Individual protein sequences are annotated with domain-level annotations through use of EnzymeBERT, a token classification head and two graph-based algorithms: Community-Based Polishing and Greedy Grouping (3) Whole Protein Annotation - Each protein sequence is annotated with a known biosynthetic functionality in the form of EC# or specific gene class (e.g. *vanA*) (4) Proximity-based BGC Boundary Calling - Based on the biosynthetic families associated with the annotations, the proteins are merged into BGCs using greedy grouping and a rule set. (5) Domain Substrate and Functionality Prediction - Any adenylation, acyltransferase or polyketide domains are further processed by the other T5 models for substrate prediction and determining whether or not the domain is functional. (6) Molecule Unit Prediction - Using BEAR molecular units can be predicted from the peptide annotations within each putative BGC.

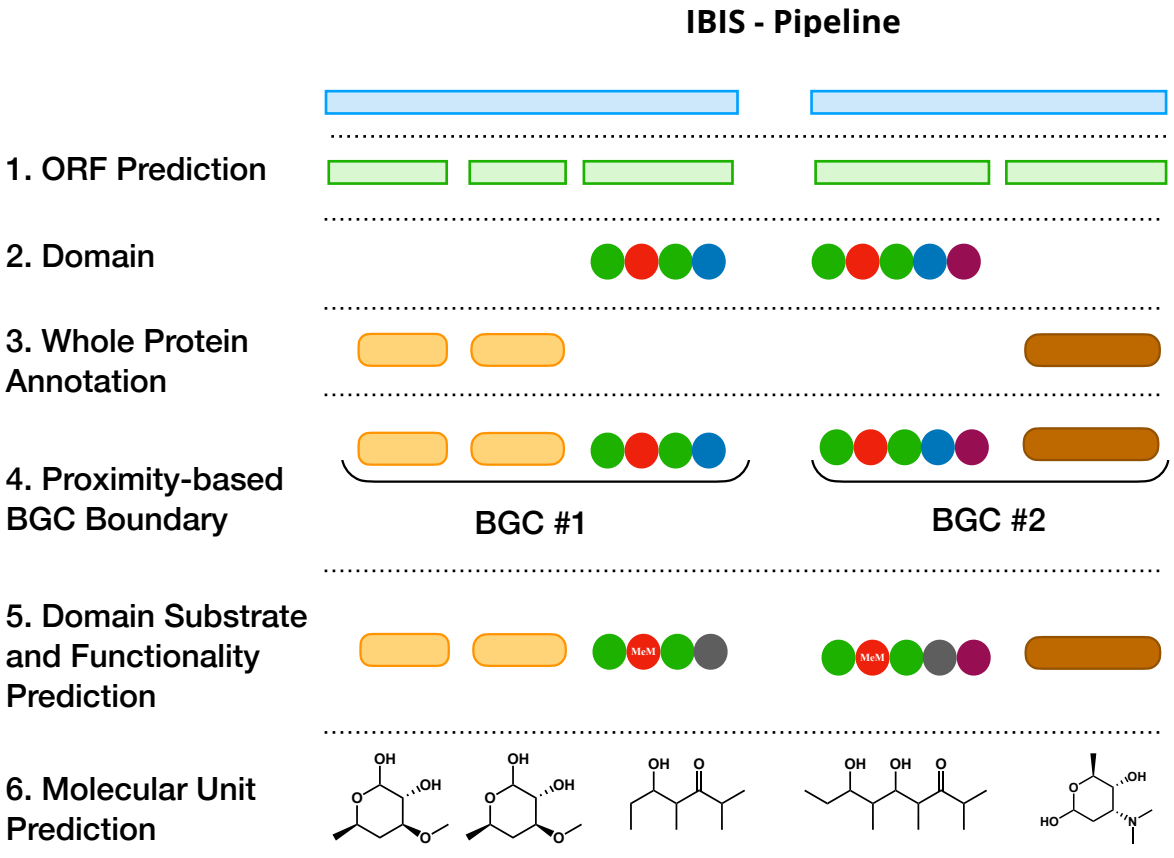


Figure 3.2 Example of Structure Prediction using BEAR and IBIS. The rich library of annotations provided by the IBIS pipeline facilitates the prediction of the molecular units produced by a biosynthetic gene cluster. The visualisation was created using the Natural Product Toolkit web application.

	Aromatic		Hydrogen Bond Acceptors		Hydrogen Bond Donors		LogP	TPSA			Substrate	
	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test
Accuracy Score	0.955	0.918	0.912	0.916	0.887	0.887	0.785	0.744	0.850	0.850	0.773	0.719
Balanced Accuracy Score	0.777	0.656	0.832	0.876	0.758	0.634	0.759	0.725	0.780	0.739	0.653	0.561
F1 Score	0.955	0.918	0.912	0.916	0.887	0.887	0.785	0.744	0.850	0.850	0.773	0.719
Hamming Loss	0.045	0.082	0.088	0.084	0.113	0.113	0.215	0.256	0.150	0.150	0.773	0.719
Zero One Loss	0.045	0.082	0.088	0.084	0.113	0.113	0.215	0.256	0.150	0.150	0.227	0.281
Hinge Loss	1.170	1.129	2.476	2.474	2.558	2.535	8.773	8.669	3.263	3.227	59.671	60.993
Jaccard Score	0.913	0.849	0.839	0.845	0.796	0.796	0.646	0.592	0.739	0.740	0.630	0.561
Matthew's Correlation Coefficient	0.852	0.721	0.828	0.836	0.785	0.782	0.763	0.718	0.792	0.791	0.761	0.702

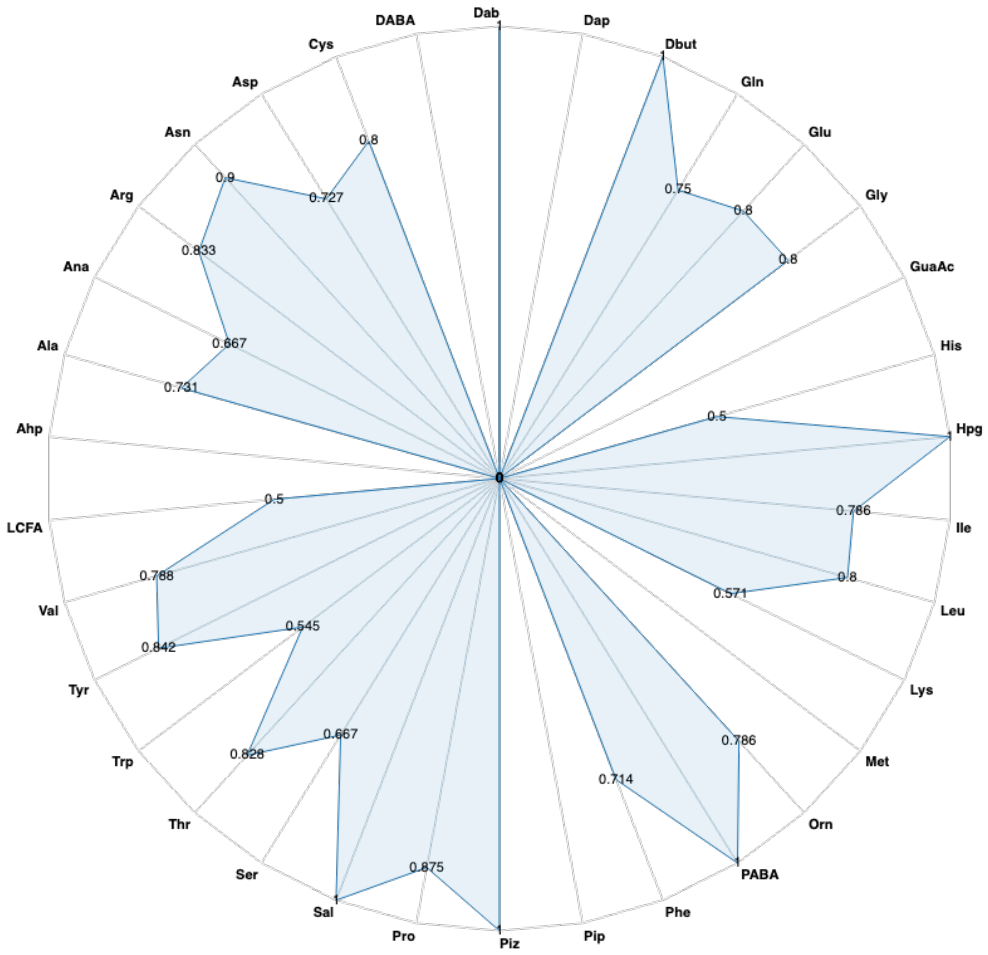
Table 3.1 A table summarising the Adenylation T5 model's performance across the different classification tasks as outputted from each individual classification head. With the exception of LogP and the substrate sequence classification heads, each task achieves over 85% accuracy in the validation set.

	Substrate		TPSA		Hydrogen Bond Acceptors		LogP		CH_2		CH_3	
	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test	Validation	Test
Accuracy Score	0.954	0.991	0.983	1.000	0.983	1.000	0.960	0.991	0.989	0.986	0.931	0.927
Balanced Accuracy Score	0.980	0.995	0.991	1.000	0.991	1.000	0.978	0.994	0.994	0.993	0.627	0.625
F1 Score	0.954	0.991	0.983	1.000	0.983	1.000	0.960	0.991	0.989	0.986	0.931	0.927
Hamming Loss	0.046	0.009	0.017	0.000	0.017	0.000	0.040	0.009	0.011	0.014	0.069	0.073
Hinge Loss	3.632	3.647	0.977	0.959	0.977	0.959	2.477	2.491	0.954	0.959	1.437	1.431
Jaccard Score	0.912	0.982	0.966	1.000	0.966	1.000	0.923	0.982	0.977	0.973	0.871	0.863
Matthew's Correlation Coefficient	0.924	0.985	0.829	1.000	0.829	1.000	0.934	0.985	0.907	0.888	0.868	0.863
Zero One Loss	0.046	0.009	0.017	0.000	0.017	0.000	0.040	0.009	0.011	0.014	0.069	0.073

Table 3.2 A table summarising the Acyltransferase T5 model's performance across the different classification tasks as outputted from each individual classification head. The Acyltransferase T5 model performed well across all tasks and both datasets, with the exception of the CH3 prediction.

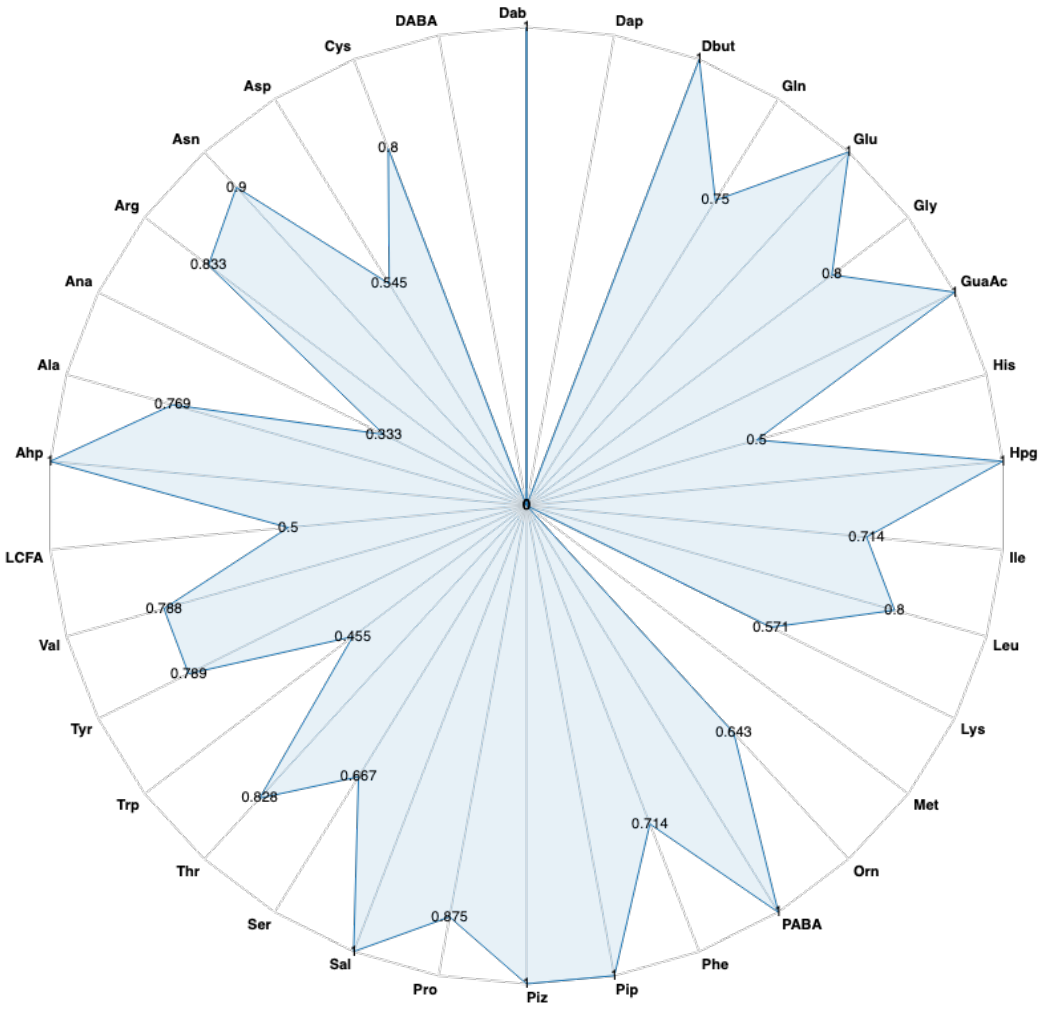
Figure 3.3: Radar plots of validation accuracy of the different substrate prediction strategies using embeddings from the model. (A) The standard sequence classification predicted labels. This uses a linear feed-forward layer to fit the embeddings to logits of the classes trained on. (B) An ensemble model trained using Microsoft's Explainable Boosted Machine (EBM) implementation. It used the predicted properties as input. (C) The FAISS nearest neighbours strategy. Across substrates, the nearest neighbours strategy outperformed or shared similar performance as the other two strategies.

A



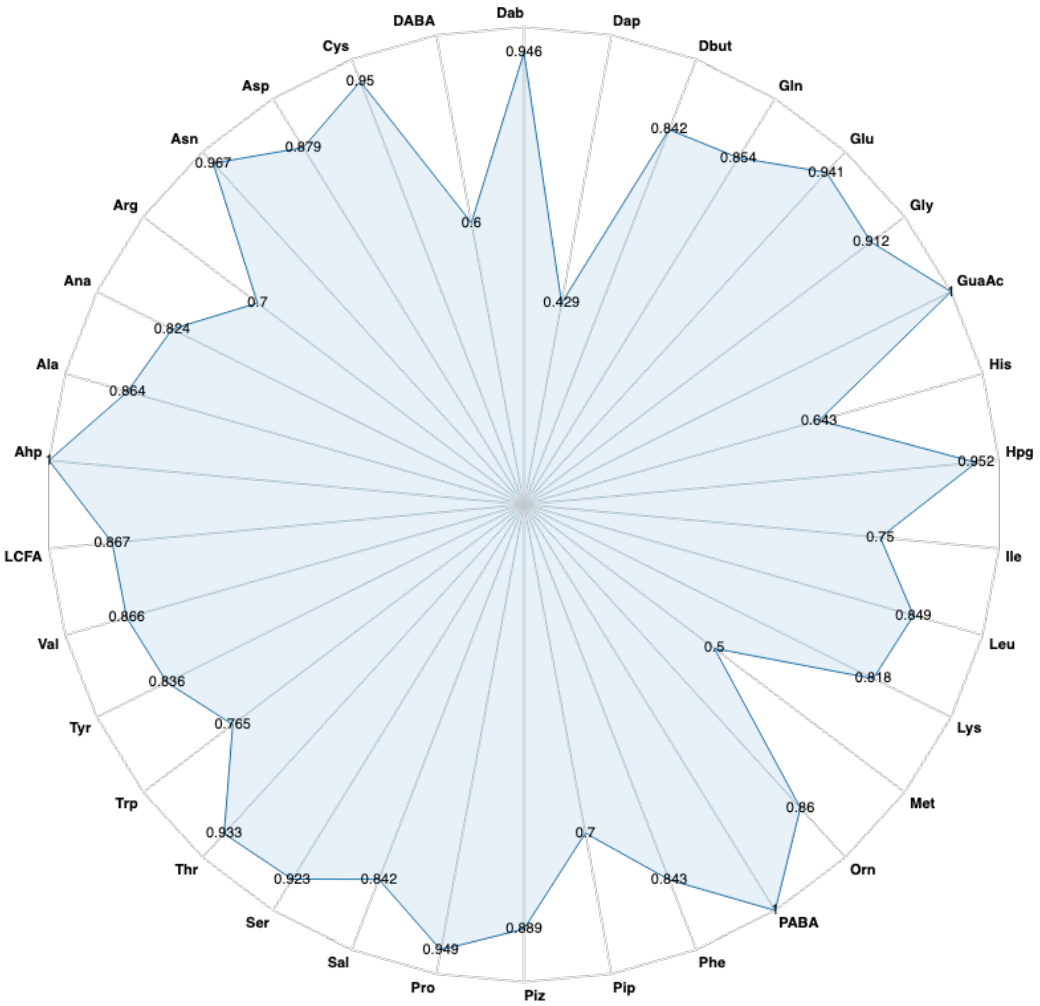
Sequence Classification Head

B



Explainable Boosted Machine

C.



FAISS Nearest Neighbours

Figure 3.4 Charts showing the changing pooled loss, functional accuracy and clade accuracy for the different Polyketide T5 Models. Some models were stopped early due to increased validation loss. Thiolation domains were not trained on clade accuracy.

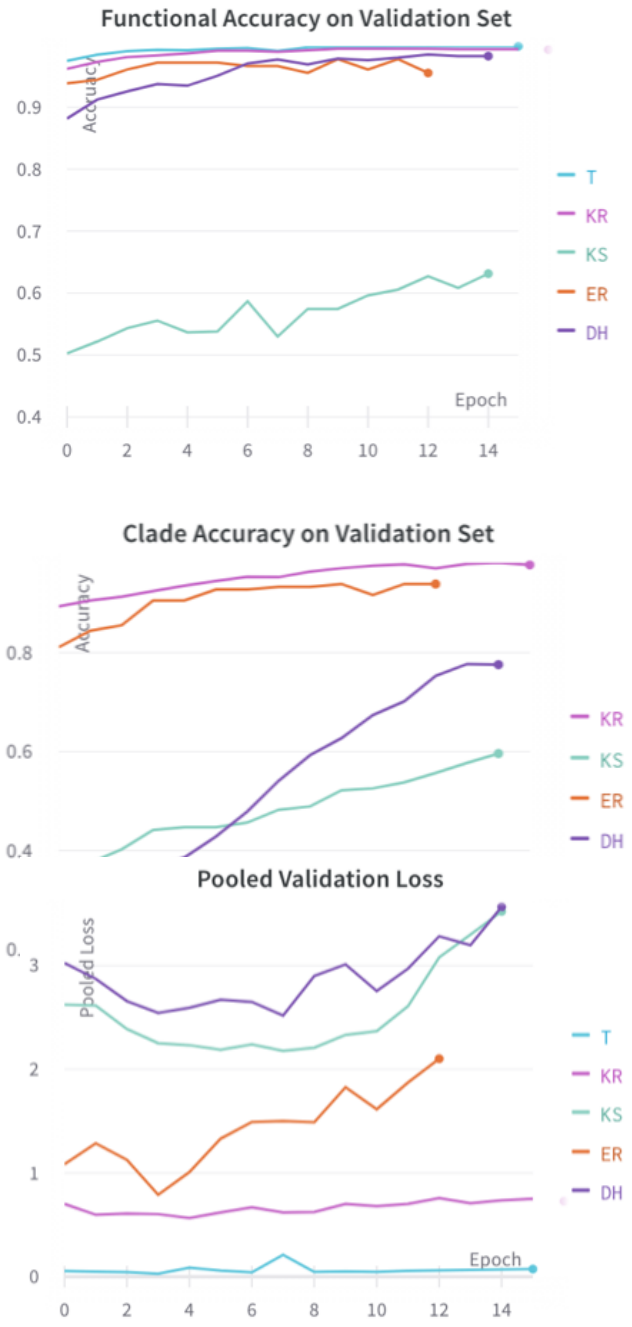


Table 3.3 Performance of the Nearest Neighbour lookup strategy with the polyketide domain models. F1 Scores were above 0.7 for all models in determining whether or not a polyketide domain was functional.

Polyketide Domain T5 LLMs	F1	Precision	Recall
Ketosynthase (KS)	0.71	0.743	0.685
Dehydrotase (DH)	0.88	0.877	0.894
Enoylreductase (ER)	0.911	0.911	0.911
Ketoreductase (KR)	0.976	0.980	0.979
Thiolation (T)	0.987	0.987	0.988

Table 3.4 The performance of EnzymeBERT in predicting enzyme commission numbers, protein families and protein domains. Protein families and enzyme commission number labels were assigned using the nearest neighbour semantic search strategy. Protein domains were assigned using the token classification head.

Annotation Family	F1	Precision	Recall
EC Number: Level 1	0.989	0.989	0.989
EC Number: Level 2	0.984	0.984	0.984
EC Number: Level 3	0.98	0.98	0.98
EC Number: Level 4	0.963	0.968	0.964
Protein Family	0.906	0.909	0.907
Protein Domains	0.92	0.97	0.89

Figure 3.5: The combined attention map for different positions along the Acetyltransferase conserved domain (length of 48 amino acids). The global alignment of the 535 sequences, spans 131 indices. (A) When embedding the first residue in each sequence (i=1), residues within the first seven global positions are mainly attended to. (B) When embedding the 15th residue (i=15), attention is more widespread, with attention mainly spanning from the 10th position to the 40th position. (C) Attention heads for the 30th residue (i=30) and (D) at the 48th residue (i=48), displayed local attention. The attention of residues at a global index position smaller than 52 was negligible. (E) The sequence logo of the acetyltransferase HMM's alignment showed a few gaps. The gaps in the global alignment resulted in exaggerated local attention windows.

