UNIVERSITÉ DU QUÉBEC À MONTRÉAL

LEARNING TO DISCOVER BIOSYNTHETIC GENE CLUSTERS IN FUNGI

THESIS

PRESENTED

AS PARTIAL REQUIREMENT

TO THE PH.D IN COMPUTER SCIENCE

BY

HAYDA MARCIA SOARES ALMEIDA

APRIL 2022

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

APPRENDRE À DÉCOUVRIR LES GROUPES DE GÈNES BIOSYNTHÉTIQUES CHEZ LES CHAMPIGNONS

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE

PAR

HAYDA MARCIA SOARES ALMEIDA

AVRIL 2022

ACKNOWLEDGMENTS

First, I would like to express my deepest gratitude to my thesis supervisors.

To Professor Abdoulaye Baniré Diallo, thank you for your invaluable support, patience, enthusiasm, and words of wisdom.

To Professor Adrian Tsang, thank you for your precious guidance, generosity, dependability, and encouragement.

I am beyond grateful for their faith in me. Their immeasurable knowledge and experience have inspired me. Without their guidance and support this thesis would not have been possible.

To Professor Marie-Jean Meurs, thank you for your support and for introducing me to the PhD program.

I would like to extend my deepest gratitude to my committee members Professors Vladimir Makarenkov, Vladimir Reinharz, and Dr. Pascale Gaudet for their willingness to accept this role, especially in such challenging times as a global pandemic.

I gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council of Canada and the Fonds de Recherche du Québec – Nature et technologies.

I am very fortunate to have been part of these supportive and inspiring research groups.

To my LATECE labmates Diego Maupomé, Marc Queudot, Sara Zacharie, and Antoine Briand, thank you for your friendship.

To the CSFG team, thank you for your assistance, ever since I was a master's student. Special thanks to Sylvester Palys for patiently sharing your time and expertise with me.

To the UQAM LaboBioinfo team, thank you for welcoming me and making me feel appreciated. Special thanks to Amine Remita and Golrokh Kiani for your support, friendship and generosity.

Most importantly, I am forever grateful to my family and friends for their unconditional encouragement, for their patience in the past few years, and for being a source of strength to me.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

A3C	Asynchronous Advantage Actor-Critic
antiSMASH	antibiotics & Secondary Metabolite Analysis Shell
AUC	Area Under the Curve
BARLEY	Basic Alignment of Ribosomal Encoded Products Locally
BGC	Biosynthetic Gene Cluster
BiLSTM	Bidirectional Long-Short Term Memory recurrent neural network
BLAST	Basic Local Alignment Search Tool
BOW	Bag-Of-Words
CASSIS	Cluster Assignment by Islands of Sites
CLAMS	Computational Library for Analysis of Mass Spectra
CNN	Convolutional Neural Network
DMAT	Dimethylallyltryptophan synthases
GO	Gene Ontology
F-m	F-measure
FN	False Negative
FP	False Positive
HMM	Hidden Markov Model
IMG/ABC	Integrated Microbial Genomes: Atlas of Biosynthetic Gene Clusters
LSTM	Long-Short Term Memory recurrent neural network

MIBiG	Minimum Information about a Biosynthetic Gene cluster
MIDDAS-M	Motif-Independent *De novo* Detection Algorithm for Secondary Metabolites
MIxS	Minimum Information about any Sequence
ML	Machine Learning
NCBI	National Center for Biotechnology Information
NP	Natural Product
NRPS	Non-Ribosomal Peptide Synthetase
P	Precision
PKS	Polyketide Synthase
PRISM	Prediction Informatics for Secondary Metabolomes
R	Recall
RF	Random Forest
RiPPs	Ribosomally synthesized and Post-translationally modified Peptide
RL	Reinforcement Learning
RODEO	Rapid ORF Description and Evaluation Online
RRE	RiPP Recognition Elements
SL	Supervised Learning
SM	Secondary Metabolite
SMIPS	Secondary Metabolites by InterProScan
SMURF	Secondary Metabolite Unique Regions Finder
SVM	Support Vector Machine
TC	Terpene synthase
TP	True Positive
UV	Ultraviolet