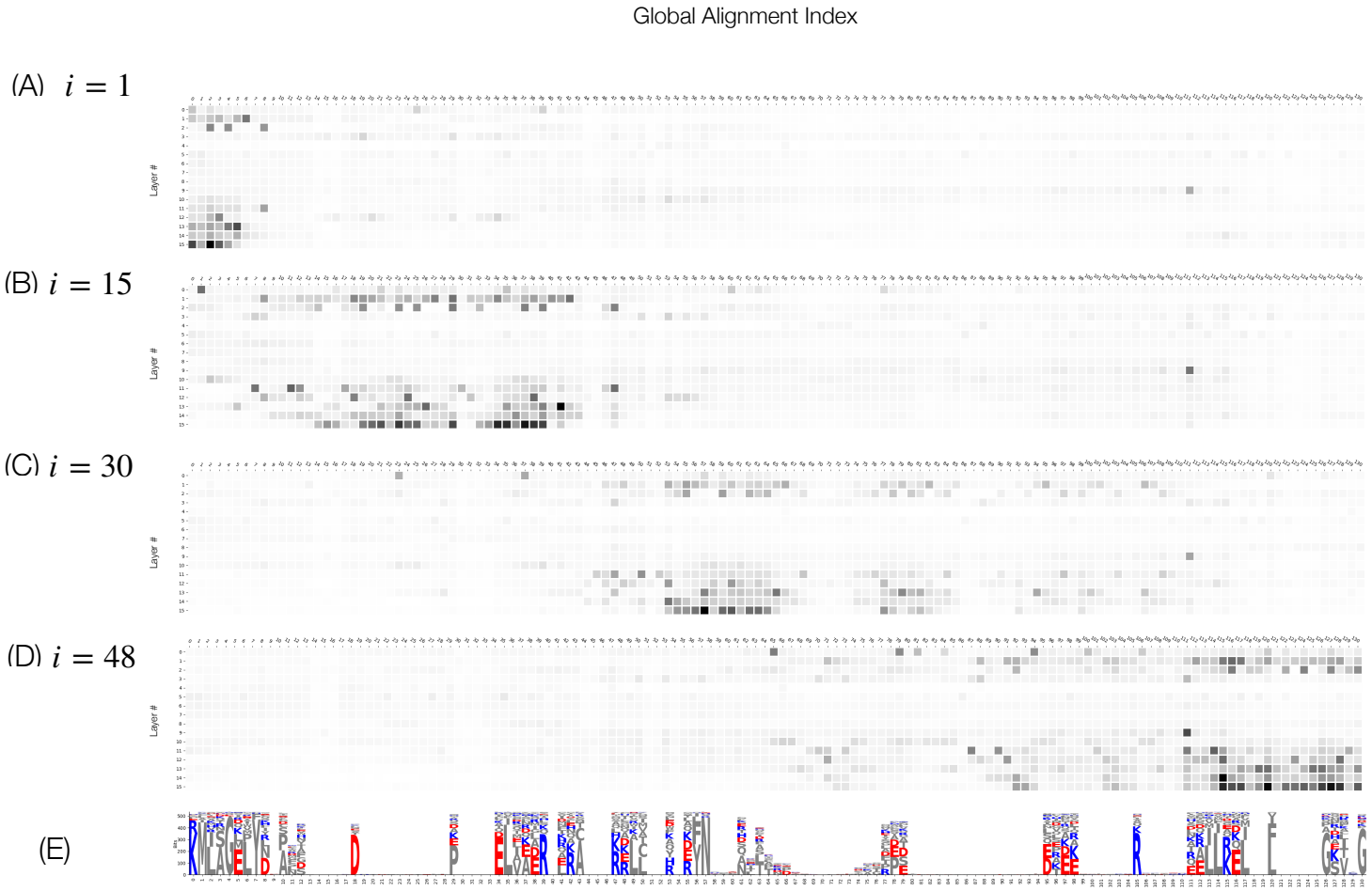
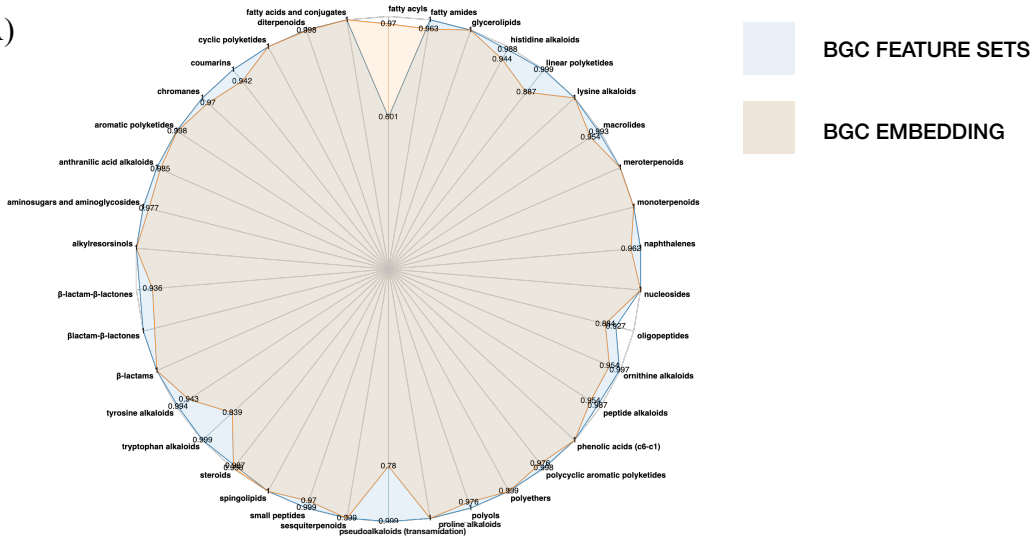


Figure 3.6 Radar plots summarising the chemical validity of BGC comparison methods using a dataset of BGCs with experimentally validated metabolites (n=441). (A) Triplets were derived from the FCFP6 featurization of molecules and dissimilarity was calculated using the Jaccard Index. (B) Triplets were derived using the NP-BERT embeddings of the molecules and the Euclidean distance was calculated. In blue, BGCs are treated as sets using features derived from IBIS; dissimilarity is calculated using the Jaccard Index. In orange, BGCs are treated as the averaged vector of EnzymeBERT embeddings. When a triplet relationship is maintained (the distance of a positive example to the anchor is less than the distance of a negative example to the anchor), a point is awarded. The accuracy is categorised by the superclass of the encoded metabolites.

(A)



(B)

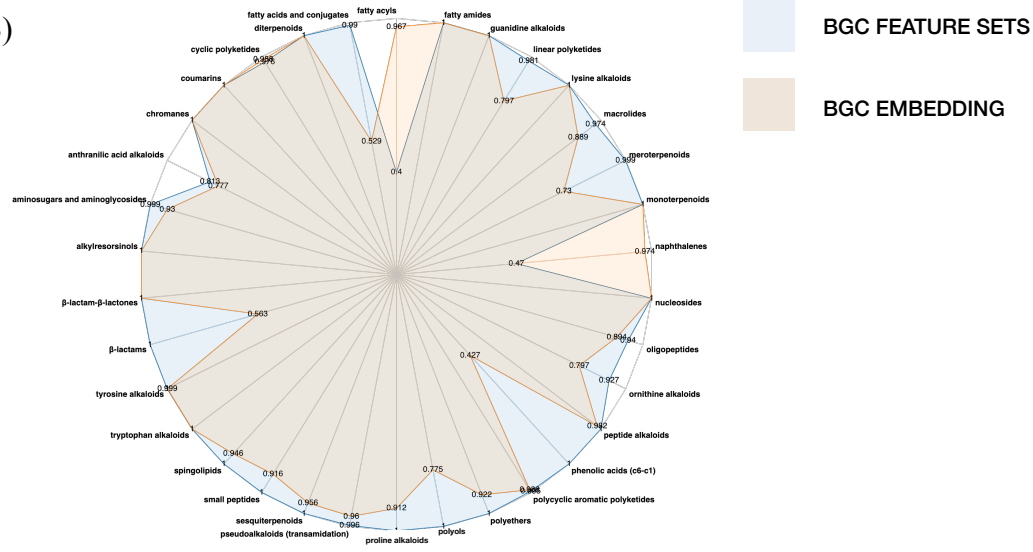


Figure 3.7 The internal dataset of biosynthetic gene clusters was embedded using EnzymeBERT, (n = 296,216) projected to two dimensions (trustworthiness = 0.999), and plotted using UMAP and datashader. Gene clusters were also projected to 128 dimensions (trustworthiness = 0.999) and clustered using HDBSCAN (silhouette score = 0.402). A total of 231,772 BGCs were clustered into 12,495 gene cluster families (GCFs).

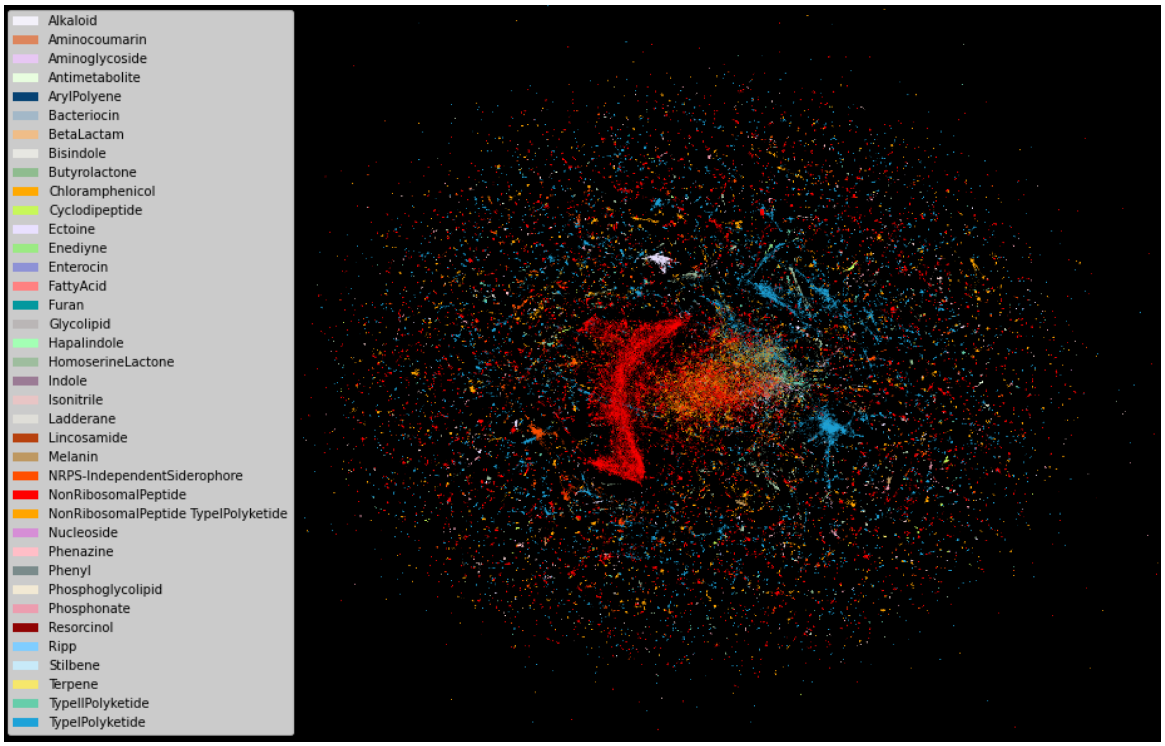


Figure 3.8 Example GCF matching the Curamycin A gene cluster. The original producer of Curamycin A is *Streptomyces cyaneus*. The GCF spans: *Streptomyces katrae* s3, *Streptomyces lavendulae* NRRL B-2774, *Streptomyces viridosporus* DSM 40243, *Streptomyces sp.* NRRL S-87, *Streptomyces sp.* AG109 G2-1, *Streptomyces sp.* RU-71, *Streptomyces sp.* NWU-339, *Streptomyces sp.* TRM-SA0054. (A) The taxonomic network graph showed no conservation on a species level, with the only matches being *Streptomyces* strains without a designated species. (B) The reference Curamycin A gene cluster was nearly identical to the reference BGC with the exception of a different EC number. (C) The Curamycin A metabolite was detected in the metabolomics analysis.

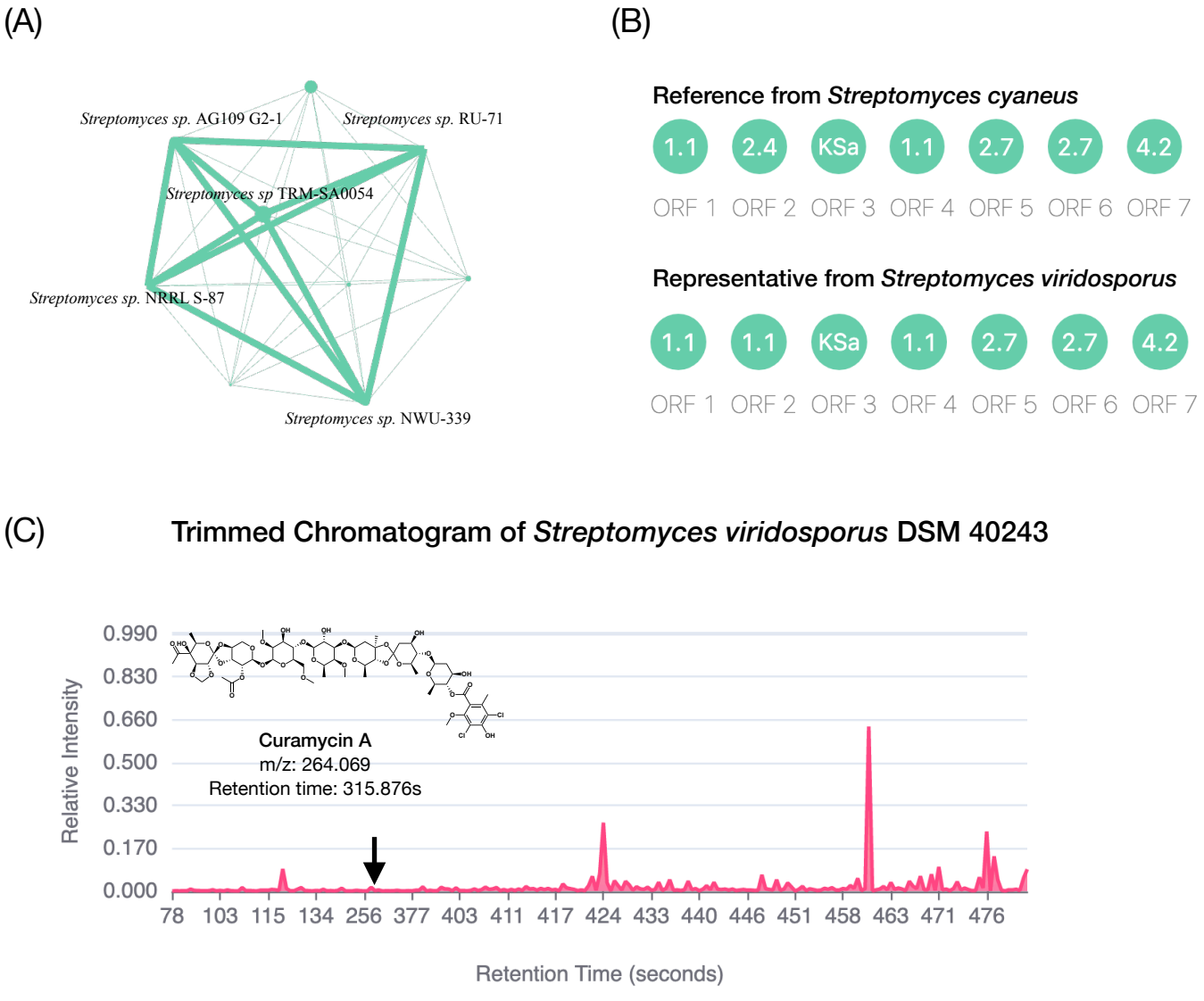


Figure 3.9 Example GCF matching the erythromycin gene cluster. The original producer of Erythromycin is *Saccharopolyspora erythraea* NRRL 2338. The GCF spans multiple genera including: *Saccharopolyspora erythraea* NRRL 2338, *Saccharopolyspora erythraea* DSM 41009, *Saccharopolyspora erythraea* DSM 40517, *Saccharomonospora paurometabolica* YIM 90007, *Aeromicrobium erythreum* AR18, *Micromonospora rosaria* DSM 803, *Streptomyces noursei* ATCC 11455, *Streptomyces yunnanensis* CGMCC 43555, *Streptomyces sp.* MG1, *Streptomyces sp.* NRRL F5193, *Streptomyces sp.* NWU49, and *Streptomyces sp.* CB02120-2. (A) The taxonomic network graph showed the GCF is conserved mainly between two genera *Streptomyces* and *Saccharopolyspora*. (B) There are minor changes between the *Streptomyces* representative versus the *Saccharopolyspora* reference BGC in the first polyketide synthase. (C) Erythromycin G was detected in the alternative *Saccharopolyspora erythraea* strain in the metabolomics analysis.

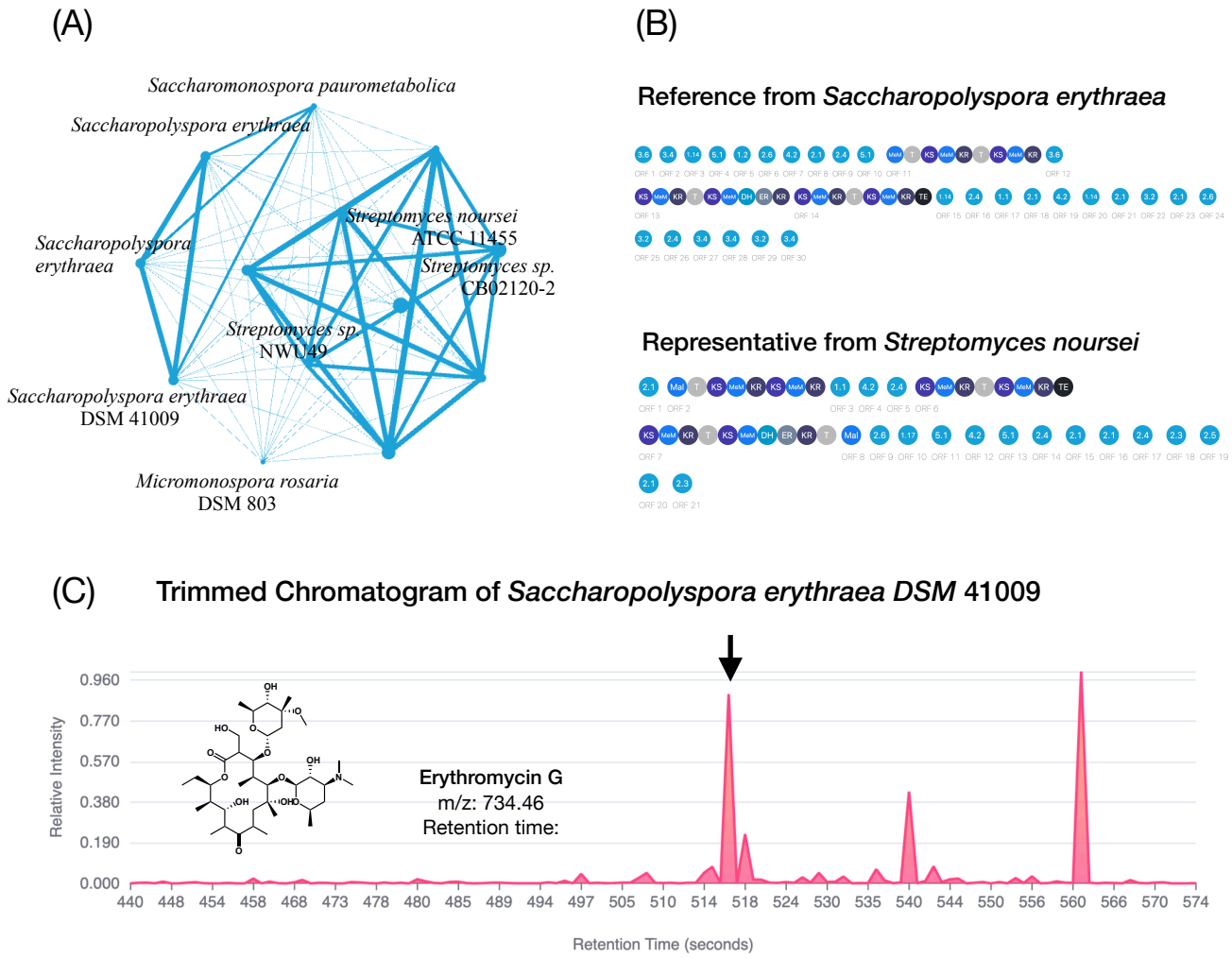
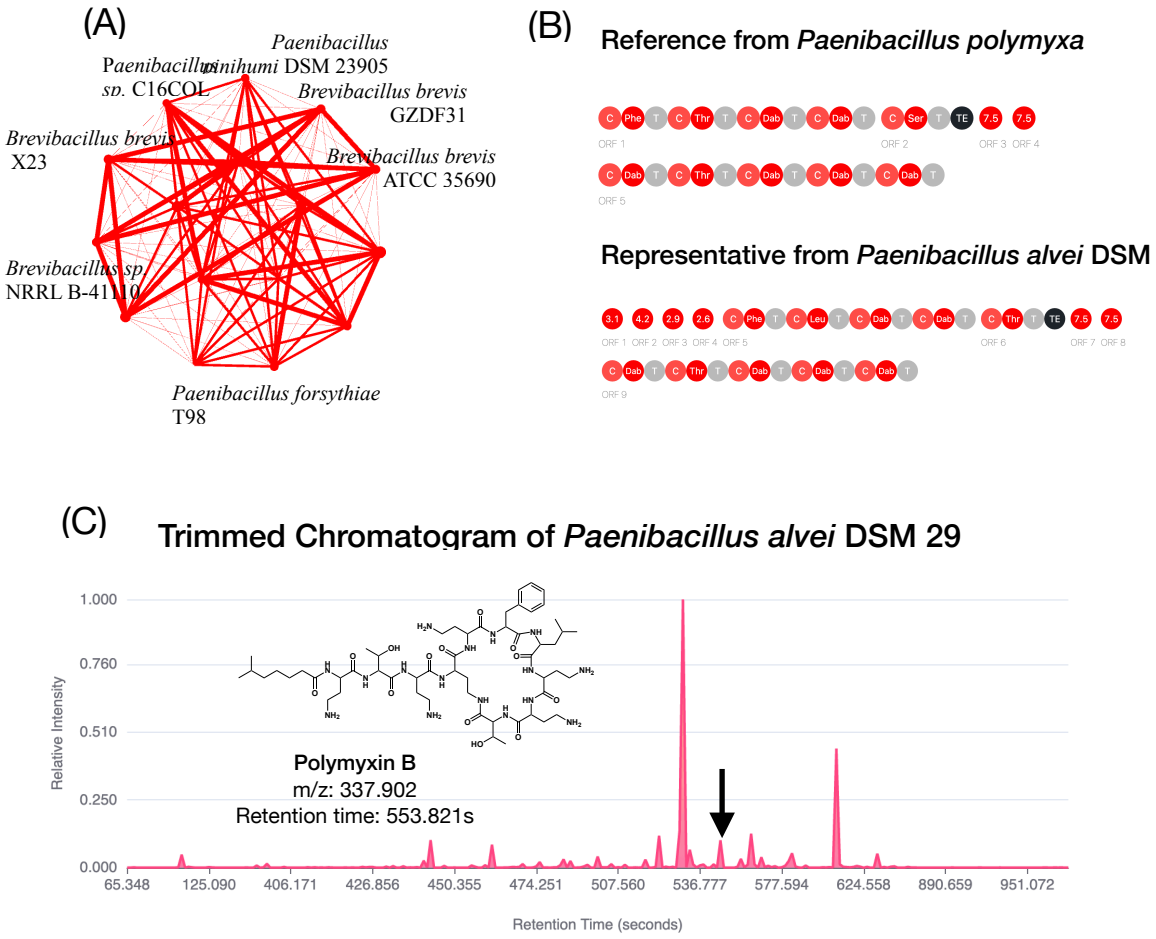
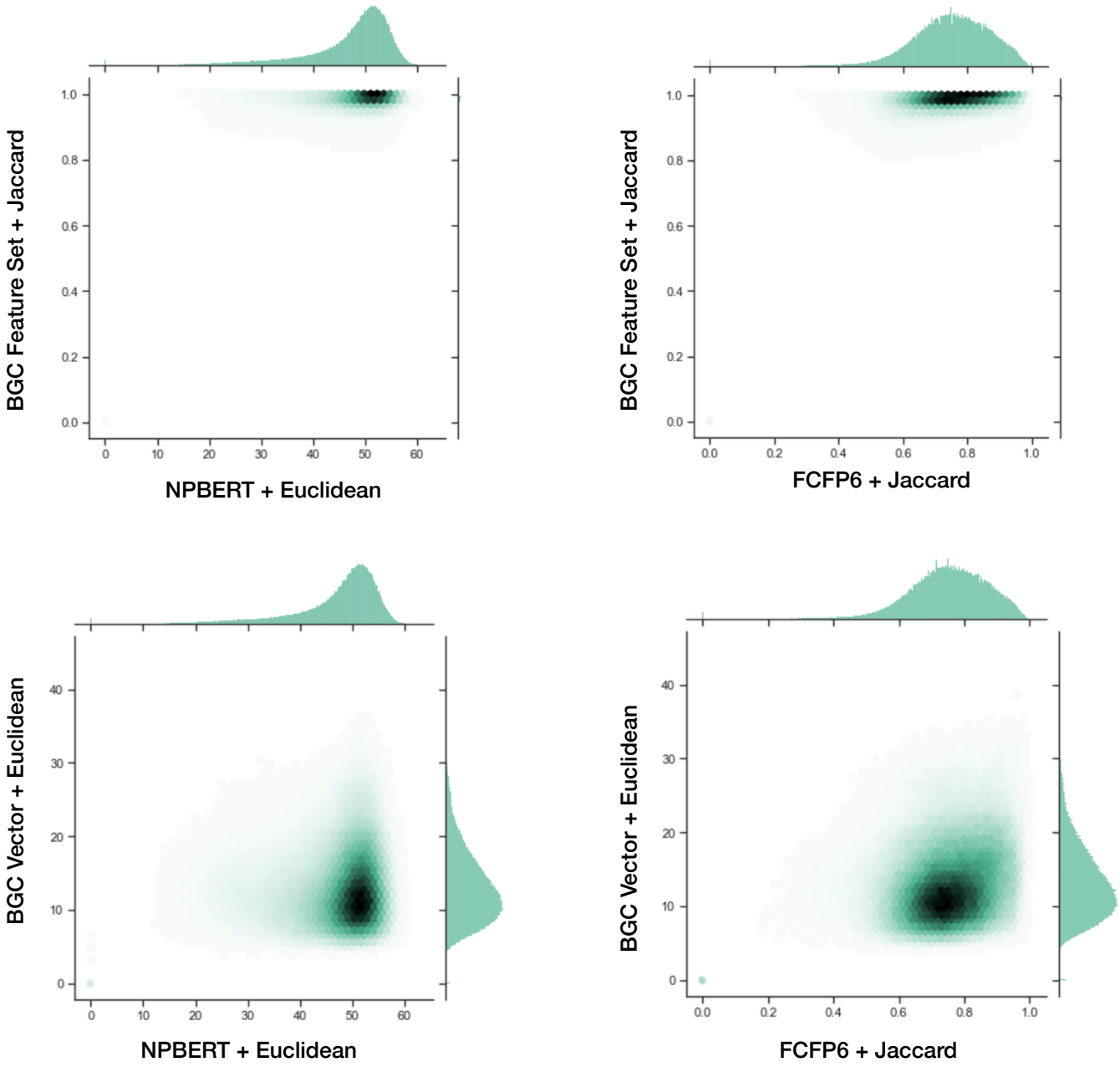


Figure 3.10 Example GCF matching the Polymyxin gene cluster. The original producer of Polymyxin is *Paenibacillus polymyxa*. The GCF spans: *Paenibacillus forsythiae* T98, *Paenibacillus aquistagni* Strain 11, *Paenibacillus alvei* A6-6I-X, *Paenibacillus sp.* IHBB 10380, *Paenibacillus sp.* UNC217MF, *Paenibacillus sp.* ST-S, *Paenibacillus sp.* C16COL, *Paenibacillus alvei* DSM 29, Paenibacillus pinihumi DSM 23905, *Brevibacillus brevis* X23, *Brevibacillus sp.* BC25, *Brevibacillus brevis* DZQ7, *Brevibacillus brevis* ATCC 35690, *Brevibacillus brevis* GZDF31, *Brevibacillus sp.* NRRL B-41110 and *Brevibacillus brevis* NBRC 100599-47. (A) All members of this GCF were mined from the family Paenibacillaceae, so all edges have some weight. There is only a light separation between *Paenibacillus* and *Brevibacillus* mined BGCs. (B) The differences between the reference BGC versus the representative from *Paenibacillus alvei* are minor; two adenylation domains have different substrates predicted in the first two NRPSs, possibly due to error. (C) Polymyxin B was confirmed to be produced by *Paenibacillus alvei* in the metabolomics analysis.



Supplementary Figure S3.1: Joint histogram and density plots for different combinations of BGC and molecular metrics. Histograms are on the outside of the upper and right axis. Density plots are plotted within.



3.7 Bibliography

1. Zazopoulos, E., et al., A genomics-guided approach for discovering and expressing cryptic metabolic pathways. Nature Biotechnology, 2003. **21**(2): p. 187-190.
2. Begani, J., J. Lakhani, and D. Harwani, Current strategies to induce secondary metabolites from microbial biosynthetic cryptic gene clusters. Annals of Microbiology, 2018. **68**(7): p. 419-432.
3. Medema, M.H., et al., antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences. Nucleic Acids Research, 2011. **39**(Web Server issue): p. W339-W346.
4. Skinnider, M.A., et al., Genomes to natural products PRediction Informatics for Secondary Metabolomes (PRISM). Nucleic Acids Research, 2015. **43**(20): p. 9645-9662.
5. Finn, R.D., J. Clements, and S.R. Eddy, HMMER web server: interactive sequence similarity searching. Nucleic Acids Research, 2011. **39**(suppl_2): p. W29-W37.
6. Blin, K., et al., The antiSMASH database version 2: a comprehensive resource on secondary metabolite biosynthetic gene clusters. Nucleic Acids Research, 2019. **47**(D1): p. D625-D630.
7. Skinnider, M.A., et al., Comprehensive prediction of secondary metabolite structure and biological activity from microbial genome sequences. Nature Communications, 2020. **11**(1): p. 6058.
8. Palaniappan, K., et al., IMG-ABC v.5.0: an update to the IMG/Atlas of Biosynthetic Gene Clusters Knowledgebase. Nucleic Acids Research, 2020. **48**(D1): p. D422-D430.
9. Navarro-Muñoz, J.C., et al., A computational framework to explore large-scale biosynthetic diversity. Nature chemical biology, 2020. **16**(1): p. 60-68.
10. Kautsar, S.A., et al., BiG-SLiCE: A highly scalable tool maps the diversity of 1.2 million biosynthetic gene clusters. GigaScience, 2021. **10**(1): p. giaa154.
11. Devlin, J., et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs], 2019.
12. onnx/onnx: Open standard for machine learning interoperability.
13. Altschul, S.F., et al., *Basic local alignment search tool.* Journal of Molecular Biology, 1990. **215**(3): p. 403-410.
14. White, J., et al., Cunningham: a BLAST Runtime Estimator. Nature Precedings, 2011: p. 1-1.
15. Elnaggar, A., et al., ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. arXiv:2007.06225 [cs, stat], 2021.
16. Johnson, J., M. Douze, and H. Jégou, *Billion-scale similarity search with GPUs*. 2017, arXiv.
17. Wang, J., et al. Milvus: A Purpose-Built Vector Data Management System. 2021. Association for Computing Machinery.

18. Grootendorst, M., BERTopic: Neural topic modeling with a class-based TF-IDF procedure. 2022, arXiv.
19. Stachelhaus, T., H.D. Mootz, and M.A. Marahiel, The specificity-conferring code of adenylation domains in nonribosomal peptide synthetases. Chemistry & Biology, 1999. **6**(8): p. 493-505.
20. Shen, J.-J., et al., Substrate Specificity of Acyltransferase Domains for Efficient Transfer of Acyl Groups. Frontiers in Microbiology, 2018. **9**.
21. Chevrette, M.G., et al., SANDPUMA: ensemble predictions of nonribosomal peptide chemistry reveal biosynthetic diversity across Actinobacteria. Bioinformatics, 2017. **33**(20): p. 3202-3210.
22. Landrum, G., et al., rdkit/rdkit: 2020_03_1 (Q1 2020) Release. 2020, Zenodo.
23. Jenks, G.F. The Data Model Concept in Statistical Mapping. 1967.
24. Rasley, J., et al. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. 2020. Association for Computing Machinery.
25. Rajbhandari, S., et al., ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning. arXiv:2104.07857 [cs], 2021.
26. Falcon, W., *Pytorch lightning.* GitHub. Note: https://github. com/ PyTorchLightning/pytorch-lightning, 2019. **3**: p. 6.
27. Pedregosa, F., et al., Scikit-learn: Machine Learning in Python. 2018, arXiv.
28. Wolf, T., et al., HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs], 2020.
29. developers, O.R., *ONNX Runtime*. 2021.
30. Nori, H., et al., InterpretML: A Unified Framework for Machine Learning Interpretability. CoRR, 2019. **abs/1909.09223**.
31. Rajbhandari, S., et al., ZeRO: Memory Optimizations Toward Training Trillion Parameter Models. arXiv:1910.02054 [cs, stat], 2020.
32. Commission of Editors of Biochemical, J., *Enzyme Nomenclature.* Science, 1965. **150**(3697): p. 719-721.
33. Mitchell, A.L., et al., InterPro in 2019: improving coverage, classification and access to protein sequence annotations. Nucleic Acids Research, 2019. **47**(D1): p. D351-D360.
34. Mistry, J., et al., Pfam: The protein families database in 2021. Nucleic Acids Research, 2021. **49**(D1): p. D412-D419.
35. The UniProt, C., UniProt: the Universal Protein Knowledgebase in 2023. Nucleic Acids Research, 2023. **51**(D1): p. D523-D531.
36. Blondel, V.D., et al., *Fast unfolding of communities in large networks.* Journal of Statistical Mechanics: Theory and Experiment, 2008. **2008**(10): p. P10008.

37. Overbeek, R., et al., *The use of gene clusters to infer functional coupling.* Proceedings of the National Academy of Sciences of the United States of America, 1999. **96**(6): p. 2896-2901.
38. Karp, P.D., et al., *The MetaCyc Database.* Nucleic Acids Research, 2002. **30**(1): p. 59-61.
39. Kanehisa, M., et al., KEGG as a reference resource for gene and protein annotation. Nucleic Acids Research, 2016. **44**(D1): p. D457-D462.
40. Chang, A., et al., BRENDA, the ELIXIR core data resource in 2021: new developments and updates. Nucleic Acids Research, 2021. **49**(D1): p. D498-D508.
41. Leach, A.R., et al., Implementation of a System for Reagent Selection and Library Enumeration, Profiling, and Design. J. Chem. Inf. Comput. Sci., 1999. **39**(6): p. 1161-1172.
42. Logomaker: beautiful sequence logos in Python | Bioinformatics | Oxford Academic.
43. xxHash - Extremely fast non-cryptographic hash algorithm.
44. Zhu, E. and V. Markovtsev, Ekzhu/Datasketch: First Stable Release. 2017, Zenodo.
45. Nolet, C.J., et al., Bringing UMAP Closer to the Speed of Light with GPU Acceleration. 2021, arXiv.
46. Venna, J. and S. Kaski. Neighborhood Preservation in Nonlinear Projection Methods: An Experimental Study. 2001. Springer.
47. McInnes, L., J. Healy, and J. Melville, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 [cs, stat], 2018.
48. Bednar, J.A., et al., holoviz/datashader: Version 0.14.3. 2022, Zenodo.
49. McInnes, L., J. Healy, and S. Astels, hdbscan: Hierarchical density based clustering. The Journal of Open Source Software, 2017. **2**.
50. Rousseeuw, P.J., Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 1987. **20**: p. 53-65.
51. Bastian, M., S. Heymann, and M. Jacomy, *Gephi: An Open Source Software for Exploring and Manipulating Networks.* Proceedings of the International AAAI Conference on Web and Social Media, 2009. **3**(1): p. 361-362.
52. Hunter, S., et al., InterPro: the integrative protein signature database. Nucleic Acids Research, 2009. **37**(Database issue): p. D211-D215.
53. Ryu, J.Y., H.U. Kim, and S.Y. Lee, Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. Proceedings of the National Academy of Sciences of the United States of America, 2019. **116**(28): p. 13996-14001.
54. Patel, M., TinySearch -- Semantics based Search Engine using Bert Embeddings. 2019, arXiv.
55. Deshmukh, A.A. and U. Sethi, IR-BERT: Leveraging BERT for Semantic Search in Background Linking for News Articles. 2020, arXiv.

56. Morgan, K.D., R.J. Andersen, and K.S. Ryan, Piperazic acid-containing natural products: structures and biosynthesis. Natural Product Reports, 2019. **36**(12): p. 1628-1653.
57. Baranašić, D., et al., Predicting substrate specificity of adenylation domains of nonribosomal peptide synthetases and other protein properties by latent semantic indexing. Journal of Industrial Microbiology and Biotechnology, 2014. **41**(2): p. 461-467.
58. Rogers, A., O. Kovaleva, and A. Rumshisky, A Primer in BERTology: What we know about how BERT works. arXiv:2002.12327 [cs], 2020.
59. Wang, P., et al., Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network. 2015, arXiv.
60. Liu, Y., et al., RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs], 2019.
61. Beltagy, I., M.E. Peters, and A. Cohan, *Longformer: The Long-Document Transformer*. arXiv:2004.05150 [cs], 2020.
62. Zaheer, M., et al., Big Bird: Transformers for Longer Sequences. 2021, arXiv.
63. Medema, M.H., E. Takano, and R. Breitling, Detecting sequence homology at the gene cluster level with MultiGeneBlast. Molecular Biology and Evolution, 2013. **30**(5): p. 1218-1223.
64. Hadjithomas, M., et al., IMG-ABC: new features for bacterial secondary metabolism analysis and targeted biosynthetic gene cluster discovery in thousands of microbial genomes. Nucleic Acids Research, 2017. **45**(D1): p. D560-D565.
65. Kautsar, S.A., et al., BiG-SLiCE: A highly scalable tool maps the diversity of 1.2 million biosynthetic gene clusters. GigaScience, 2021. **10**(1): p. giaa154.
66. Hill, F., K. Cho, and A. Korhonen. Learning Distributed Representations of Sentences from Unlabelled Data. in NAACL-HLT 2016. 2016. Association for Computational Linguistics.
67. Kenter, T., A. Borisov, and M. de Rijke, Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. 2016, arXiv.
68. Arora, S., Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. in 5th International Conference on Learning Representations, ICLR 2017. 2019.
69. Sinoara, R.A., et al., Knowledge-enhanced document embeddings for text classification. Knowledge-Based Systems, 2019. **163**: p. 955-971.
70. Radford, A., et al., Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs], 2021.

Chapter 4: Deep Learning Guided De Novo Assembly of Bacterial Genomes

4.1 Chapter Preface

To develop a deep learning approach to bacterial genome assembly, I initially worked with two co-op students, Cameron Default and Daniel Di Cesare on pivoting DNABERT. While the framework worked with PacBio reads, it did not work with Illumina datasets. Unfortunately, Illumina datasets made up the majority of the unfinished bacterial genomes in RefSeq. After much testing, it was concluded that calculating semantic meaning from short reads was not a feasible task.

I developed a new workflow dedicated to joining partial scaffolds in a similar manner to hybrid sequencing. As a trained bioinformatician, I was able to deconstruct the Unicycler pipeline and deduce where would be most suitable to add additional inference. I created two models dedicated to de Bruijn graph bridging: (1) a Bacterial T5 model and (2) a graph deep learning model. I created a custom framework to train a T5 model from scratch using PyTorch. Norman Spencer curated a dataset of representative bacterial genomes selected from the tree of life using genomic distances. Using the dataset, I trained a Bacterial genomic model. I created a new method for performing sentence pair classification that minimised the memory impact of conventional next-sentence prediction. After analysing the GCFs calculated by IBIS, it was observed that many of the family members were fragmented ORFs that otherwise appeared related to the complete

BGCs. EnzymeBERT embedded fragmented and whole peptides in the same manner. To build off of this behaviour, I developed a custom graph deep learning framework to teach the model spatial relations between ORFs. I trained the model using public datasets. I developed the FuzzyAlign algorithm along with the procedures to perform Sweep optimisation using the server cluster.

Irina Sementchoukova and Tonya Malcom developed a strategy for using public datasets to direct sequencing efforts. Irina and Tonya performed the DNA preparation for hybrid sequencing. Using my pipeline and the protocol from the Surette lab, I was able to reconstruct hybrid genomes, with one being fully finished. Dr Xiaxia Di performed all analytic chemistry experiments to isolate and identify chymostatin A from *Streptomyces orinoci* DSM 40571. Through a literature search and reassembly, I was able to find the BGC responsible for chymostatin.

4.2 Abstract

While genomic mining technologies continue to improve in the mapping of the natural product space, fragmented genomes remain a bottleneck for progress. Previous tools have successfully utilised biosynthetic domains as a bridge for recovering whole microbial contigs. We present NALA (Neural-network-guided Arrangement and Linkage Assembly), a new pipeline utilising a graph deep learning strategy in combination with

biosynthetic large language models to assist in de novo assembly. We utilise NALA to improve four publicly available assemblies from *Cohnella, Streptomyces, Chitinophaga* and *Pseudozobellia*. We demonstrate increased contig sizes and improved genomic mining using our pipeline.

4.3 Introduction

The creation of Next-generation sequencing (NGS) has provided a wealth of bacterial genomes for the public. The Reference Sequence (RefSeq) project at National Center for Biotechnology Information (NCBI) contains genomes from over 74,000 bacterial and archaeal taxa.[1] The majority of bacterial genomes in RefSeq have been assembled using short-reads from Illumina sequencing efforts (Figure 4.1). The popularity of the Illumina platform can be attributed to its high accuracy and low cost per base. Unfortunately, the short fragments of Illumina sequencing (500 bp or less) are smaller than many of the repetitive elements in bacterial genomes. In bacterial genomes, it is common for secondary metabolism to be encoded by repeating biosynthetic modules (e.g. non-ribosomal synthases, polyketide synthases). The repetitive nature of the bacterial genomes leads to highly fragmented de novo assemblies when sourced from short-read libraries alone. Many of the RefSeq genomes assembled from Illumina short reads display a low N50 (Figure 4.2).

Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT) are alternative sequencing platforms with longer reads (10+ kbp) but at a higher cost than Illumina.[2, 3] Many long-read de-novo assemblers have been made such as Canu, Flye, Raven, Miniasm, and wtdbg2.[4-8] The long-read platforms are not without their faults; they have a lower base call accuracy (87%) and chimeric reads are probable.[9, 10]. To create the most accurate genomes with the least amount of fragmentation, hybrid approaches to assembly have become popular. One approach involves using the highly accurate Illumina reads to correct the long reads, before using a long-read assembler; the high sequencing depth required for assembly makes this an expensive approach. A cheaper approach is to use low-coverage PacBio or Nanopore long-reads to fill gaps in a short-read assembly as is performed in Unicycler.[11, 12]

Unicycler first uses SPADES to develop the initial de Bruijn graph with short-read sequencing data.[13, 14] Long-reads are then semi-globally aligned with the short-read assembled contigs to resolve repeats in the graph structure. Based on the alignment, Unicycler creates bridges between the contigs and scores them according to a series of metrics (alignment quality, read agreement, graph path etc). It also creates bridges based on a series of other strategies including repair information and the Miniasm assembly.[4] All of the bridges are sorted by quality and applied in reverse order so the highest quality bridges are applied last. Assemblies can then be optionally polished using Pilon.[15] The

strategy is very effective, with Unicycler outperforming other hybrid assembly pipelines. [12, 16]

Pipelines have also been developed for reference-guided de novo assembly. In the same manner that hybrid pipelines use long-reads to bridge incomplete short-read graphs, reference-guided de novo assembly utilises the completed genomes of related species to construct a genome.[17-23] There are some disadvantages, including introducing biases into the assembly towards the reference genome, diverging regions failing to be assembled, and genomic rearrangements differing between species leading to misassemblies.[20, 24-27] One solution proposed for these problems is to introduce multiple reference genomes for increased flexibility.[28, 29]

The short-read RefSeq genomes do not have a corresponding long-read dataset to perform a hybrid assembly. As an alternative, reference datasets can be of use to create more robust genomes. The bias introduced by selecting a poor reference genome can negatively impact the downstream analyses.[30-32] Instead, we propose teaching a deep learning model of the spatial relationships in bacteria genomics and allowing it to predict bridges. In 2021, a deep learning model trained on human genomics called DNABERT was released. It demonstrated a transformer could understand DNA sequences based on up and downstream nucleotide contexts. It was able to predict promoters, splice sites and transcription factor binding sites. Using DNABERT's attention, conserved sequence

motifs and functional genetic variant candidates could be identified.[33] The latent space of a transformer trained on nucleotide information can be useful for detecting features relevant to bacterial genome assembly.

Beyond utilising reference genomes to build bridges, we also propose using encoded peptide information. Taboada, Verde and Merino created an operon-calling pipeline using deep learning.[34] They trained a multilayer perceptron using functional relationships defined by STRING and intergenic distances as features.[35] The pipeline was able to predict operons within bacterial genomes with over 90% accuracy. They demonstrated protein function can be informative in deducing genomic spatial relationships.

In this work, we present our Neural-network-guided Arrangement and Linkage Assembly (NALA) pipeline. We use a transformer trained on bacterial genomes to predict spatial relationships between short-read contigs. We also use a graph neural network to predict spatial relationships between peptide sequences encoded on the periphery of short-read contigs. Inference from both models is integrated into Unicycler for robust assembly. Metrics for both bridging techniques are calculated on an in silico sequenced dataset of biosynthetic gene clusters. In addition, we prepare new assemblies of four publicly available genomes with our pipeline.

4.3 Methods

The deep learning-based Unicycler bridges are based on two models: (1) Bacterial T5 - a T5 generative model trained to predict whether or not two contigs are spatially lateral to one another. (2) Enzyme-GNN - A graph neural network trained to predict which peptide sequences are spatially connected based on EnzymeBERT embeddings. Hyperparameter optimisation was used to integrate both custom bridges into the Unicycler pipeline.

4.3.1 Bacterial T5

4.3.1.1 Architecture

There are no publicly available transformers trained on bacterial genomes. DNABERT was trained on the human genome. Based on the performance of ProtT5 versus ProtBERT, we decided on a T5Encoder instead of BERT as the base architecture for bridging.[33, 36] The T5ForConditionalGeneration model from Hugging Face's Transformers package was used to pre-train the initial BacterialT5 text-to-text model.[37]

4.3.1.2 Input/Output

As an input for the model, we split genomic sequences into 6-mers as described by DNABERT.[33] For tokenisation, we use the DNABERT tokenizer with a modified

vocabulary; we added 100 sentinel tokens to allow for span-masked language modelling. In addition, a class token was appended to every tokenised sentence as is customary with BERT tokenisation.[38]

4.3.1.2 Pre-training

We pretrained BacterialT5 with the span-masked language modelling training protocol as described in the original T5 paper. [39] The base transformer used was the T5ForConditionalGeneration interface implemented in HuggingFace's transformers package.[37] The Pytorch implementation was used to take advantage of optimisations made for training in PyTorch Lightning and DeepSpeed.[40-42] There only T5 span-masked language modelling collator is programmed using JAX. We reimplemented the collator reimplemented using PyTorch.

Span-masked language modelling is a combination of question and answering with masked language modelling. The collator randomly masks spans of an input sequence, where each masked span is enumerated. The model will encode the masked sequence to an embedding, and then decode the embedding back into an output sequence. The output returns the masked spans with the enumeration intact.[39] We set a maximum of 30% of the sequence to be masked. The model was trained on batches of 16, with the maximum length of an input sequence constrained to 1024 tokens.

To create a training dataset all of the bacterial genomes from RefSeq were downloaded. To avoid training on all of RefSeq, 1000 genomes were sampled based on genomic distances. Distances between genomes were calculated using the tool Dashing.[43] All genomes were annotated with open-reading frames (ORFs) using Pyrodigal.[44] The ORFs were used as sentence input for the span-masked language modelling task. The model was trained for a total of 40 epochs across 5 GPUs. The results of the pretraining can be found in Section 4.4.1.

4.3.1.3 Finetuning Tasks

To predict spatial relationships between genomic regions, a next-sentence prediction (NSP) task was prepared. A custom collator was designed to create the dataset for sentence pair classification. The collator would randomly sample a group of nucleotide sequences, split the sequences into halves, and then randomly pair the halves together. Halves belonging to the same sequence were considered a match while those belonging to separate sequences were not. Each nucleotide sequence pair was separately embedded. The embeddings for the pair's class tokens were concatenated together to be classified as a joint embedding. To fine-tune BacterialT5 for the NSP task, the T5Encoder from the T5ForConditionalGeneration model was extracted. A custom sentence pair classification head was developed.

Two separate datasets were created for curriculum learning. The first NSP fine-tuning task was performed with ORFs detected from another sample of 1000 RefSeq genomes. ORFs were split into chunks of 1024 tokens. Based on the literature, there is also some conservation in intergenic regions, possibly due to evolutionary degradation. To learn these relationships, a second NSP fine-tuning task was performed with intergenic regions. The intergenic regions were extracted from the first sample of 1000 RefSeq genomes. Intergenic regions were split into chunks of 256 tokens. The results of both fine-tuning tasks can be found in Section 4.4.1.

4.3.2 Network Graph Link Prediction with EnzymeBERT and RevGNN

4.3.2.1 Architecture

The deepest graph neural network architecture with a low memory footprint currently in production is the Reversible GNN (RevGNN) developed by Li et al. [45] A link-prediction framework built around the RevGNN was set up using PyTorch Geometric, PyTorch Lightning and a custom in-house framework.[40, 46] After preliminary tests, a total number of 20 RevGNN layers were chosen for the final model.

4.3.2.2 Input/Output

To train the RevGNN, we represented contigs as weighted network graphs in PyTorch Geometric.[46] Each node was an open reading frame and weighted edges were drawn between ORFs within 10 kbp. Edge weights were calculated as the absolute distance between predicted peptide sequences within the contig. Node embeddings were initialised using the EnzymeBERT transformer.

4.3.2.3 Training Tasks

Using a link prediction task, the RevGNN was trained to predict which peptides were spatially collocated. MiBiG 2.0 was used as the training dataset; the dataset contains trimmed nucleotide sequences corresponding to different biosynthetic gene clusters (BGC). [47] Genes found within a BGC have inherently related functionalities that are captured by EnzymeBERT.

For the link prediction task, a custom collator was used to randomly select a batch of 10 contigs and created a weighted network graph; collocated genes within the BGCs were represented as connected components. In addition, false edges were created using PyTorch Geometric's negative sampling method. The dot-product of nodes embedded by the RevGNN was used to calculate the probability of an edge being real.[48] This task was trained for 400 epochs. To speed up training, Facebook's 8-bit AdamW optimiser was used.[49] The results of the link prediction task training can be found in Section 4.4.2.

4.3.2.4 Post-Processing

While the EnzymeGNN model predicts spatial relationships between open reading frames, it does not provide direct inference between contigs. To create weighted edges between contigs in the same manner as BacterialT5, contigs are represented as nodes and weighted edges are drawn using pooled predictions of peripheral ORFs (encoded within 10 kbp of the end of a contig). Weighted edges are calculated where the weight is equal to the total number of positive linkages divided by the total number of potential linkages.

4.3.3 Integration of Neural Network Inference into the Unicycler Framework

Unicycler avoids reinventing the wheel and only optimises assembly where it can integrate new information.[12] The different strategies used for bridging are all assigned separate ranks and qualities depending on their source. Bridges are ranked by two factors, the strategy rank and the bridge quality score. Bridges are then applied in reverse order, where the least reliable bridges are applied first. To add the neural network inference to Unicycler, additional bridges were created reflective of the two strategies with their ranks and quality scores determined using hyperparameters optimisation.

4.3.3.1 Adding Bacterial T5 and EnzymeGNN Inference to Unicycler

In the modified Unicycler pipeline, after the initial assembly with SPADES, contigs were processed by BacterialT5.[14] The NSP classification head was used to predict which contigs should be spatially paired together. A bridge was created for every pair the BacterialT5 model classified as True. The probability of the pair being True was multiplied by a weight and assigned as the quality of the bridge. The bridge weight, the bridge weight cut-offs, the bridging strategy rank and other bridge-related variables were all assigned as dynamic hyperparameters for optimisation.

To add EnzymeBERT-GNN bridges, each contig was first mined for ORFs using Pyrodigal.[44] Each ORF was then embedded by EnzymeBERT and used to create a network graph where all peripheral ORFs were connected. The RevGNN model was then passed all of the edges and using the link prediction fine-tuning head, it determined which of the edges were correctly assigned. The dot-product probabilities were multiplied by a separate weight. As with the BacterialT5 bridges, all variables were assigned as dynamic hyperparameters for optimisation.

4.3.3.2 Bridge Hyperparameter Optimisation with Weights and Biases Sweeps

To create the dataset for optimisation, a small ground truth dataset was created using in-silco sequencing data generated with InSilcoSeq.[50] Source contigs were

selected from an internal dataset of 1,061 BGCs. A total of 200,000 reads were generated for each BGC using the short-read error model “miseq”.

Alignment-based scoring metrics were too slow to optimise for robust assemblies quickly. As a heuristic, a novel metric was created called FuzzyAlign. To score the similarity of the two assembled contigs (ground truth versus Unicycler assembly), both sequences were first split into chunks of 500 base pairs. Each sequence was compared using the Levenshtein distance to create a distance matrix between the chunks.[51] Using the Hungarian method, a linear sum assignment was solved to determine which chunks were the most similar between the two sequences. [52] The indices were then compared with the Spearman correlation to score how similar the assigned indices are to one another. The package “FuzzyWuzzy” was used to score Levenshtein distances. The package SciPy was used for linear sum assignment and correlations.[53]

The sweeps tool designed by Weights and Biases was used to perform a bayesian hyperparameter optimisation.[54] Every hyperparameter was optimised to maximise the FuzzAlgin score; there were a total of 783 iterations. After determining the optimal parameters, the pipeline was assigned static values. Results for the hyperparameter optimisation can be found in Section 4.4.3.

4.3.4 Experiments

4.3.4.1 Comparison of different assembly strategies.

To identify the impact of the deep learning-based bridges on the assembly, QUAST was used to score different combinations of assembly tools on the BGC dataset. Assemblies were generated using BiosyntheticSPADES, Unicycler, Unicycler + Bacterial T5, Unicycler + EnzymeGNN, and Unicycler + EnzymeGNN with Bacterial T5.[55] Unicycler-modified pipelines used the base SPADES de Bruijn graph. Only incomplete assemblies from SPADES (i.e. more than one contig) were selected for QUAST scoring (n=521). Results can be found in Section 4.4.4.

4.3.4.2 Publicly Available Datasets Reassembled with Multiplexed Non-Barcoded PacBio Long-Reads

While PacBio sequencing is more expensive per kilo base pair than Illumina short-read sequencing, there are multiplexing strategies that can greatly bring down the price. A protocol for non-barcoded multiplexed PacBio sequencing was recently released. First a putative genome is assembled using short-reads from Illumina sequencing of an isolate; this is done using Unicycler. Similar to strategy used in reference guided de novo assembly, the putative genome is used as a reference to align the multiplexed PacBio and create isolate pools; this is performed using minimap2. The Illumina reads and aligned PacBio reads are then used for hybrid assembly in Unicycer.[4, 56]

We implemented this protocol in-house, by first performing a pooled, non-barcoded PacBio sequencing run of multiple microbes of interest. Microbes were selected on the basis of publicly available genome availability. Short-read datasets were sourced from NCBI's SRA database. The samples were prepped with the SMRTbell® Express Template Prep Kit 2.0. The PacBio read pools and publicly available Illumina reads were assembled using Unicycler and the NALA pipeline. Assembly metrics were generated using QUAST.[57] In addition, assembled were mined by PRISM 3 for BGCs and additional metrics were calculated.[58] Results from the assembly can be found in Section 4.4.5.

4.3.4.3 Reassembly of *Streptomyces orinoci* DSM 40571

Streptomyces orinoci DSM 40571 was fermented in-house and a number of metabolites were isolated including Chymostatin A. The only publicly available genome *Streptomyces orinoci* DSM 40571 is "GCA_003121295.1". The genome is highly fragmented with 44 contigs with an N50 of 434,919 bp and a sequence length of 7,502,208 bp. There are multiple SRA projects available for *Streptomyces orinoci* DSM 40571. An Illumina MiSeq project is available at SRX3418672 and a PacBio RS II project is available at SRX3418671. Using these two projects, we were able to perform a hybrid assembly using the NALA pipeline. The completed genome was mined by PRISM

3, annotated by IBIS and putative gene clusters was investigated. Results for the assembly can be found in Section 4.4.5.

A putative gene cluster for Chymostatin A was recently reported from *Streptomyces mobaraensis*.[59] While the gene cluster has not been added to MiBiG as yet, the UniProt IDs were published. To determine if the peptides were located in the reassembled genome, *cstA-G* were embedded by EnzymeBERT and the nearest neighbour from the reassembled *Streptomyces orinoci* DSM 40571 genome was located. Results from the IBIS analysis of the putative chymostatin gene cluster can be found in Section 4.4.6.

4.4 Results

4.4.1 Bacterial T5 Training

During the pretraining of the BacterialT5 model, the loss plateaued after 7 epochs. A training session was started with a new dataset of genomes but was stopped early after no change loss was observed. The training and validation loss curves for span-masked language modelling of the BacterialT5 model can be found in Figure 4.3. During the next-sentence prediction fine-tuning task on ORFs, the performance plateaued within 3 epochs at 96.129% accuracy. The next sentence prediction fine-tuning task on intergenic regions began to gain loss at the 14th epoch and was stopped early. The maximum accuracy

observed was 91.429%. Both loss and accuracy curves for NSP fine-tuning can be found in Figure 4.4.

4.4.2 EnzymeGNN Training

The RevGNN with 20 layers performed better than the RevGNN with 40 layers. Performance plateaued around the 300th epoch. The maximum validation accuracy for the link prediction task observed was 85.236%. Validation loss and accuracy for the link prediction task across both architectures can be found in Figure 4.5

4.4.3 Unicycler Hyperparameter Optimisation

Hyperparameter optimisation with the Sweeps framework calculated the most important features in the performance of the FuzzyAlign score. The feature with the most impact was the probability cut-off used to determine if an EnzymeGNN link should be reported as positive. Other trends observed were lowering the EnzymeGNN-based bridging cut-offs and increasing the quality weight of EnzymeGNN bridges. The opposite trends were observed for the BacterialT5 bridges. Results are summarised in Table 4.1.

The relationship of the maximum FuzzyAlign metric with N50, ORF count and segment count can be found in Figure 4.6. Optimising for other metrics would result in

lower alignment accuracies. The maximum observed FuzzyAlign score after 783 iterations was 0.424 (Figure 4.7).

4.4.4 Comparison of different assembly strategies

Across the different strategies, the modified Unicycler pipeline with Bacterial T5 bridging resulted in the largest NA50, the smallest number of contigs, and the largest alignment. Biosynthetic spades capture the largest fraction of the genome but also had the most contigs with the lowest NA50. The plain Unicycler made the smallest number of misassembles while the modified Unicycler pipeline with Bacterial T5 bridging made the most.

To determine if the modified Unicycler pipeline was only making larger contigs because of misassemblies the metrics were recalculated using only assemblies containing no mistakes. In assemblies with no mistakes, Unicycler with BacterialT5 still outperforms the plain unicycler pipeline and BiosyntheticSPADES. With exclusively perfect assemblies, the combination of Unicycler with EnzymeGNN and Bacterial T5 bridging (NALA) resulted in the largest NA50. All metrics are summarised in Table 4.2.

4.4.5 Hybrid Assembly of Publicly Available Genomes

The NALA assemblies of the different bacterial strains result in larger contigs. An example of the bandage plots for DSM 22224 using the different strategies can be found in Figure 4.8. While Unicycler and SPADES maintain the circularisation of the de Bruijn graph, when the contigs are exported, they are shorter than those created by NALA. Unicycler and SPADES conserve more of the bacterial genome but result in more fragmented contigs. QUAST summary statistics of the NALA hybrid assemblies versus the original publicly available genomes are found in Table 4.3. In general larger N50s, larger contigs, fewer contigs and smaller genomes were observed for assemblies prepared by the NALA pipeline.

The effect the larger contigs had on genomic mining with PRISM 3 was summarised in Table 4.4. In general, we observed larger BGCs in the NALA assemblies. In the case of DSM 25239 and DSM 22224, we observed reduced fragmentation in the BGCs. An example of a highly fragmented NRPS/PK hybrid from DSM 22224 can be found in Figure 4.9. We also observed a recovery of PRISM 3 chemotyping for the highly fragmented DSM 40571 genome; an unknown thiotemplated BGC was reclassified as Type I Polyketide after NALA assembly.

4.4.6 Chymostatin Biosynthetic Gene Cluster Discovery

After mining the reassembled *Streptomyces orinoci* DSM 40571, the putative chymostatin gene cluster was annotated with adenylation domain substrates using IBIS (Figure 4.10). In comparison to the paper, 2 of the 3 predicted substrates were correct (phenylalanine and valine) [59]. Using the EnzymeBERT embeddings of the *Steptomyces mobaraensis* chymostatin gene cluster, euclidean distance nearest neighbour ORFs were found and summarised in Table 4.5. Besides the transcriptional regulator *cstA*, the remainder of the genes were found with a conserved gene order.

4.5 Discussion

4.5.1 Deep Learning Enhanced Assembly

We have demonstrated that deep learning models can be used to enhance bacterial genome assembly. Utilising information encoded on the genomic level and peptide level, we are able to create larger contigs and recover fragmented biosynthetic gene clusters. Our pipeline was able to outperform biosynthetic spades and unicycler in generating larger NA50s for our internal dataset of BGCs. The ability to robustly reassemble contigs with repetitive sequences was exemplified when our pipeline was able to rescue multiple thiotemplated BGCs across multiple genomes. In DSM 22224, NALA was able to pool the highly fragmented non-ribosomal BGCs into a single BGC resulting in the recovery of an additional condensation domain (Figure 4.10). In DSM 25239, the recovery of the contig provided enough information to enable PRISM to chemotype the Type I polyketide

that was previously annotated as an unknown thiotemplated BGC. NALA assemblies enable more robust genomic mining. In addition, we were able to locate the chymostatin BGC within our reassembled genome for *Streptomyces orinoci* DSM 40571. Within the *Streptomyces mobaraensis* genome, *cstB-G* span 13,820bp and within DSM 40571 they span 13,668 bp. Despite the misassemblies being made by the NALA pipeline, the spatial relationships for BGCs discovered within our reassemblies remain conserved.

While the NALA pipeline demonstrates remarkable recovery of information with short-read libraries alone, the mistakes introduced limit its propensity for superior performance. NA50s and average contig counts were optimally scored when no misassemblies were made. The major source of error in the NALA pipeline is BacterialT5. To maximise the size of contigs recovered, the Bacterial T5 model needs improvement. Upon closer inspection of misassemblies, relocation and inversion were the most common. Inversion is caused by BacterialT5 interpreting relationships observed on both strands. To compensate for this peculiarity, a gene direction-based correction step can be introduced. Relocations are rooted in a fundamental misunderstanding of inter-gene spatial relationships. While EnzymeGNN was trained using network graphs of ORFs to understand spatial arrangements, BacterialT5 was not; this was done purposefully as teaching a transformer to interpret multiple contigs at once would be difficult. To mitigate BacterialT5's naivety of intergenic relationships, a graph-based approach can be used similar to EnzymeGNN. Through the representation of contigs as nodes in a network

graph, a graph convolution network could be used to learn the conserved gene order relationships on a genomic level while still leveraging the resolution provided by BacterialT5.

4.5.2 Rescue of Illumina Short-Read Assemblies with Targeted Sequencing

In this work, we also demonstrated an approach for rescuing the RefSeq assemblies comprised of only Illumina short-reads. NALA is able to flesh out larger contigs without PacBio information as demonstrated in Figure 4.8. The usage of deep learning-based inference can enhance Illumina short-read assemblies. If longer contigs are still needed, we demonstrated that publicly available PacBio datasets can be assembled with publicly available Illumina datasets. Our hybrid assembly of DSM 40571 with exclusively publicly available datasets demonstrated there is a wealth of underutilised sequencing data. Microbes with multiple publicly available sequencing datasets should be subjected to reassembly to ensure the public has robust genomes available. Integration of our NALA pipeline can streamline this process. In addition, if no additional long-read data is publicly available, cheap PacBio sequencing can be performed strategically. The NALA pipeline was able to greatly enhance the quality of publicly available genomes using non-barcoded, multiplexed PacBio pools. In the case of DSM 19858, the genome went from 36 contigs to a perfect assembly.

4.5.3 Future work

We introduce completely novel deep-learning models in this work. BacterialT5 can be used as a base for a variety of bacterial genomic-based problems. EnzymeGNN is the first model we have seen that uses a transformer-based embedding to instantiate the node embeddings. This approach can be used to leverage the latent space of a transformer while utilising the network relationships captured by a graph convolutional network. The architecture of the EnzymeGNN framework can be used in areas where relationships between peptide sequences need to be modelled (e.g. protein-protein interaction learning and enhanced biosynthetic gene cluster calling). Through the deconstruction of the Unicycler pipeline and the new FuzzyAlign metric developed, hyperparameter optimisation of genomic assembly is readily accessible. We demonstrated a highly scalable method for adding new inference-based brides to Unicycler. By following the protocols proposed in this paper, more tooling can be added to the Unicycler pipeline enabling a holistic approach to bacterial genome assembly.

4.5 Figures and Tables

Figure 4.1 A stacked barplot showing the distribution of bacterial RefSeq genomes sequenced with a single platform, categorised by the submission year. There are thousands of legacy RefSeq genomes assembled with exclusively short-read Illumina sequencing technology. To compile the figure, RefSeq metadata was downloaded from the NCBI and manually cleaned using Tableau's Prep Builder. The cleaned dataset was plotted using Tableau Desktop.

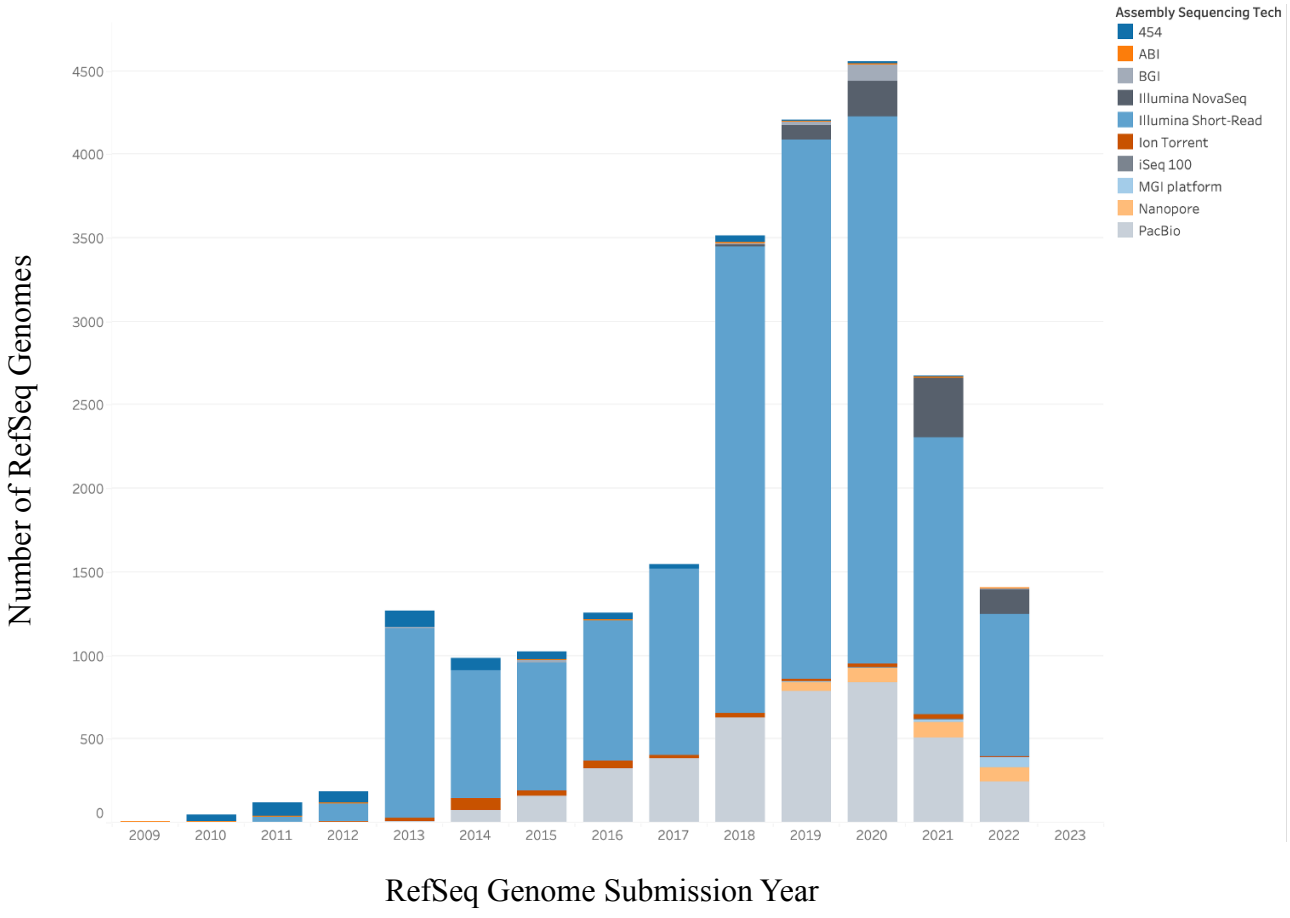


Figure 4.2 A stacked barplot showing the distribution of contig N50s for bacterial RefSeq genomes sequenced with a single platform. The majority of low N50 bacterial RefSeq genomes are those sequenced with exclusively Illumina short-read technology. To compile the figure, RefSeq metadata was downloaded from the NCBI and manually cleaned using Tableau's Prep Builder. The cleaned dataset was plotted using Tableau Desktop.

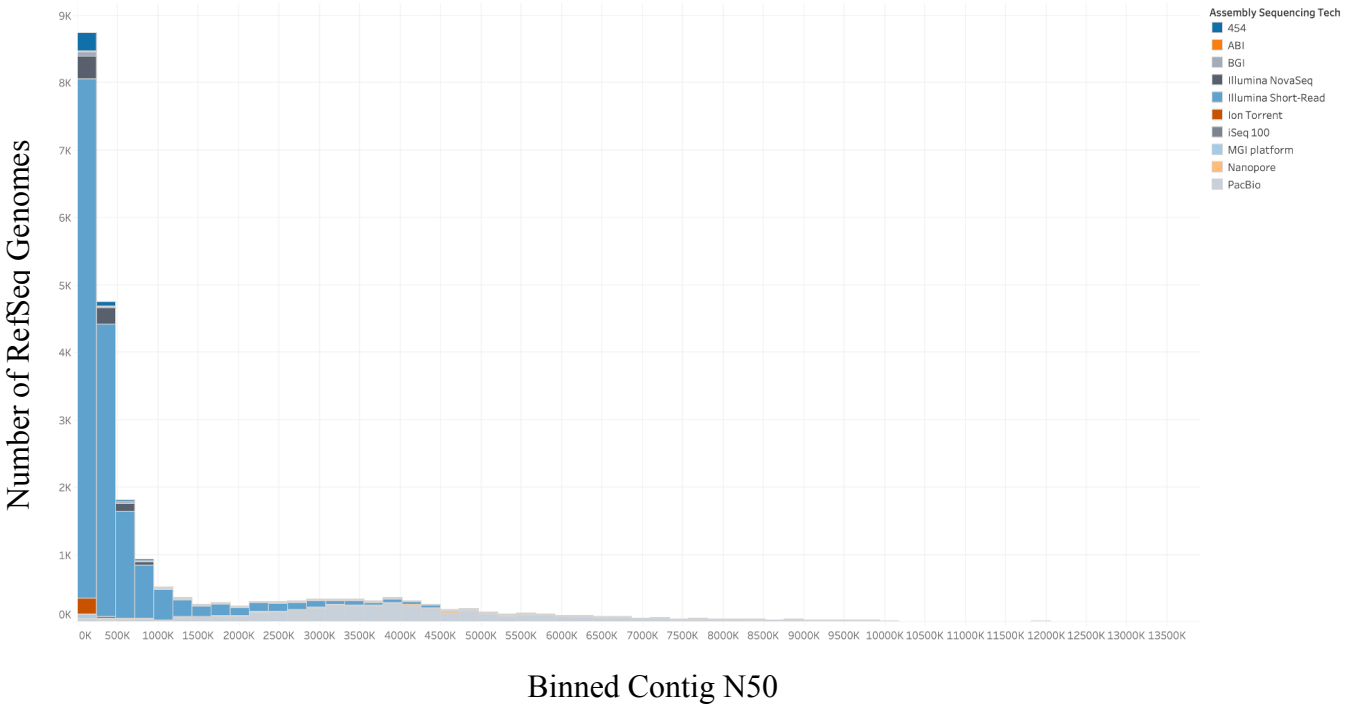
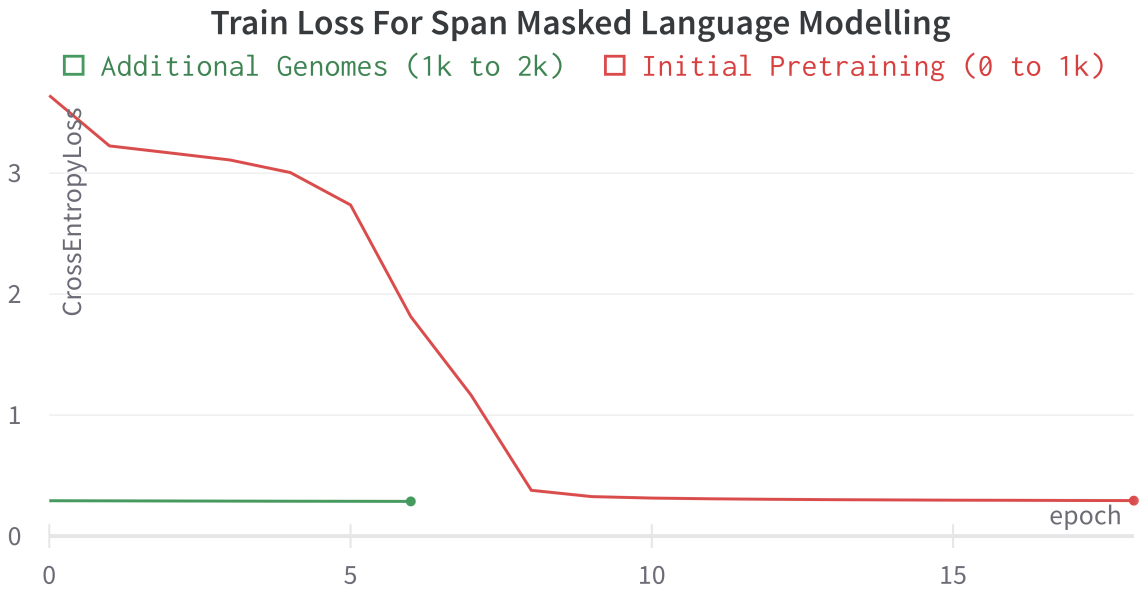


Figure 4.3 Training and validation loss curves for span masked language modelling of the BacterialT5 model. After 7 epochs, the loss had plateaued. A training session was started with a new dataset of genomes but was stopped early after no change loss was observed.



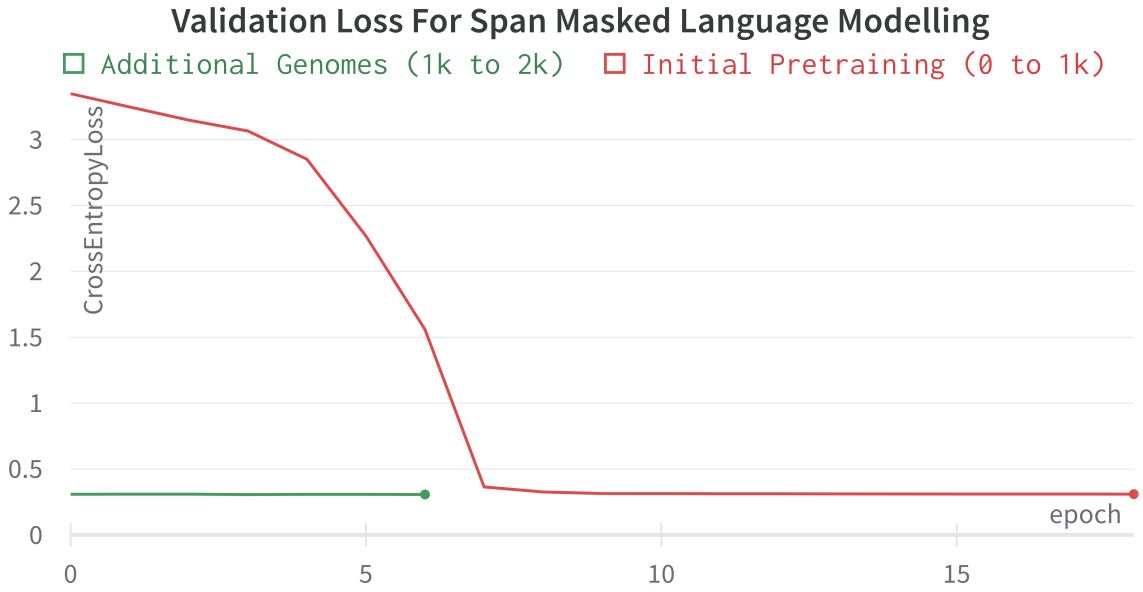
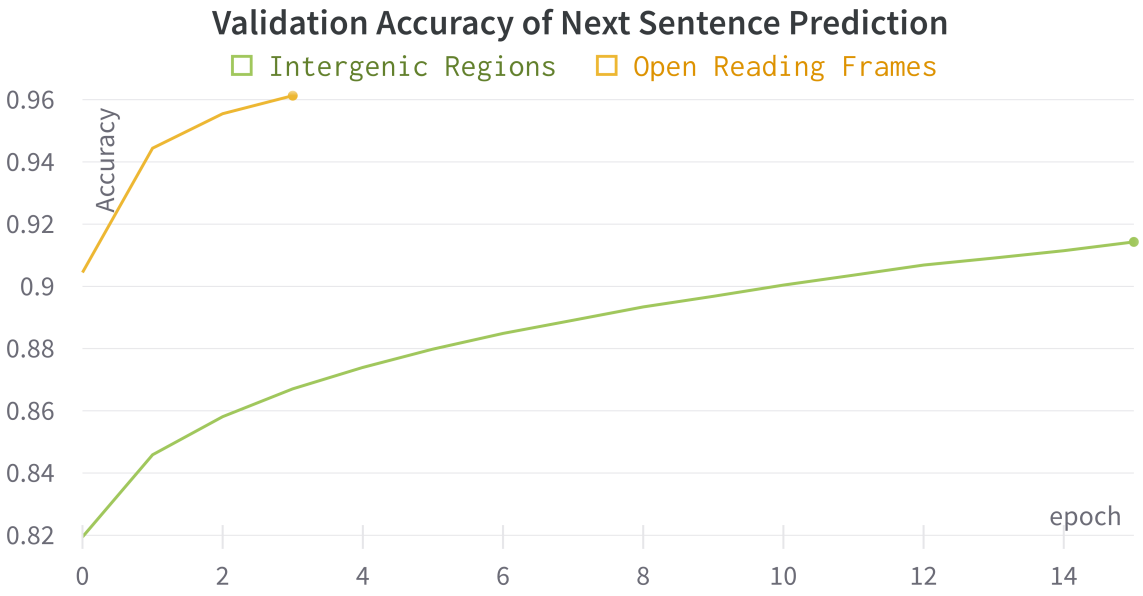


Figure 4.4 Validation accuracy and validation loss curves for the next sentence prediction tasks. The NSP on open reading frame data plateaued within 3 epochs. The NSP on intergenic regions began to gain loss at the 14th epoch and was stopped early.



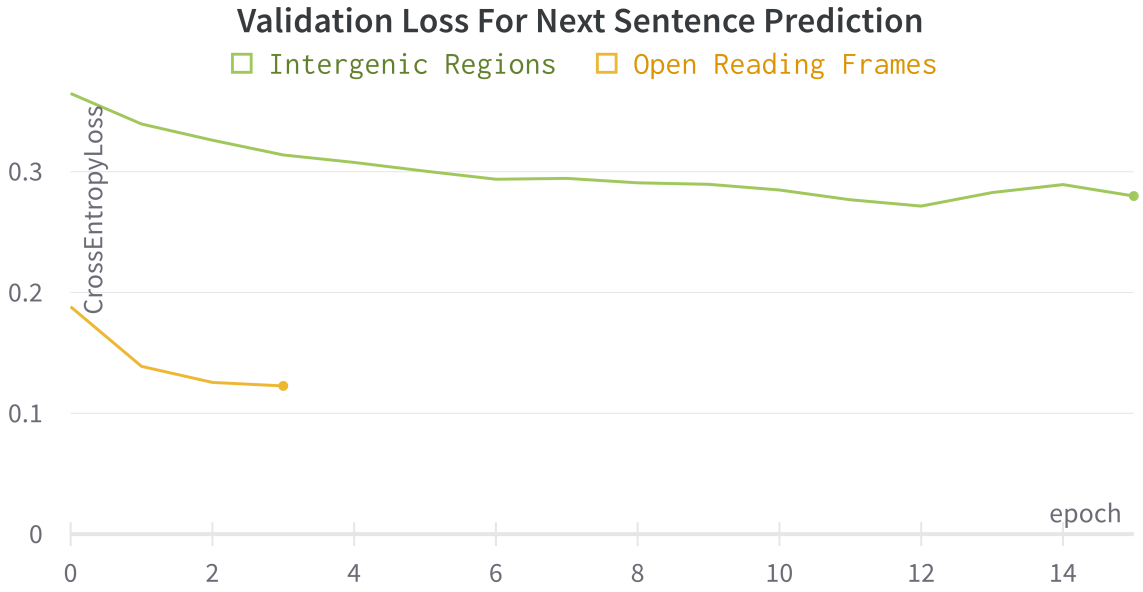


Figure 4.5 Validation Accuracy and Loss of the RevGNN model with 40 and 20 Layers. The maximum distance between NSP ORFs is 10,000 BP.

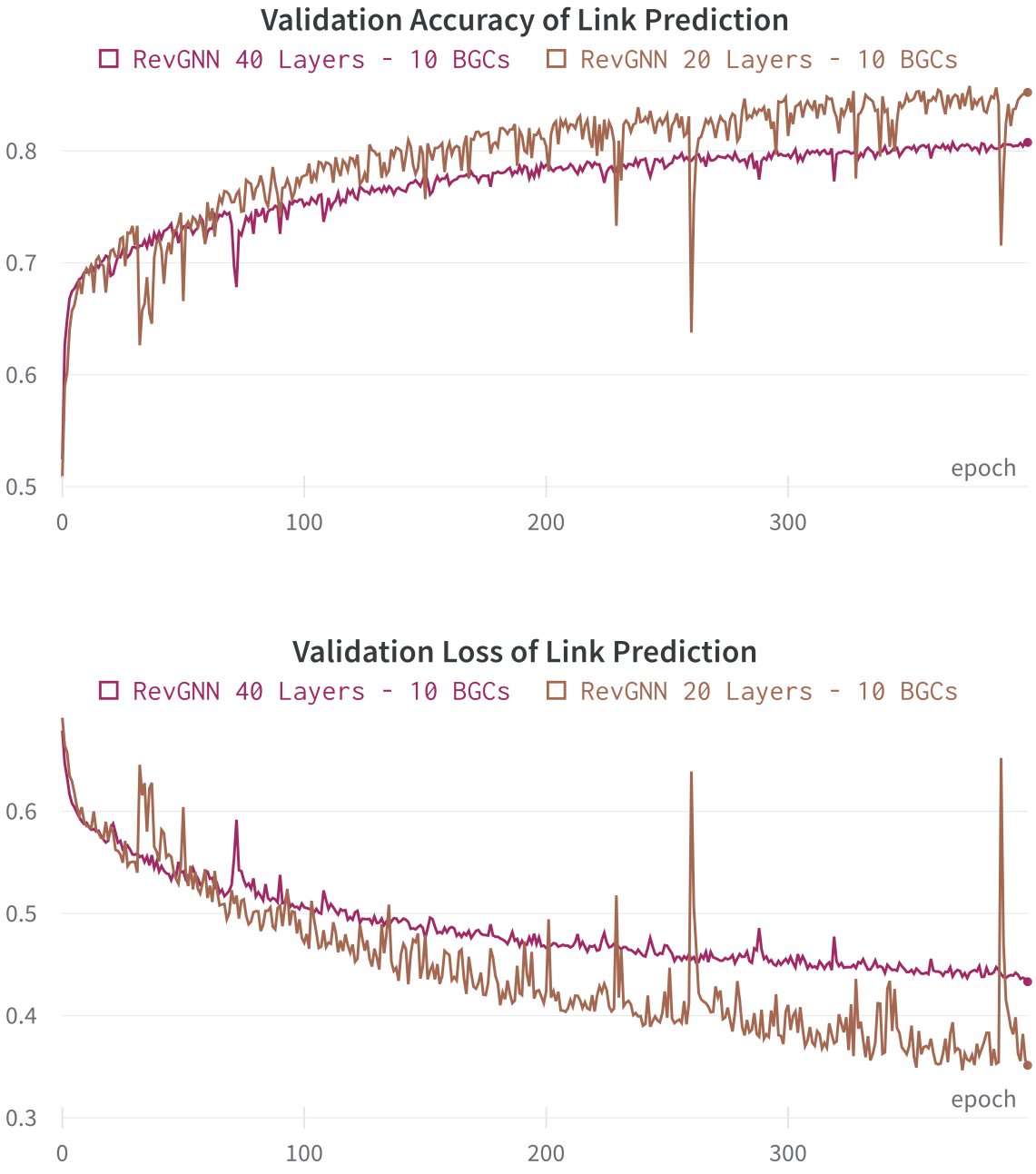


Table 4.1 Bayesian Hyperparameter Optimization result from sweeps. The most important factor for the FuzzyAlign scores was the minimum edge probability to declare whether or not a RevGNN predicted link was true. The higher the minimum link probability increased the accuracy of the bridge qualities and thereby increased the overall score. RevGNN bridges introduce the least amount of misassemblies; increasing the weight of the bridges and decreasing the cut-off would allow for more RevGNN bridging. As BacterialT5 bridging introduced the most misassemblies, it would be beneficial to decrease the weights and increase the cutoffs, as quantified by the correlation scores.

	Importance	Correlation
EnzymeBERT-GNN: Minimum Link Probability	0.844	0.474
EnzymeBERT-GNN: Bridge Quality Weight	0.052	0.168
BacterialT5: Bridge Quality Cut-Off	0.045	0.187
BacterialT5: Bridge Quality Weight	0.032	-0.162
EnzymeBERT-GNN: Bridge Quality Cut-Off	0.027	-0.023

Figure 4.6 A sankey graph showing the relationship between maximising for Mean Segment Count, N50, Open Reading Frame Count and FuzzyAlign. Maximising for metrics other than FuzzyAlign resulted in more misalignments.

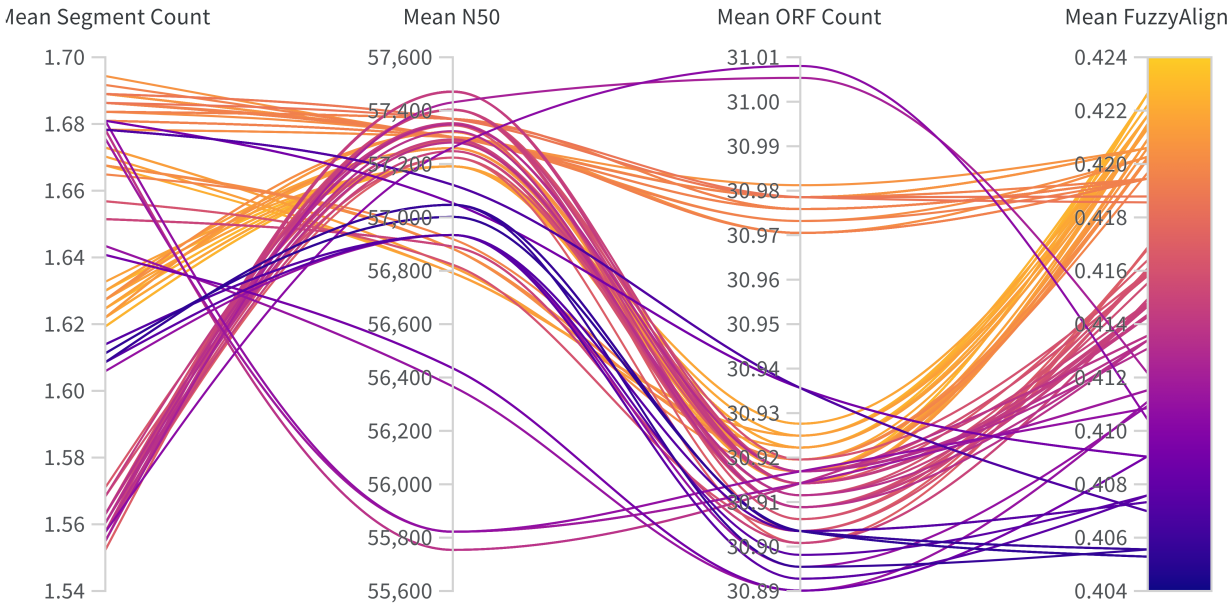


Figure 4.7 Bayesian optimisation of the bridge parameters to maximise the FuzzyAlign score. The maximum FuzzyAlign score observed was 0.424. The optimisation was stopped after 783 iterations.

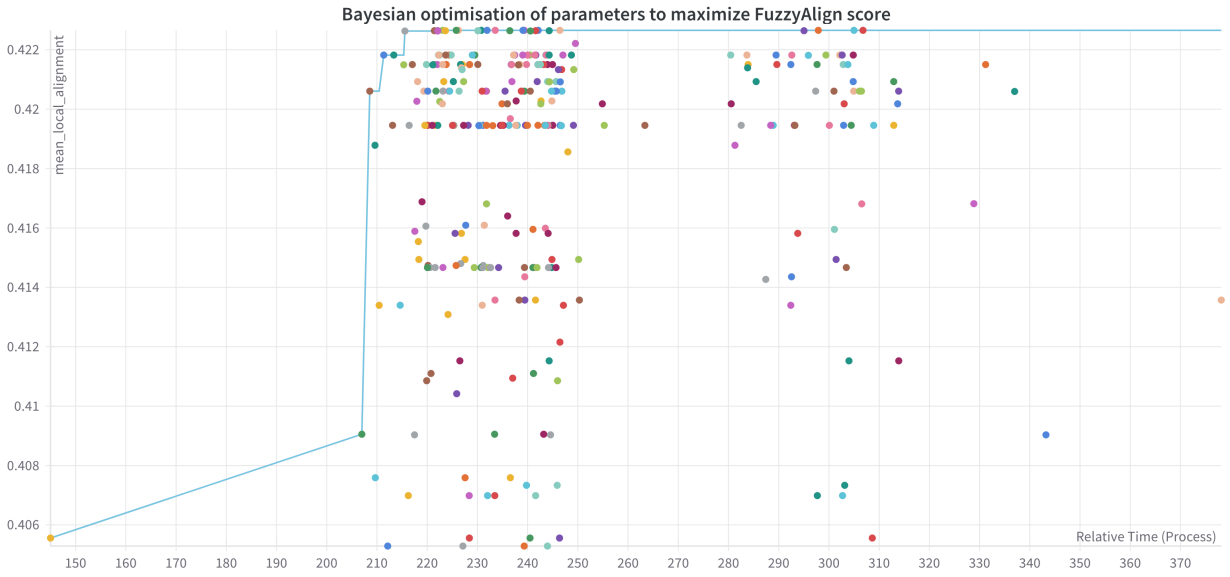


Table 4.2 Averaged assembly statistics of an in-silico Illumina MiSeq sequenced dataset of (n=521) Biosynthetic Gene Clusters with at least 10x coverage calculated using QUAST.

	BiosyntheticSPADES	Unicycler	Unicycler + GNN	Unicycler + Bacterial T5	Unicycler + GNN + Bacterial T5
Average NA50	47449	57087	56748	**57601**	57371
Average Number of Misassemblies	0.0475	**0.0089**	0.1840	0.5757	0.5163
Average Number of Contigs	2.944	2.388	1.807	**1.395**	1.525
Average Largest Alignment	49464	58074	57580	**58345**	58169
Average Genome Fraction (%)	**98.279**	93.476	92.741	92.549	92.677
			Metrics Calculated with No Misassemblies Dataset		
Average NA50	47469	57148	60716	64345	**65720**
Average Number of Contigs	2.833	2.334	1.528	**1.180**	1.204

Figure 4.8 Bandage Plots for various assembly methods on DSM 2224. (A) The optimal SPADES assembly was created using a k-mer of 127 (B) The unicycler hybrid assembly used PacBio reads and Illumina short-reads alone. (C) NALA pipeline with only Illumina short-reads (D) NALA pipeline with PacBio long reads and Illumina short-reads.

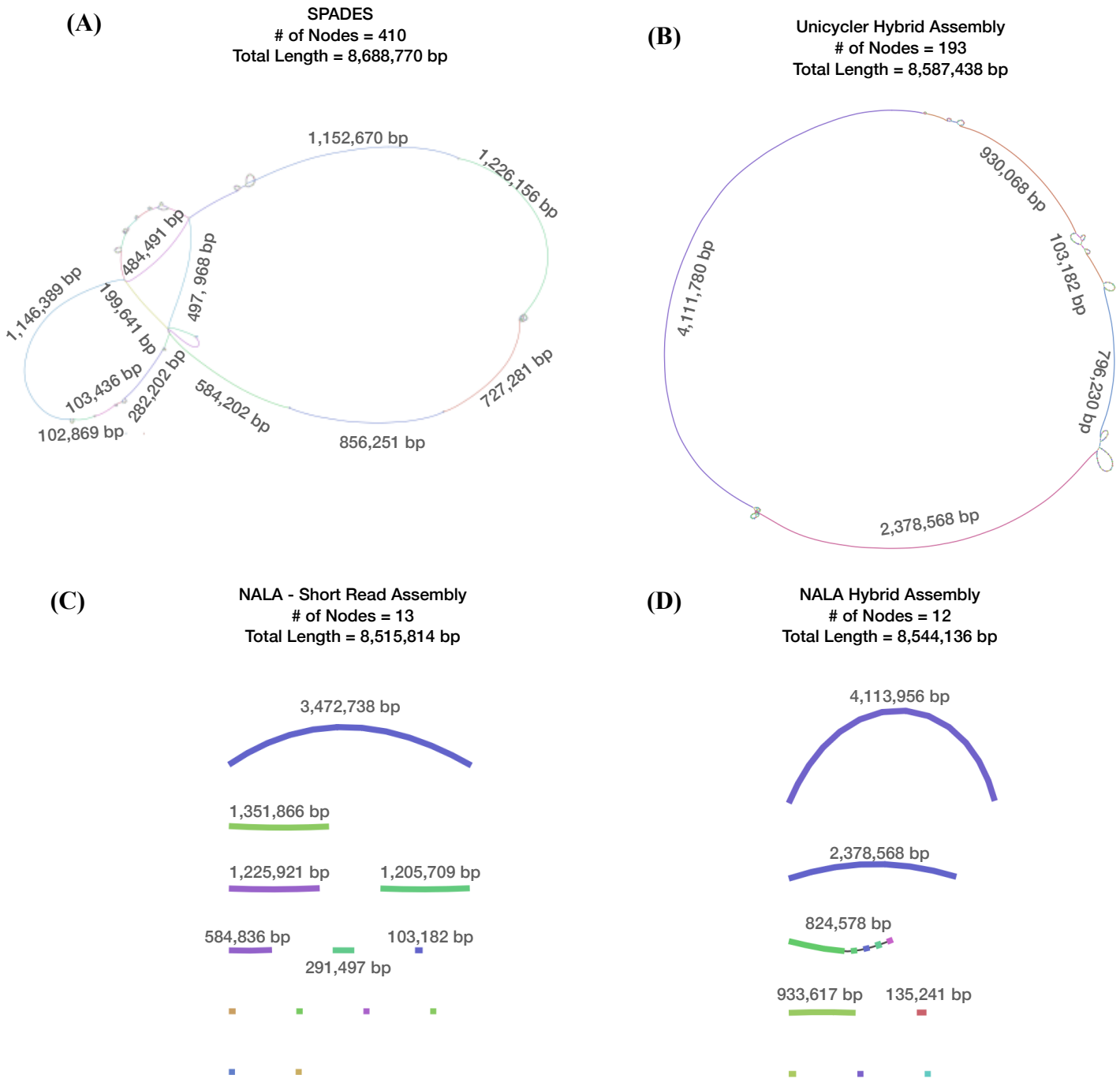


Table 4.3 QUAST assembly statistics of the hybrid reassemblies using the deep-learning guided assembly pipeline versus the original publicly available genome.

	DSM 19858		**DSM 22224**		**DSM 25239**		**DSM 40571**	
	NALA	Original	NALA	Original	NALA	Original	NALA	Original
# of contigs	**1**	36	**8**	30	**47**	180	**13**	44
Largest contig	**5066218**	1274834	**4113956**	2381325	**881154**	200475	**3155585**	1759048
Total length	**5066218**	5036817	8544136	**8552642**	7339869	**7356379**	7359118	**7502208**
GC (%)	47.11	47.08	50.44	50.44	60.84	60.75	70.89	70.79
N50	**5066218**	475669	**2378568**	2168259	**276498**	78320	**1040784**	434919

Table 4.4 Gene Cluster Assembly Statistics Generated Using PRISM for Publicly Available Genomes versus New Genome.

	DSM 19858		**DSM 22224**		**DSM 25239**		**DSM 40571**	
	NALA	Original	NALA	Original	NALA	Original	NALA	Original
Number of BGCs	2	2	**21**	27	**1**	2	23	23
Average BGC Length	20969.00	20969.00	**24298.19**	18516.93	**40073.00**	23750.50	**26110.13**	25936.13
Butyrolactone	-	-	-	-	-	-	1	1
Non-Ribosomal Peptides	1	1	12	18	-	-	5	5
NRP-Type I PK Hybrid	-	-	4	4	1	1	3	3
NIS-Synthase	-	-	1	1	-	-	1	1
Resorcinol	1	1	-	-	-	-	-	-
Ribosomal	-	-	3	3	-	-	3	3
Terpene	-	-	1	1	-	-	-	-
Type I Polyketide	-	-	-	-	-	1	**8**	7
Type II Polyketide	-	-	-	-	-	-	1	1
Unknown	-	-	-	-	-	-	**1**	2

Figure 4.9 The recovered NRP-PK Hybrid biosynthetic cluster from DSM 2224 using the deep learning guided assembly. The fragmented gene clusters from the hybrid assembly alone are highlighted by their corresponding colours.

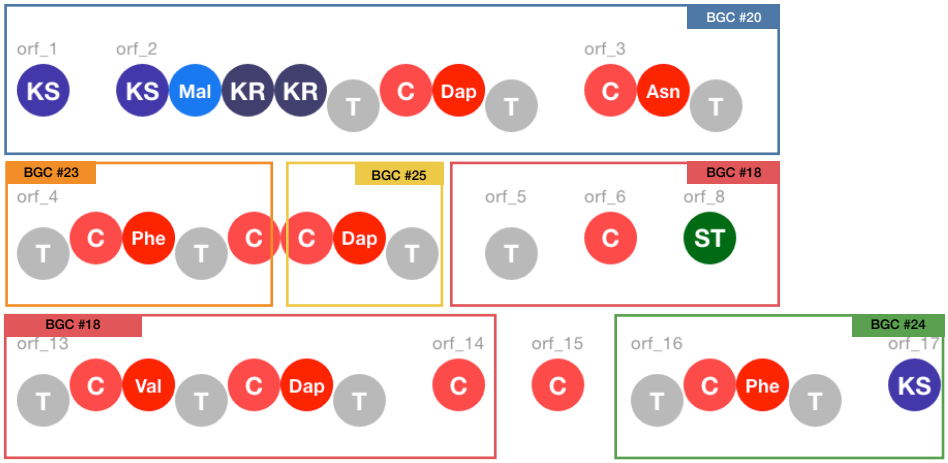


Figure 4.10 The Chymostatin BGC discovered within *Streptomyces orinoci* DSM 40571 (contig 3 at 180,440 to 306,289). Using IBIS-LLMs, the adenylation domain substrates were predicted. Phenylalanine and Valine were both correctly predicted by the Adenylation T5 model.

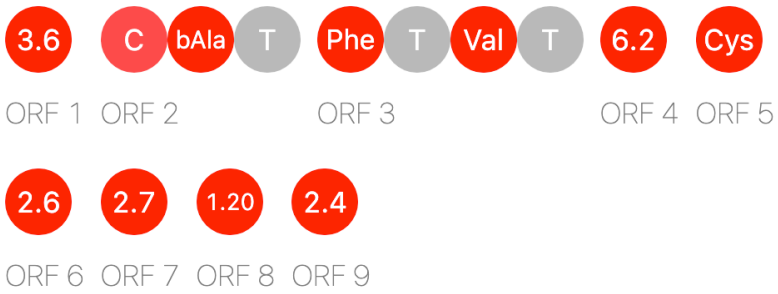


Table 4.5 Chymostatin biosynthetic gene cluster found in *Streptomyces mobaraensis* and related peptide sequences discovered in *Streptomyces orinoci* DSM 40571 using EnzymeBERT euclidean distances. The only gene not found was the transcriptional regulator *cstA*. The relative sizes of the gene clusters were conserved with *cstB-G* spanning 13,820bp in the source genome versus 13,667 in DSM 40571.

	Steptomyces mobaraensis				*Streptomyces orinoci* DSM 40571			
Gene	UniProt code	Protein Length (aa)	Start	End	Protein Length (aa)	Start	End	$d_{EnzymeBERT}$
cstA	M3BF97	169	18103	18612	-	-	-	-
cstB	M3C2P0	416	18942	20192	421	292622	293888	5.73
cstC	M3C2S2	422	20262	21530	427	293960	295244	37.23
cstD	M3AXD2	1035	21674	24781	1045	295323	298461	5.50
cstE	M2ZZP2	385	24778	25935	385	298457	299615	2.11
cstF	M3BFA2	1387	25990	30153	1326	299669	303650	22.01
cstG	M3C2P6	870	30150	32762	880	303646	306289	22.66

4.6 Bibliography

1. Li, W., et al., RefSeq: expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation. Nucleic Acids Research, 2021. **49**(D1): p. D1020-D1028.
2. Rhoads, A. and K.F. Au, *PacBio Sequencing and Its Applications.* Genomics, Proteomics & Bioinformatics, 2015. **13**(5): p. 278-289.
3. Jain, M., et al., The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. Genome Biology, 2016. **17**(1): p. 239.
4. Li, H., Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics, 2016. **32**(14): p. 2103-2110.
5. Koren, S., et al., Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Research, 2017. **27**(5): p. 722-736.
6. Kolmogorov, M., et al., Assembly of long, error-prone reads using repeat graphs. Nature Biotechnology, 2019. **37**(5): p. 540-546.
7. Ruan, J. and H. Li, Fast and accurate long-read assembly with wtdbg2. Nature methods, 2020. **17**(2): p. 155-158.
8. Vaser, R. and M. Šikić, Time- and memory-efficient genome assembly with Raven. Nature Computational Science, 2021. **1**(5): p. 332-336.
9. Quail, M.A., et al., A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. BMC Genomics, 2012. **13**(1): p. 341.
10. Fichot, E.B. and R.S. Norman, Microbial phylogenetic profiling with the Pacific Biosciences sequencing platform. Microbiome, 2013. **1**(1): p. 10.
11. Miller, J.R., et al., Hybrid assembly with long and short reads improves discovery of gene family expansions. BMC Genomics, 2017. **18**(1): p. 541.
12. Wick, R.R., et al., Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. PLOS Computational Biology, 2017. **13**(6): p. e1005595.
13. Compeau, P.E.C., P.A. Pevzner, and G. Tesler, Why are de Bruijn graphs useful for genome assembly? Nature biotechnology, 2011. **29**(11): p. 987-991.
14. Bankevich, A., et al., SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. Journal of Computational Biology, 2012. **19**(5): p. 455-477.
15. Walker, B.J., et al., Pilon: An Integrated Tool for Comprehensive Microbial Variant Detection and Genome Assembly Improvement. PLOS ONE, 2014. **9**(11): p. e112963.
16. Chen, Z., D.L. Erickson, and J. Meng, Benchmarking hybrid assembly approaches for genomic analyses of bacterial pathogens using Illumina and Oxford Nanopore sequencing. BMC Genomics, 2020. **21**(1): p. 631.

17. Schneeberger, K., et al., Reference-guided assembly of four diverse Arabidopsis thaliana genomes. Proceedings of the National Academy of Sciences, 2011. **108**(25): p. 10249-10254.
18. Vezzi, F., F. Cattonaro, and A. Policriti, e-RGA: enhanced Reference Guided Assembly of Complex Genomes. EMBnet.journal, 2011. **17**(1): p. 46-54.
19. McAdam, P.R., et al., Molecular tracing of the emergence, adaptation, and transmission of hospital-associated methicillin-resistant Staphylococcus aureus. Proceedings of the National Academy of Sciences, 2012. **109**(23): p. 9107-9112.
20. Bao, E., T. Jiang, and T. Girke, AlignGraph: algorithm for secondary de novo genome assembly guided by closely related references. Bioinformatics, 2014. **30**(12): p. i319-i328.
21. Mentasti, M., et al., Rapid detection and evolutionary analysis of Legionella pneumophila serogroup 1 sequence type 47. Clinical Microbiology and Infection, 2017. **23**(4): p. 264.e1-264.e9.
22. Pérez-Losada, M., M. Arenas, and E. Castro-Nallar, *Microbial sequence typing in the genomic era.* Infection, Genetics and Evolution, 2018. **63**: p. 346-359.
23. Ellington, M.J., et al., Contrasting patterns of longitudinal population dynamics and antimicrobial resistance mechanisms in two priority bacterial pathogens over 7 years in a single center. Genome Biology, 2019. **20**(1): p. 184.
24. Landan, G. and D. Graur, Characterization of pairwise and multiple sequence alignment errors. Gene, 2009. **441**(1): p. 141-147.
25. Alkan, C., S. Sajjadian, and E.E. Eichler, Limitations of next-generation genome sequence assembly. Nature Methods, 2011. **8**(1): p. 61-65.
26. Farrer, R.A., et al., Using False Discovery Rates to Benchmark SNP-callers in next-generation sequencing projects. Scientific Reports, 2013. **3**(1): p. 1512.
27. Hurgobin, B. and D. Edwards, SNP Discovery Using a Pangenome: Has the Single Reference Approach Become Obsolete? Biology, 2017. **6**(1): p. 21.
28. Wang, B., et al., Whole genome sequencing of the black grouse (Tetrao tetrix): reference guided assembly suggests faster-Z and MHC evolution. BMC Genomics, 2014. **15**(1): p. 180.
29. Lischer, H.E.L. and K.K. Shimizu, Reference-guided de novo assembly approach improves genome reconstruction for related species. BMC Bioinformatics, 2017. **18**(1): p. 474.
30. Lee, R.S. and M.A. Behr, Does Choice Matter? Reference-Based Alignment for Molecular Epidemiology of Tuberculosis. Journal of Clinical Microbiology, 2016. **54**(7): p. 1891-1895.
31. Usongo, V., et al., Impact of the choice of reference genome on the ability of the core genome SNV methodology to distinguish strains of Salmonella enterica serovar Heidelberg. PLOS ONE, 2018. **13**(2): p. e0192233.

32. Bush, S.J., et al., Genomic diversity affects the accuracy of bacterial single-nucleotide polymorphism–calling pipelines. GigaScience, 2020. **9**(2): p. giaa007.
33. Ji, Y., et al., DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. Bioinformatics, 2021. **37**(15): p. 2112-2120.
34. Taboada, B., C. Verde, and E. Merino, High accuracy operon prediction method based on STRING database scores. Nucleic Acids Research, 2010. **38**(12): p. e130.
35. Szklarczyk, D., et al., The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Research, 2021. **49**(D1): p. D605-D612.
36. Elnaggar, A., et al., ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. arXiv:2007.06225 [cs, stat], 2021.
37. Wolf, T., et al., HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv:1910.03771 [cs], 2020.
38. Devlin, J., et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs], 2019.
39. Raffel, C., et al., Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs, stat], 2020.
40. Falcon, W., *Pytorch lightning*. GitHub. Note: https://github. com/ PyTorchLightning/pytorch-lightning, 2019. **3**: p. 6.
41. Rasley, J., et al. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. 2020. Association for Computing Machinery.
42. Rajbhandari, S., et al., ZeRO-Infinity: Breaking the GPU Memory Wall for Extreme Scale Deep Learning. arXiv:2104.07857 [cs], 2021.
43. Baker, D.N. and B. Langmead, Dashing: fast and accurate genomic distances with HyperLogLog. Genome Biology, 2019. **20**(1): p. 265.
44. Larralde, M., Pyrodigal: Python bindings and interface to Prodigal, an efficient method for gene prediction in prokaryotes. Journal of Open Source Software, 2022. **7**(72): p. 4296.
45. Li, G., et al., Training Graph Neural Networks with 1000 Layers. 2022, arXiv.
46. Fey, M. and J.E. Lenssen, Fast Graph Representation Learning with PyTorch Geometric. 2019, arXiv.
47. Kautsar, S.A., et al., MIBiG 2.0: a repository for biosynthetic gene clusters of known function. Nucleic Acids Research, 2020. **48**(D1): p. D454-D458.
48. Passino, F.S., et al., Link prediction in dynamic networks using random dot product graphs. Data Mining and Knowledge Discovery, 2021. **35**(5): p. 2168-2199.
49. Dettmers, T., et al., 8-bit Optimizers via Block-wise Quantization. 2022, arXiv.

50. Gourlé, H., et al., Simulating Illumina metagenomic data with InSilicoSeq. Bioinformatics, 2019. **35**(3): p. 521-522.
51. Levenshtein, V.I., Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, 1966. **10**(8): p. 707-710.
52. Kuhn, H.W., The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 1955. **2**(1-2): p. 83-97.
53. Virtanen, P., et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 2020. **17**: p. 261-272.
54. Biewald, L., Experiment Tracking with Weights and Biases. 2020.
55. Meleshko, D., et al., BiosyntheticSPAdes: reconstructing biosynthetic gene clusters from assembly graphs. Genome Research, 2019. **29**(8): p. 1352-1362.
56. Derakhshani, H., et al., Completion of draft bacterial genomes by long-read sequencing of synthetic genomic pools. BMC Genomics, 2020. **21**(1): p. 519.
57. Gurevich, A., et al., QUAST: quality assessment tool for genome assemblies. Bioinformatics, 2013. **29**(8): p. 1072-1075.
58. Skinnider, M.A., et al., PRISM 3: expanded prediction of natural product chemical structures from microbial genomes. Nucleic Acids Research, 2017. **45**(W1): p. W49-W54.
59. Juettner, N.E., et al., Decoding the Papain Inhibitor from Streptomyces mobaraensis as Being Hydroxylated Chymostatin Derivatives: Purification, Structure Analysis, and Putative Biosynthetic Pathway. Journal of Natural Products, 2020. **83**(10): p. 2983-2995.

Chapter 5: Prediction of Metabolite Activity

5.1 Chapter Preface

Our lab was engaged in a joint collaboration to find new human microbiome metabolites. We worked with Dr Michael Surette's lab to mine out human microbiome BGCs using culture-enriched metagenomes to guide isolation efforts. Nishanth Merwin performed all genomic mining of the metagenomes using PRISM 3. There were thousands of novel BGCs but no criteria on which BGCs to select for. During the time of the collaboration, a similar initiative was taking place with the NIH's integrated Human Microbiome Project (iHMP). I developed a pipeline to leverage the iHMP data to deduce which of the BGCs would be active.

I developed the metatranscriptomic pipeline based on advice I had received from colleagues at an NCBI-sponsored hackathon. I developed the method for mass-matching microbial metabolites based on advice from Dr Haoxin Li. I developed the proteomics matching pipeline based on standard protocols for peptide identification. I developed the statistical protocols for data fusion, linear regression modelling and gene ontology enrichment from advice provided by Dr Benjamin Haibe-Kains.

Developing the deep learning model for the discovery of the bacteriocins was a collaborative effort. I designed the deep learning framework for multi-task training of the ProtT5 transformer and all custom fine-tuning heads. The NLPPrecursor dataset was

curated by Nishanth Merwin. The bacteriocin dataset was a joint curation effort between myself, Nishanth Merwin, Bilal Athar, Dr Walaa Mousa, and Mathusan Gunabalasingam. The putative bacteriocin from Scardovia was initially mined out using an HMM-based protocol. HMMs were built using a workflow I created in collaboration with Nishanth Merwin. Upon the development of my LLM-based approach to peptide comparison, I replaced the HMM workflow with the vector database approach described.

5.2 Abstract

Human microbiota have demonstrated profound effects on the health of their hosts. To study the host-microbial relationship, a variety of integrated multi-omic datasets have been released publicly. We present a pipeline to disentangle the complex relationships between different -omics data using statistics. Our pipeline, MANGO (Microbiome Associated Natural-product Gene Ontology) can be used to predict the gene signature of a microbial metabolite using the host-transcriptomic data and heuristic measures for the molecule of interest. Using metabolomics, proteomics and metatranscriptomics we demonstrate three separate approaches for modelling the host-metabolite relationship.

5.3. Introduction

Encoded microbial metabolites can be readily discovered using genomic mining platforms such as PRISM and AntiSMASH. [1, 2] Microbial metabolites have a wide variety of therapeutically relevant properties such as antimalarial, anticancer, and

antiviral.[3] It is of great interest to screen encoded metabolites for activity before isolating the compound. Typically, the prediction of a molecule's activity would be performed using Quantitative Structure-Activity Relationship (QSAR) approaches. QSAR frameworks are heavily dependent on the molecular structures' accuracy.[4-6] While both genomic mining platforms can produce probable scaffolds, a large combinatoric space is required due to the limited predictability of tailoring enzymes.[1, 7] It is not feasible to run every permuted structure through a QSAR framework.

PRISM 4 instead approaches activity prediction by using molecular fingerprinting techniques to find related active compounds.[8] The approach is limited to structures PRISM has been developed to characterise and is highly dependent on the accuracy of the predicted structure. [7] Moving away from chemical structures, Walker and Clardy propose an approach using a machine learning classifier trained on PFAM annotations of BGCs.[9] In the same vein, DeepBGC used a deep learning model to embed BGC enzymes and train a classifier to predict antibiotic activity.[10] While both strategies are promising, the lack of BGCs with annotated activities limits the practicality and scalability of both approaches.

The human microbiome is a unique environment where the synthesis of microbial metabolites inherently affects the host's phenotype. The origins of some human diseases are attributed to dysbiosis of the gut.[11] One of the known mechanisms by which these

bugs are affecting our health is via secondary metabolites such as butyrate.[12] Recently a multi-omics dataset was released for the human microbiome.[13] The Integrative Human Microbiome Project (iHMP) is a longitudinal study carried out over ten years with comprehensive data catalogued for the human microbiome. There are various sources of data that were catalogued including disease state, inflammation measures, host transcriptomics, gut metatranscriptomes, gut metagenomes and more. The dataset allows for a comprehensive approach to studying the interaction of human microbiota with the host.

In this work, we introduce a statistical pipeline for detangling and integrating the multi-omics datasets from iHMP to perform in-situ activity prediction of microbial metabolites. Our pipeline named MANGO (Microbiome Associated Natural-product Gene Ontology) uses heuristic measures of encoded metabolites and host transcriptomic modulations to decode potential microbial metabolite activities (Figure 5.1). We demonstrate three techniques for integrating multi-omics data with host transcriptomics: (1) Quantifying biosynthetic expression using gut metatranscriptomic data, (2) quantifying metabolites using mass matching in gut metabolomic data, and (3) quantifying bacteriocins using gut proteomics data. Our approach focuses on the dynamic relationship of microbial metabolism and its influence on the host's transcriptome.

5.4 Methodology

5.4.1 Mining the Human Gut Metatranscriptome for Biosynthetic Gene Clusters

Typically human microbiome metagenomes are not useful for genomic mining. Previously metagenomes from the Human Microbiome Project were mined but only eight thiopeptide clusters were found.[14] Instead, we used four metagenomes generated from human faecal samples using culture enrichment. [15] Three of the samples were carried in mice before sequencing; the change in host resulted in less microbiota diversity. The four metagenomes were then run through PRISM 3 to find all biosynthetic gene clusters (BGCs). [1] Mined gene clusters were then quantified in terms of expression using metatranscriptomic datasets. To use the BGC expression as a heuristic approximation of metabolite production, we made the assumption only ORFs within a gene cluster containing biosynthetic domains would be necessary (i.e. immunity and regulator gene expression may not be representative of the amount of metabolite produced). With this in mind, only biosynthetic transcripts were quantified.

All non-redundant biosynthetic open reading frames (ORFs) were compiled into a template file for transcript quantification (n=1,188). Microbial transcript quantification was calculated using the quasi-mapping software SALMON.[16] Quasi-mapping is much faster and sometimes more accurate than traditional alignment. For the multi-omics analysis, the metatranscriptomic samples (sequenced from human faecal matter) with

corresponding human transcriptomic samples (sequenced from bowel biopsy) were downloaded from IBDMDB (n=737).[17] Quantification was reported in Transcripts Per Million (TPM). An algorithm for cross-sample normalization was found in the R package SLEUTH.[18] It was reimplemented in Python and was used to normalise the results from SALMON. BGCs expressed in the IBDMDB dataset were selected for downstream analysis with MANGO. Results are found in Section 5.5.1.

5.4.2 Mining the Human Gut Metabolome for Microbial Metabolites

Faecalibacterium prausnitzii is the most abundant bacterium in the human intestinal microbiota of healthy adults. It represents more than 5% of the total bacterial population. It is associated with improving gut health through its anti-inflammatory properties.[19, 20] We isolated a novel compound from *Faecalibacterium prausnitzii* (Figure 5.2). Its structure is related to butyrate and from our in-house testing, demonstrated anti-inflammatory activity.[21] To see if the *Faecalibacterium prausnitzii* metabolite is implicated in gut health, we mined out the abundance of the metabolite across publicly available data downloaded from IBDMDB.[17] Only gut metabolomics data (extracted from human faecal matter) with corresponding overlapping human transcriptomic samples (sequenced from a bowel biopsy) were used. The mass spectrometry results were downloaded in RAW format. Using Proteowizard, the spectra were processed and converted to mzML format.[22] All mzML files were parsed using

the XML package in Python. The perform mass matching, the monoisotopic mass for the *Faecalibacterium prausnitzii* compound, and the masses for additional adducts were calculated. The samples were run on an Orbitrap mass spectrometer; the mass measurement accuracy (MMA) according to the manufacturer's specification is 1-5 ppm. [23] With regard to this threshold, we mass-matched the compound and adducts with an error threshold of 5ppm across the metabolomic datasets. The relative abundance values of matching peaks were selected for downstream analysis with MANGO. Results are found in Section 5.5.2.

5.4.3 Linear Regression of *Faecalibacterium prausnitzii* compound and SSCAI

Upon metabolite matching, it was observed that the *Faecalibacterium prausnitzii* compound was in higher numbers within healthy patients versus ulcerative colitis patients (Figure 5.3). A linear model was fit to determine if there was a linear relationship between the *Faecalibacterium prausnitzii* compound and a decrease in the symptomology of UC. The Simple Clinical Colitis Activity Index (SSCAI) is a measure of the severity of symptoms in patients with Ulcerative Colitis (UC).[24] For the linear regression model, the independent variable was the relative abundance of peaks matching the *Faecalibacterium prausnitzii* compound and the dependent variable was the SSCAI. Results are found in Section 5.5.3.

5.4.4 Mining the Human Gut Proteome for Bacteriocins

To find the abundance of bacteriocins within the human gut proteome, we first needed to create a dataset of human microbiome bacteriocin peptide sequences. We compiled a dataset of microbes isolated from the human microbiome and mined out putative bacteriocin peptides. After the sequences were mined out, a proteomics analysis pipeline was used for quantification within the IBDMDB dataset.

5.4.4.1 Mining bacteriocin sequences from human microbiome associated genomes

We designed a pipeline for mining bioactive peptide compounds from microbial genomes using a transformer trained on our NLP-Precursor dataset. NLP-Precursor was a software designed to classify input peptide sequences as a class of Ribosomally encoded Post-translationally modified Peptide (RiPP) and predict the locations that will be trimmed within the propeptide.[25] As a base transformer, we selected the encoder from the ProtT5 model.[26] Two custom fine-tuning heads were made to train the ProtT5 model on RiPP classification. For token classification, a custom fine-tuning head was made with an integrated conditional random field layer (CRF). To fine-tune sequence classification, a modified RoBERTa head was used. The sequence classification head was used for whole peptide sequence RiPP classification. The token classification head was used to classify individual residues for trimming. Tokenisation used the stock ProtT5 tokenizer. Multiple training strategies were implemented to maximise performance: (1)

Both tasks were trained separately, (2) Both tasks were trained together and (3) The model was first fine-tuned for the RiPP classification task, and then multi-task fine-tuned for RiPP and residue classification. Results for training can be found in Section 5.5.4.

To find putative bacteriocin sequences, a vector database of reference bacteriocin embeddings was created. To create this, we first hand-curated a dataset of bacteriocins from the literature. Each bacteriocin peptide sequence was then embedded using the fine-tuned transformer. The embeddings were stored in a vector database and indexed with a flat inverted file index, optimised for Euclidean distance searches. Using Pyrodigal, peptide sequences were predicted from the human microbiome strain *Scardovia wiggsiae* F0424.[27] Using the vector database of bacteriocins, we found a peptide sequence highly related to the bacteriocin class II aureocin-like peptides (Section 5.5.5). [28, 29] The putative bacteriocin peptide sequence was synthesised and screened for activity. It demonstrated anti-inflammatory and anti-microbial activity (Section 5.5.6). We chose this peptide to explore proteomics abundance.

5.4.4.2 Mining *Scardovia* Bacteriocin Abundencies from Human Microbiome

Proteomics samples (extracted from human faecal matter) with corresponding human transcriptomic data (extracted from human bowel biopsies) were downloaded from IBDMDB.[17] The mass spectrometry results were downloaded in RAW format.

Using Proteowizard, the spectra were centroided and converted to mzML format.[22] MS-GF+ was used with a decoy library to detect peptide matches.[30] The calculated relative abundances of matched peaks were selected for downstream analysis with MANGO. Results can be found in Section 5.5.6.

5.4.5 Mango Processing of Samples

The MANGO statistical pipeline starts with fitting a series of linear regression models. Input abundances (measured through mass matching, proteomics, or metatranscriptomics) are treated as the independent variable. The abundance of each human gene's expression (found in the host-transcriptomic data) is treated as the dependent variable. The relationship tested is whether or not the measures of microbial metabolite are linearly related to the host system's modulation. In cases where there are multiple quantities of the metabolite to be tested (e.g. multiple metabolomic peaks, multiple ORFs used in biosynthesis etc.), the independent linear regression tests for each of the different quantities can be combined using the data fusion technique called Fisher's combined probability test.[31]

There are hundreds of thousands of linear regression tests occurring in the pipeline. When too many inferences are made at the same time, there is a higher chance of erroneous inferences occurring; this is a case of the "multiple testing problem".

Multiple methods have been developed to correct this. We chose to use the Holm-Sidak post-hoc test to adjust p-values because it has more power than Bonferroni and Tukey methods.[32] Human genes significantly associated with the metabolite abundances after correction were considered a part of the gene signature. The PANTHER Enrichment web API is then used to perform a gene ontology enrichment analysis.[33] It will report the most enriched GO terms and a level of significance for a provided list of genes.

5.5.6 Supplementary Methods - Antimicrobial activity and determination of the minimum inhibitory concentration of *Scardovia wiggsiae* bacteriocin

The *Scardovia wiggsiae* bacteriocin was synthesized by GenScript (Piscataway, NJ, USA) and the primary structure was validated by LC/MS/MS. We conducted the antimicrobial screen using *Clostridium difficile* DSM 27147 [Ribotype 027, producer of toxins A and B (*tcdA* and *tcdB*) the binary toxin (*ctdA* and *ctdB*)] maintained on carbohydrate chopped meat (CCM) agar medium supplemented with 5% defibrinated horse blood (SR0050, ThermoScientific). The medium composition is 30 g/L peptone, 5 g/L yeast extract, 5 g/L K2HPO4, 4 g/L glucose, 1 g/L cellobiose, 1 g/L maltose, 1 g/L starch, 4 ml/L resazurin solution (0.025%), 15 g/L agar. The volume is made up of 1 litre of chopped meat broth composed of 500 g/L fat-free ground beef boiled with 25 ml/L NaOH (1N) and deionized water of up to 1 litre. As a preliminary screening to assess if the bacteriocin possesses anti-*C. difficile* activity, we conducted an agar well diffusion assay. Briefly, 10 µl of the overnight actively grown culture of *C. difficile* DSM 27147 were plated on the top of

CCMA plates then holes were punctured in the agar using a sterile glass pipette and 20 µl of 1-5 µM *Scardovia* peptide was applied into the holes. The plates were incubated anaerobically at 37 °C for 24 h. Thereafter, plates were screened for any developed zone of inhibition. To determine the minimum inhibitory concentration (MIC) of *Scardovia* peptide, we conducted a broth microdilution antimicrobial assay in 96-well microlitre plate. Briefly, a single colony of *C. difficile* DSM 27147 grown for 48 h in CCM agar supplemented with 5% defibrinated horse blood was inoculated into CCM broth for 24 h and then diluted with the same medium to 1: 10,000. Thereafter, 196 µl of this inoculated medium were added to each well, and 4 µl different serial dilutions of *Scardovia* peptide were added to the well resulting in the final concentration range starting from 100 µM to 100 nM. Blank control wells contain 196 µl non-inoculated CCM broth and 4 µl DMSO (solvent used to solubilize *Scardovia* peptide). Positive control was the wells containing 196 µl inoculated CCM broth and 4 µl DMSO. The FDA-approved antibiotic, fidaxomicin (1µM) was used as a positive control. The plates were incubated anaerobically at 37 °C. After 24 h, the OD600 of each well was measured with a microplate reader. Thereafter, MIC100 and MIC 50, defined as the lowest concentration of the peptide that results in 100% and 50% growth inhibition, respectively, were measured. Each concentration was tested in triplicates and the entire assay was repeated independently in duplicates. The percentage of growth inhibition was determined according to the following equation:

$$\%_{inhibition} = 1 - \frac{OD600_{\text{test}} - OD600_{\text{blank medium}}}{OD600_{\text{pathogen only}} - OD600_{\text{blank medium}}} \times 100$$

5.5 Results

5.5.1 BGC Gene Signatures

A total of 339 BGCs were mined from the culture-enriched metagenomes. Only 31 of 737 samples had demonstrated some for the mined BGCs. A total of 62,345,270 p-values were calculated using the linear regression between human transcripts and microbial ORF expression. After data fusion into the 339 BGCs, a total of 18,904,335 linear regression p-values remained. After p-value correction, 166 BGCs had statistically significant gene signatures. Three of the BGCs with bioactivities similar to those catalogued in human microbiota literature are explored.

5.5.1.1 Vasodilation/Vasoconstriction

For one of the BGCs, the GO Terms most enriched were:

- “detection of stimulus involved in sensory perception”
- “detection of chemical stimulus involved in sensory perception of smell’
- “detection of chemical stimulus involved in sensory perception”

These terms were tied to the olfactory receptor and taste receptor genes: OR51A4, OR14A2, OR13C3, TAS2R13, OR11H1 and OR5H15. Olfactory receptors act as

chemoreceptors throughout the body. In mice, SCFAs produced by the gut microbiota, act as olfactory receptor agonists to modulate blood pressure via renin secretion.[34] SCFAs will cause vasodilation in the human colon (in vitro). [35] Studies have shown direct correlations between microbiota-derived metabolites and blood pressure but have not discovered the pathway. [36] It is possible this encoded metabolite is modulating blood pressure in human hosts through an olfactory receptor-related pathway.

Enriched terms for another BGC were also associated with vasodilation albeit in a more straightforward manner:

- regulation of blood vessel diameter
- regulation of tube diameter
- regulation of blood vessel size
- regulation of tube size
- vascular process in the circulatory system
- regulation of blood pressure

MANGO has found secondary metabolites produced by microbes to modulate blood pressure.

5.5.1.2 Inflammation and Fatty Acid Oxidation

The top enriched terms for a potentially multi-functional BGC metabolite were:

- regulation of interleukin-1 alpha production
- inflammatory response
- negative regulation of fatty acid oxidation
- regulation of chemokine secretion

Immunomodulation is a well-documented activity associated with human microbiome-derived metabolites, especially using short-chain fatty acids (SCFAs). [37-39] Another study suggested microbes may regulate fatty acid oxidation to increase the fatty acids' bioavailability. [40] The metabolite encoded is influencing inflammation, possibly by preventing the breakdown of the SCFAs.

5.5.1.3 Cancer

One BGC appears to act on known chemotherapeutic pathways:

- ERBB2-ERBB3 signalling pathway
- ERBB3 signalling pathway
- G protein-coupled receptor signalling pathway

- non-canonical Wnt signalling pathway via JNK cascade non-canonical Wnt signalling pathway via MAPK cascade

HER2 (ERBB2) and HER3 (ERBB3) are prominent targets in breast cancer.[41] *Bacillus polyfermenticus* was shown to stop tumour cell growth by acting on HER2 and HER3.[42] The metabolite encoded in this BGC may be similar to that of *B. polyfermenticus*. GPCR signalling is known to interact with MAPK/JNK (MAPK8). [43] HER2 also interacts with Wnt signalling. The Wnt signalling pathway is tightly associated with the carcinogenesis of colorectal cancer.[44] Inhibition of Wnt signalling killed HER2 breast cancer stem cells in vivo.[45] If this metabolite is able to knock down Wnt signalling it would be a viable breast cancer and colon cancer therapeutic.

5.5.2 Metabolite Enriched Terms

The compound was found in 80.4% of samples. A total of 10 separate monoisotopic masses within the error threshold were matched to the *Faecalibacterium prausnitzii* compound. One of the masses was associated with the modulation of a large number of genes (n=294). The mass matched the M+H adduct. The gene ontology enriched terms were:

- negative regulation of activation of Janus kinase activity

- negative regulation of cytolysis by symbiont of host cells

The JAK/STAT pathway is a common target for inflammation.[46-48] Therapeutics developed to knock down Janus kinase have demonstrated effectiveness in ulcerative colitis.[48] The GO terms are reflective of anti-inflammatory activity and regulation of the human cells by the microbe.

5.5.3 Metabolite Relationship with Ulcerative Colitis

The relative abundance of the *Faecalibacterium prausnitzii* metabolite was shown to have a negative linear relationship with symptomology in Ulcerative Colitis (-0.231, $p < 0.005$). When the compound is present, inflammation is reduced.

5.5.4. RiPP Transformer

5.5.4.1 RiPP Classification Task

When trained alone RiPP Classification was able to hit 98% accuracy within 500 training steps. Its maximum performance was 99.09% accuracy on the test set. When trained in combination with the trimming head, performance is only able to hit a maximum accuracy of 82.06%. When the trained RiPP classification head was retrained with multi-task fine-tuning, it resulted in a maximum accuracy of 91.87%. (Figure 5.4)

5.5.4.2 Propeptide Trimming Task

When trained alone, the propeptide trimming head was only able to hit a maximum accuracy of 66.13%. When trained in combination with the RiPP Classification task, it was able to hit a maximum accuracy of 98.87%. The multi-task fine-tuning with a pre-trained RiPP classification head was able to reach 99.0% accuracy. (Figure 5.5)

5.5.5 *Scardovia wiggsiae* peptide relatedness

The ProtT5 model fine-tuned on RiPP classification determined the most related neighbour for the peptide as Epidermicin NI01 with a Euclidean distance of 3.926. After alignment with PRANK, its shared percentage identity was most conserved with Lactocin Z (58%). Both bacteriocins are a part of the Class II family. Other relatives and their alignments visualised with MView are found in Table 5.1.[49]

5.5.6. Peptide Enriched terms

The putative bacteriocin was found in 55% of human proteomic samples. The peptide was statistically significantly associated with 72 genes. The GO terms enriched were:

- negative regulation of toll-like receptor 7 signalling pathway
- negative regulation of interferon-alpha production

Both the toll-like receptor 7 pathway (TLR-7) and interferon-alpha (IFN-α) production are tied to inflammatory responses.[50, 51] IFN-α regulates the inflammation response and is typically associated with inducing viral clearance.[51-53] While some bacteriocins are capable of stimulating IFN-α production, none have shown the ability to knock it down.[54] Overstimulation of the TLR7 pathway and IFN-α are associated with autoimmune disorders including lupus.[55, 56] Peptide compounds, such as Thiostrepton, inhibit inflammation by modulating the TLR7 pathway.[54] Knockdown of these pathways would result in an anti-inflammatory response.

5.6 Discussion

In this work, we demonstrated a novel use of the iHMP data. While the IBDMDB metagenomes are insufficient for genomic mining, using biosynthetic gene clusters extracted from similar niches, we can still leverage the multi-omics data. The BGCs we mined from the culture-enriched metagenomes, we expressed in the publicly available data. MANGO was able to predict gene signatures for many of the BGCs. With the popularisation of metagenome-assembled genomes (MAGs), larger contigs are now

available for genomic mining of human microbiota.[57] MANGO can be used to guide microbe isolation efforts based on predicted activities encoded in the MAGs.

While the processed metabolomics data of IBDMDB had very few metabolite matches, the raw data had untapped potential. By profiling the mass spectra with a molecule we knew was produced by human microbiota, we were able to derive a gene signature reflective of its activity. As genomic mining tools get better at structure prediction, this approach can be used to deduce if the encoded metabolite is produced in situ and what potential host effects it can induce. We can also use this approach to directly investigate which metabolite is the active agent of a living medicine. *Faecalibacterium prausnitzii* is known as a staple for a healthy gut and various molecules have been proposed as mechanisms of action. Our approach allowed for the direct quantification of our active metabolite and its effect on the host's system in situ. In addition, the use of patient metadata facilitated the discovery of the metabolite as a potential mechanism for reduced symptomology in UC patients when *Faecalibacterium prausnitzii* is present. Beyond using mass-matching in MS1 data, utilisation of cosine distances in tandem mass spectrometry data would give this approach more resolution.[58] The data fusion of MS2-related peaks will increase the robustness of host modulation testing.

Beyond mass matching, we demonstrated a proteomics-directed pipeline for the deduction of active peptide compounds. Using the fine-tuned ProtT5 transformer, active

peptide products (RiPPs and bacteriocins) can be mined from microbial genomes. When mined from microbes found within the human microbiome, IBDMDB proteomics datasets can be used to determine whether or not the peptide exists in situ. MS-GF+ quantified peptides can be used in conjunction with MANGO to create gene signatures capable of accurately predicting peptide activity. The small size and cheap cost of peptide synthesis make this a lucrative approach for finding new bioactive compounds. Other metabolites require accurate structure prediction and potentially expensive synthetic reactions to produce the encoded metabolite. Our pipeline facilitates a straightforward approach for scalable bioactive peptide production.

While this dataset was dedicated to the understanding of the human microbiome, it can also be used to answer more nuanced questions about secondary metabolism in microbes. As more metabolites are isolated from the human microbiome, exploration of the relationships between gene cluster transcription, translation and biosynthesis of the metabolite can be explored. The regulation and silencing of expressed gene clusters are still poorly understood; integrated datasets such as this can facilitate their investigation. [59, 60]

The iHMP datasets are a rich and valuable resource for understanding microbial metabolism and host interactions. The MANGO pipeline demonstrates complex relationships can be fleshed out using simple statistics. While the human microbiome was

the use case for these three applications, the MANGO pipeline can be applied to other integrated datasets. Environmental datasets with metatranscriptomic data and plant phenotypes would be useful in detangling the complex interplay between crops and their microbial symbionts.[61-63] We hope the success of the IBDMDB datasets inspires the public release of more integrated datasets.

5.7 Figures and Tables

Figure 5.1 Visualization of the MANGO statistical pipeline for a single biosynthetic gene cluster. (A) Using the least squares approach, individual biosynthetic ORFs are fit to a linear regression model, where each human gene is the dependent variable and the normalized quantity of ORF transcript is the independent variable. (B) The significance of each model is combined using Fisher's combined probability test creating p-values representative of the relationship between the whole BGC to the human genes. (C) Due to the high number of comparisons being made, these values are corrected using the Holm-Sidak Post-hoc test. (D) A level of significance is chosen and the corrected p-values are assessed. Genes that pass are considered a part of the gene signature. (E) The PANTHER Enrichment web API is used to perform a gene ontology enrichment analysis. It will report the most enriched GO terms and a level of significance for a provided list of genes.

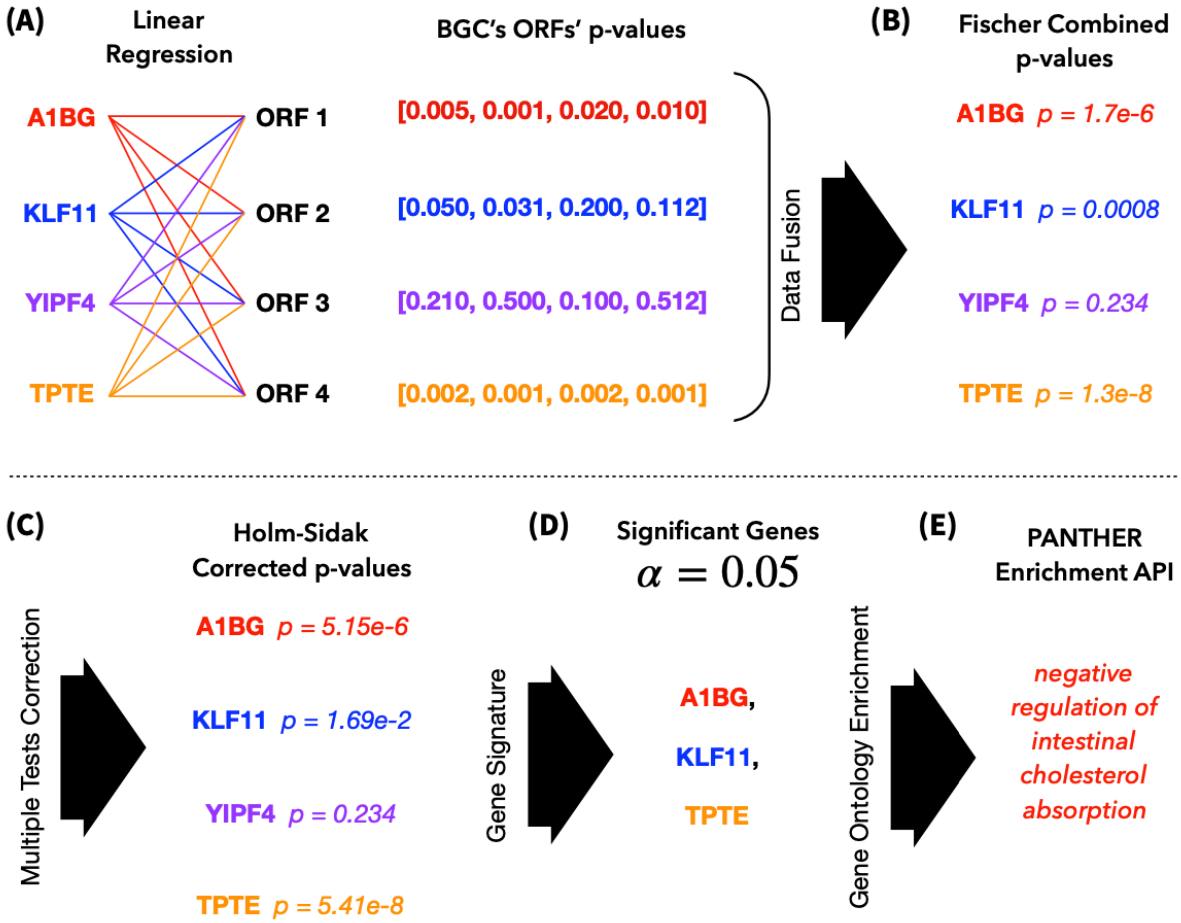


Figure 5.2: Structure of (A) *F. Prau* compound and (B) Butyrate.

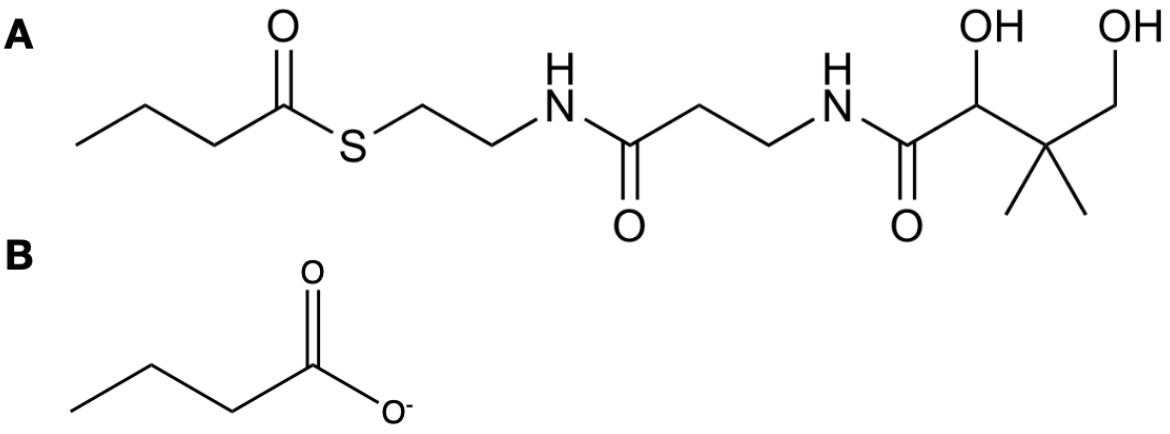
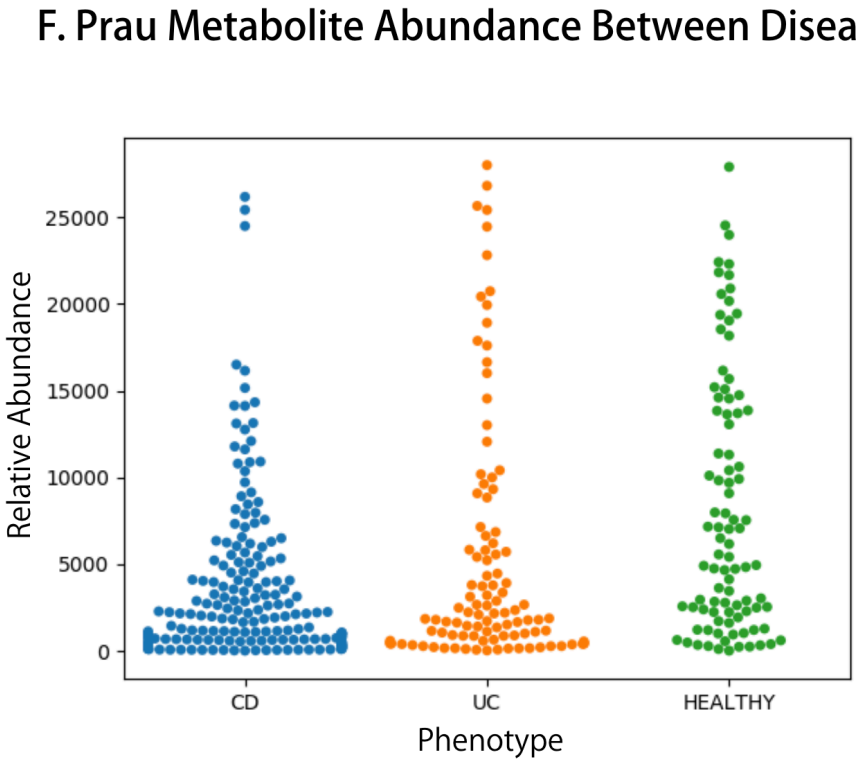


Figure 5.3 The *Faecalibacterium prausnitzii* compound's abundance across different patient subgroups. While it is mostly found in healthy patients, there are cases when the metabolite is also found in large amounts in ulcerative colitis patients.



Phenotype	Median Abundance
Healthy	**6168.0**
UC	1388.5
Crohn's	982.0

Figure 5.4 The RiPP Classification head was able to hit near-perfect performance within the first 100 training steps when trained alone. When trained with the Propeptide trimming head, the performance of this task was degraded. This degradation was evident even when pretrained to near-perfect performance before being returned with the trimming head.

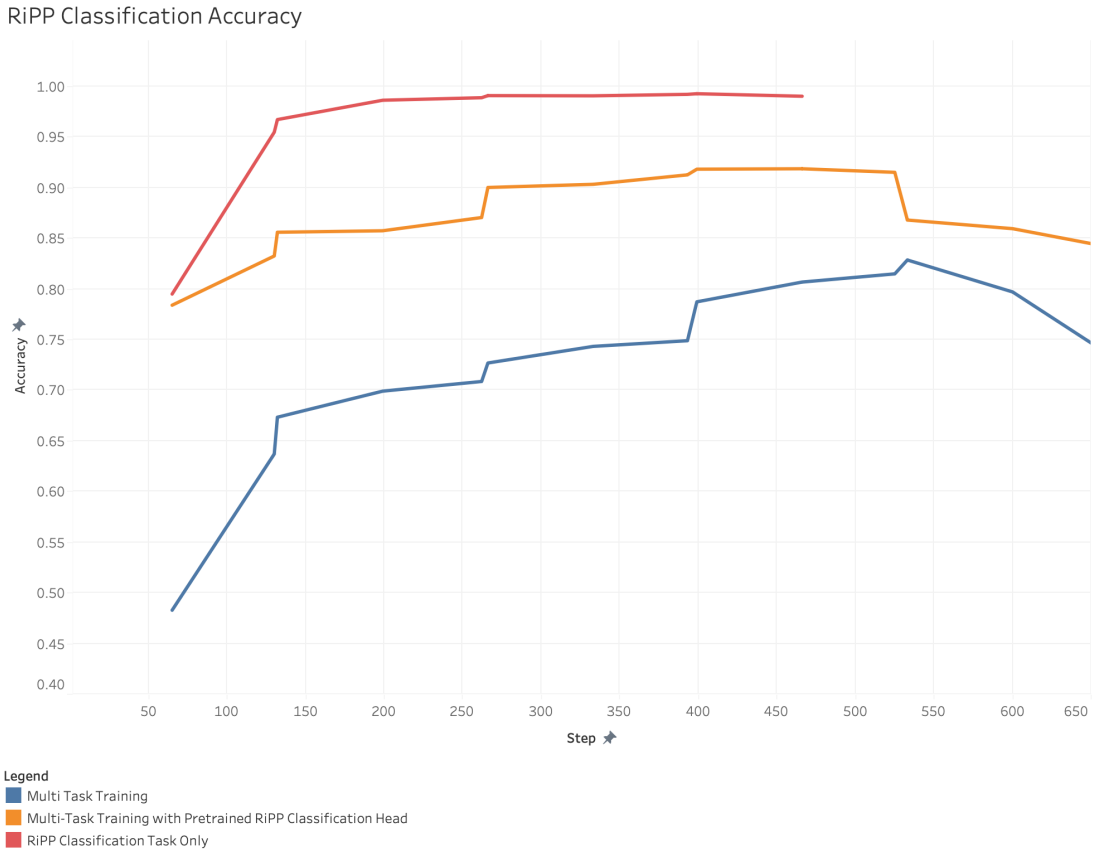


Figure 5.5 The Propeptide trimming head struggled to reach past 65% accuracy when trained alone. When combined with the RiPP classification head, the shared information brought near-perfect performance within 100 steps.

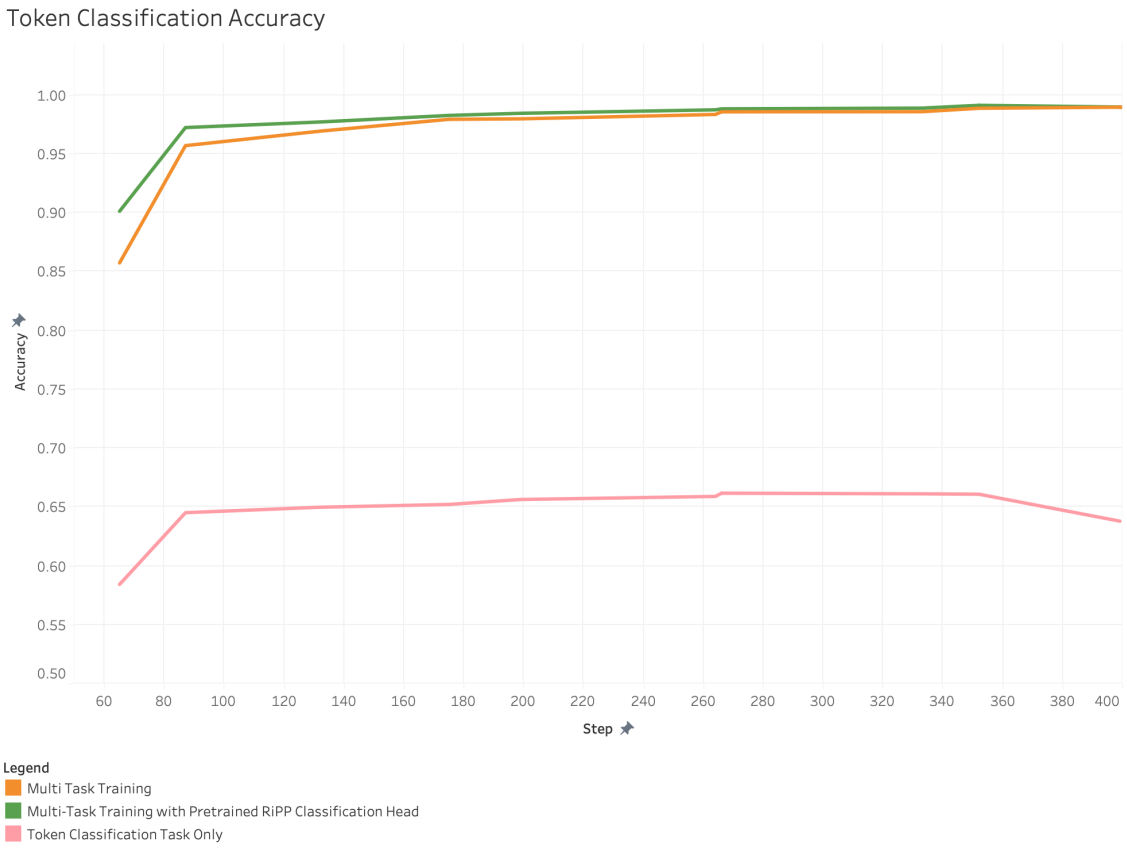


Table 5.1 The *Scardovia* peptide aligned with PRANK against its nearest neighbours according to the ProtT5 model fine-tuned for RiPP compounds. Percentage identities and alignments were visualised with the EBI's MView web tool.

Bacteriocin	Sequence	d_{RiPP}	% Identity
Scardovia Peptide	MGAFFRLLSILARYGARAVQWAWAHRGTVLRWIGAGQAIDWVIKQIKRLLGIR	-	-
Epidermicin NI01	MAAFMKLIQFLATKGQKYVSLAWKHKGTILKWINAGQSFEWIYKQIKKLWA--	3.926	49.0%
Aureocin A54	MS-WLNFLKYIAKYGKKAVSAAWKYKGKVLEWLNVGPTLEWVWQKLKKIAGL-	3.957	34.6%
Bacterocin 31	MGAIAKLV---AKFGWPIVKKYYKQ---IMQFIGEGWAINKIIDWIKKHI---	4.974	30.0%
Lacticin Q	MAGFLKVVQLLAKYGSKAVQWAWANKGKILDWLNAGQAIDWVVSKIKQILGIK	6.306	54.7%
Enterocin 7B	MGAIAKLV---AKFGWPFIKKFYKQ---IMQFIGQGWTIDQIEKWLKRH----	8.336	28.6%
Lacticin Z	MAGFLKVVQILAKYGSKAVQWAWANKGKILDWINAGQAIDWVVEKIKQILGIK	8.590	58.5%

5.8 References

1. Skinnider, M.A., et al., Genomes to natural products PRediction Informatics for Secondary Metabolomes (PRISM). Nucleic Acids Research, 2015. **43**(20): p. 9645-9662.
2. Blin, K., et al., antiSMASH 6.0: improving cluster detection and comparison capabilities. Nucleic Acids Research, 2021. **49**(W1): p. W29-W35.
3. Cragg, G.M., D.J. Newman, and K.M. Snader, Natural Products in Drug Discovery and Development. Journal of Natural Products, 1997. **60**(1): p. 52-60.
4. Tropsha, A., *4.07 - Predictive Quantitative Structure–Activity Relationship Modeling*, in *Comprehensive Medicinal Chemistry II*, J.B. Taylor and D.J. Triggle, Editors. 2007, Elsevier: Oxford. p. 149-165.
5. Yuan, H., Y.-Y. Wang, and Y.-Y. Cheng, Mode of action-based local QSAR modeling for the prediction of acute toxicity in the fathead minnow. Journal of Molecular Graphics & Modelling, 2007. **26**(1): p. 327-335.
6. Liao, S.Y., et al., QSAR, action mechanism and molecular design of flavone and isoflavone derivatives with cytotoxicity against HeLa. European Journal of Medicinal Chemistry, 2008. **43**(10): p. 2159-2170.
7. Lee, N., et al., Mini review: Genome mining approaches for the identification of secondary metabolite biosynthetic gene clusters in Streptomyces. Computational and Structural Biotechnology Journal, 2020. **18**: p. 1548-1556.
8. Skinnider, M.A., et al., Comprehensive prediction of secondary metabolite structure and biological activity from microbial genome sequences. Nature Communications, 2020. **11**(1): p. 6058.
9. Walker, A.S. and J. Clardy, A Machine Learning Bioinformatics Method to Predict Biological Activity from Biosynthetic Gene Clusters. Journal of Chemical Information and Modeling, 2021. **61**(6): p. 2560-2571.
10. Hannigan, G.D., et al., A deep learning genome-mining strategy for biosynthetic gene cluster prediction. Nucleic Acids Research, 2019. **47**(18): p. e110.
11. Liang, D., et al., Involvement of gut microbiome in human health and disease: brief overview, knowledge gaps and research opportunities. Gut Pathogens, 2018. **10**(1): p. 3.
12. Intestinal Short Chain Fatty Acids and their Link with Diet and Human Health - PMC.
13. Integrative, H.M.P.R.N.C., The Integrative Human Microbiome Project. Nature, 2019. **569**(7758): p. 641-648.
14. Donia, M.S., et al., A systematic analysis of biosynthetic gene clusters in the human microbiome reveals a common family of antibiotics. Cell, 2014. **158**(6): p. 1402-1414.

15. Lau, J.T., et al., Capturing the diversity of the human gut microbiota through culture-enriched molecular profiling. Genome Medicine, 2016. **8**: p. 72.
16. Patro, R., et al., Salmon: fast and bias-aware quantification of transcript expression using dual-phase inference. Nature methods, 2017. **14**(4): p. 417-419.
17. Lloyd-Price, J., et al., Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. Nature, 2019. **569**(7758): p. 655-662.
18. Pimentel, H., et al., Differential analysis of RNA-seq incorporating quantification uncertainty. Nature Methods, 2017. **14**(7): p. 687-690.
19. Miquel, S., et al., Faecalibacterium prausnitzii and human intestinal health. Current Opinion in Microbiology, 2013. **16**(3): p. 255-261.
20. Lopez-Siles, M., et al., Faecalibacterium prausnitzii: from microbiology to diagnostics and prognostics. The ISME Journal, 2017. **11**(4): p. 841-852.
21. Lenoir, M., et al., Butyrate mediates anti-inflammatory effects of Faecalibacterium prausnitzii in intestinal epithelial cells through Dact3. Gut Microbes, 2020. **12**(1): p. 1-16.
22. Holman, J.D., D.L. Tabb, and P. Mallick, *Employing ProteoWizard to Convert Raw Mass Spectrometry Data.* Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.], 2014. **46**: p. 13.24.1-13.24.9.
23. Yu, F., et al., Mass measurement accuracy of the Orbitrap in intact proteome analysis. Rapid communications in mass spectrometry: RCM, 2016. **30**(12): p. 1391-1397.
24. Walmsley, R.S., et al., A simple clinical colitis activity index. Gut, 1998. **43**(1): p. 29-32.
25. Merwin, N.J., et al., DeepRiPP integrates multiomics data to automate discovery of novel ribosomally synthesized natural products. Proceedings of the National Academy of Sciences, 2020. **117**(1): p. 371-380.
26. Elnaggar, A., et al., ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. arXiv:2007.06225 [cs, stat], 2021.
27. Ribeiro, F.J., et al., Finished bacterial genomes from shotgun sequence data. Genome Research, 2012. **22**(11): p. 2270-2277.
28. Netz, D.J.A., M.d.C.d.F. Bastos, and H.-G. Sahl, Mode of Action of the Antimicrobial Peptide Aureocin A53 from Staphylococcus aureus. Applied and Environmental Microbiology, 2002. **68**(11): p. 5274-5280.
29. Fagundes, P.C., et al., The antimicrobial peptide aureocin A53 as an alternative agent for biopreservation of dairy products. Journal of Applied Microbiology, 2016. **121**(2): p. 435-444.
30. Kim, S. and P.A. Pevzner, MS-GF+ makes progress towards a universal database search tool for proteomics. Nature Communications, 2014. **5**(1): p. 5277.
31. Yoon, S., et al., Powerful p-value combination methods to detect incomplete association. Scientific Reports, 2021. **11**(1): p. 6980.

32. Seaman, M.A., J.R. Levin, and R.C. Serlin, New developments in pairwise multiple comparisons: Some powerful and practicable procedures. Psychological Bulletin, 1991. **110**: p. 577-586.
33. Mi, H., et al., PANTHER version 14: more genomes, a new PANTHER GO-slim and improvements in enrichment analysis tools. Nucleic Acids Research, 2019. **47**(D1): p. D419-D426.
34. Pluznick, J., A novel SCFA receptor, the microbiota, and blood pressure regulation. Gut Microbes, 2014. **5**(2): p. 202-207.
35. Mortensen, F.V., et al., Short chain fatty acids dilate isolated human colonic resistance arteries. Gut, 1990. **31**(12): p. 1391-1394.
36. Pluznick, J.L., et al., Olfactory receptor responding to gut microbiota-derived signals plays a role in renin secretion and blood pressure regulation. Proceedings of the National Academy of Sciences of the United States of America, 2013. **110**(11): p. 4410-4415.
37. Belkaid, Y. and T. Hand, Role of the Microbiota in Immunity and inflammation. Cell, 2014. **157**(1): p. 121-141.
38. Geva-Zatorsky, N., et al., Mining the Human Gut Microbiota for Immunomodulatory Organisms. Cell, 2017. **168**(5): p. 928-943.e11.
39. Clemente, J.C., J. Manasson, and J.U. Scher, The role of the gut microbiome in systemic inflammatory disease. BMJ (Clinical research ed.), 2018. **360**: p. j5145.
40. Semova, I., et al., Microbiota regulate intestinal absorption and metabolism of fatty acids in the zebrafish. Cell Host & Microbe, 2012. **12**(3): p. 277-288.
41. Mishra, R., et al., HER3 signaling and targeted therapy in cancer. Oncology Reviews, 2018. **12**(1): p. 355.
42. Norris, V., F. Molina, and A.T. Gewirtz, *Hypothesis: Bacteria Control Host Appetites.* Journal of Bacteriology, 2013. **195**(3): p. 411-416.
43. Naor, Z., O. Benard, and R. Seger, Activation of MAPK cascades by G-protein-coupled receptors: the case of gonadotropin-releasing hormone receptor. Trends in endocrinology and metabolism: TEM, 2000. **11**(3): p. 91-99.
44. Zhao, H., et al., NPASS database update 2023: quantitative natural product activity and species source database for biomedical research. Nucleic Acids Research, 2023. **51**(D1): p. D621-D628.
45. Shah, D. and C. Osipo, Cancer stem cells and HER2 positive breast cancer: The story so far. Genes & Diseases, 2016. **3**(2): p. 114-123.
46. Schwartz, D.M., et al., JAK inhibition as a therapeutic strategy for immune and inflammatory diseases. Nature reviews. Drug discovery, 2017. **17**(1): p. 78.
47. Seif, F., et al., The role of JAK-STAT signaling pathway and its regulators in the fate of T helper cells. Cell Communication and Signaling, 2017. **15**(1): p. 23.

48. Kim, J.-W. and S.-Y. Kim, The Era of Janus Kinase Inhibitors for Inflammatory Bowel Disease Treatment. International Journal of Molecular Sciences, 2021. **22**(21): p. 11322.
49. Brown, N.P., C. Leroy, and C. Sander, MView: a web-compatible database search or multiple alignment viewer. Bioinformatics (Oxford, England), 1998. **14**(4): p. 380-381.
50. d'Hennezel, E., et al., Total Lipopolysaccharide from the Human Gut Microbiome Silences Toll-Like Receptor Signaling. mSystems, 2017. **2**(6): p. e00046-17.
51. Hadjadj, J., et al., Impaired type I interferon activity and inflammatory responses in severe COVID-19 patients. Science, 2020. **369**(6504): p. 718-724.
52. Antushevich, H., Interplays between inflammasomes and viruses, bacteria (pathogenic and probiotic), yeasts and parasites. Immunology Letters, 2020. **228**: p. 1-14.
53. Umair, M., et al., Probiotic-Based Bacteriocin: Immunity Supplementation Against Viruses. An Updated Review. Frontiers in Microbiology, 2022. **13**.
54. Huang, F., et al., *Bacteriocins: Potential for Human Health.* Oxidative Medicine and Cellular Longevity, 2021. **2021**: p. 5518825.
55. Elkon, K.B. and V.V. Stone, Type I Interferon and Systemic Lupus Erythematosus. Journal of Interferon & Cytokine Research, 2011. **31**(11): p. 803-812.
56. Brown, G.J., et al., TLR7 gain-of-function genetic variation causes human lupus. Nature, 2022. **605**(7909): p. 349-356.
57. Setubal, J.C., Metagenome-assembled genomes: concepts, analogies, and challenges. Biophysical Reviews, 2021. **13**(6): p. 905-909.
58. Bittremieux, W., et al., Comparison of Cosine, Modified Cosine, and Neutral Loss Based Spectrum Alignment For Discovery of Structurally Related Molecules. 2022, bioRxiv.
59. Amos, G.C.A., et al., Comparative transcriptomics as a guide to natural product discovery and biosynthetic gene cluster functionality. Proceedings of the National Academy of Sciences of the United States of America, 2017. **114**(52): p. E11121-E11130.
60. Mungan, M.D., et al., Secondary Metabolite Transcriptomic Pipeline (SeMa-Trap), an expression-based exploration tool for increased secondary metabolite production in bacteria. Nucleic Acids Research, 2022. **50**(W1): p. W682-W689.
61. Chukwuneme, C.F., A.S. Ayangbenro, and O.O. Babalola, Metagenomic Analyses of Plant Growth-Promoting and Carbon-Cycling Genes in Maize Rhizosphere Soils with Distinct Land-Use and Management Histories. Genes, 2021. **12**(9): p. 1431.
62. Su, P., et al., Recovery of metagenome-assembled genomes from the phyllosphere of 110 rice genotypes. Scientific Data, 2022. **9**(1): p. 254.
63. Li, P., et al., Combined metagenomic and metabolomic analyses reveal that Bt rice planting alters soil C-N metabolism. ISME Communications, 2023. **3**(1): p. 1-13.

Chapter 6: Significance and future prospective

Throughout this body of work, I demonstrated how the field of natural products research can borrow techniques used in natural language processing, and more broadly deep learning, to improve and accelerate common workflows. While the use of tried and true technology like BLAST and profile Hidden Markov Models is not harmful to the inferences made, it does bottleneck progress. As we move into an era of artificial intelligence and big data, pioneering new methods to integrate these technologies is pertinent to keep up with the rate at which data is being generated.

I developed a series of pipelines and workflows capable of finding novel microbial metabolites. Using NALA, a robust bacterial genome can be assembled. Using IBIS, the bacterial genome can be mined for molecules encoded in BGCs. Combining this software with our in-house software BEAR, a putative structure can be predicted. Using NP-BERT similar scaffolds can be found. Using my RiPP transformer encoded bioactive peptides can be found and compared. If the putative molecule or the gene cluster is found in the human microbiome, MANGO can be used to predict the metabolite's gene-drug signature. Using my tooling, a completely data-driven approach to microbial metabolite discovery can be taken.

In this work, I provided solutions for genomic mining that are scalable and easily extensible. I purposefully followed strict software engineering principles and industry standard documentation practices, to ensure the software's longevity and sustainability. The ability to add enzymes, bacteriocins and additional substrates with ease to IBIS and RiPP T5 ensures predictions will never be out of date. The high-level frameworks I designed for the retraining of BacterialT5 and EnzymeGNN, ensure robust assemblies as more diverse genomes are discovered. The attention-based approach I designed for natural product comparison, ensures substructure curation for molecular fingerprinting will never be an issue. With my sustainable solutions, resources can be allocated to other problems in the field of natural product research.

Natural Products research currently exists in multiple data domains. The fermentation of microbes is rooted in microbiology. The mining of gene clusters is rooted in genomics. The isolation of molecular scaffolds is rooted in analytical chemistry. The prediction of molecular activity is rooted in cheminformatics. Much of the tooling and data is siloed within their individual domains. My deep learning solutions facilitate cross-domain learning. I used the transformers architecture across a variety of different data domains (e.g. peptides, DNA, molecules) all with profound performance. The transformer architecture is extremely generalisable. In addition to sequence data, the Vision and Graph transformers now exist, with the ability to learn off of photos and network graphs respectively. The application of the transformer to other data domains is key to making

new inferences. Combining the VisionTransformer and Text transformer was instrumental in creating frameworks such as CLiP; cross-domain learning allowed CLiP to generate visuals when provided with a text prompt. Future efforts could be directed at creating true metabologenomic frameworks, where metabolomic-trained transformers and genomic-trained transformers could be used in conjunction. Integrated multi-omics datasets are becoming more common and will facilitate joint training operations such as this.

In my final chapter, I demonstrated the power of a simple integrated analysis. By understanding what information can be pulled out of a dataset, complex relationships can become easily disentangled. I was able to quantify gene cluster expression because I knew what BGCs would be in the transcriptomics datasets because of the culture-enriched metagenomes. The analysis does not work with MiBiG, because the human microbiome does not overlap with these gene clusters. I was able to quantify the *Faecalibacterium prausnitzii* metabolite in the metabolomics data because we had previously isolated it and F*aecalibacterium prausnitzii* is one of the most popular human microbiota. I knew the *Scardovia* peptide would be in the proteomics data because *Scardovia wiggsiae* is isolated from the oral cavity. The integration of multi-omics data requires a great appreciation for context.

To facilitate the fleshing out of contextual relationships across data domains, I started a lab initiative to build a graph database containing metadata from all domains. We have

scraped isolation information, taxonomic data, 16S sequences, molecular activities, producer data and more. Using this information, we can facilitate the cross-domain learning of new relationships. I developed a graph deep learning framework for the deep learning guided genomic assembly project. The framework will be instrumental in predicting linkages across data domains.

With my tools, and more importantly, the frameworks I created to develop my tools, I leave my lab and the field of natural products research with the potential to perform analytics on a level on par with the companies in Silicon Valley. I democratised the tooling and demonstrated proof of concepts. It is up to the next generation of graduate students to take up the torch and continue to push natural product research into another renaissance.