RÉSUMÉ

Les métabolites secondaires produits par les bactéries, les plantes et les champignons sont une riche source de composés bioactifs. Ces composés sont essentiels à plusieurs industries, notamment l'industrie pharmaceutique, pour la production de nombreux produits thérapeutiques tels que les antibiotiques, les immunosuppresseurs et les antitumoraux. Les gènes impliqués dans les voies métaboliques qui synthétisent les composés des métabolites secondaires sont connus sous le nom de groupes de gènes biosynthétiques (BGC). Les champignons filamenteux sont connus pour produire une grande variété de métabolites secondaires, et d'importants efforts de recherche ont été consacrés au développement d'approches pour la découverte de BGC dans les génomes fongiques. La découverte de nouveaux métabolites secondaires pourrait grandement bénéficier la santé humaine. Cependant l'identification des régions de BGC dans les génomes fongiques est un processus complexe et coûteux, et posant un défi aux approches de découverte de BGC fongiques. Cette thèse propose l'application d'approches d'apprentissage automatique pour identifier les BGC dans les génomes fongiques, impliquant trois étapes principales : (1) améliorer la disponibilité de données représentatives sur les BGC fongiques pour soutenir le développement des approches d'apprentissage; (2) identifier les potentielles régions de BGC sur les génomes fongiques; (3) optimiser les composants associés aux potentielles régions de BGC pour faciliter la curation par des experts ainsi que la caractérisation expérimentale des ses composés.

Pour améliorer la disponibilité de données représentatives sur les BGC fongiques, des ensembles de données de référence sont construits pour soutenir la conception de la prédiction des régions de BGC comme un problème d'apprentissage supervisé. Comme les ensembles de données contiennent des instances de BGC fongiques conservées et des instances de régions de non-BGC composées de gènes orthologues fongiques, la tâche de prédiction de BGC peut être abordée comme une classification binaire. La prédiction des régions de BGC potentielles est réalisée par TOUCAN, une plateforme d'apprentissage supervisé, pour lequel des modèles de classification sont entraînés sur la base des ensembles de données de référence proposés. TOUCAN s'appuie sur un ensemble d'attributs discriminants (k-mers d'acides aminés, domaines protéiques Pfam, et termes de la Gene Ontology), et sur des méthodes de post-traitement pour identifier les régions candidates de BGC dans les génomes fongiques. Finalement, une approche d'apprentissage par renforcement est proposée afin d'optimiser les régions de BGC potentielles

prédites par les outils de l'état de l'art pour la découverte des BGC. L'approche d'apprentissage par renforcement vise à améliorer la composition des régions candidates de BGC en se basant sur les profils de domaines protéiques trouvés dans les instances de BGC et non-BGC, et sur des annotations fonctionnelles des composants des BGC.

Mots-clés : apprentissage automatique, groupes de gènes biosynthétiques, génomique fonctionnelle

ABSTRACT

Secondary metabolites produced by bacteria, plants and fungi are a rich source of bioactive compounds. These compounds are vital to several industries, most prominently the pharmaceutical industry for the production of many therapeutics such as antibiotics, immunosuppressants, and antitumor. Genes involved in the metabolic pathways that synthesize secondary metabolite compounds are known as Biosynthetic Gene Clusters (BGCs). Filamentous fungi are known to produce a large variety of secondary metabolites, and significant research effort has been dedicated to develop approaches for BGC discovery in fungal genomes. The discovery of novel secondary metabolite compounds could greatly benefit human health. However identifying BGC regions in fungal genomes is a complex and expensive process, posing a challenge to fungal BGC discovery approaches. This thesis proposes the application of machine learning approaches to identify BGC regions in fungal genomes, involving three main steps: (1) improving the availability of representative data on fungal BGCs to support development of learning approaches; (2) identifying potential BGC regions on fungal genomes; (3) optimizing components associated to potential BGC regions to facilitate expert curation and experimental characterization of compounds.

To improve the availability of representative data on fungal BGCs, benchmark datasets are built to support designing the prediction of BGC regions as a supervised learning problem. Since the datasets contain instances of curated fungal BGCs, and instances of non-BGC regions composed of fungal orthologous genes, it allows the BGC prediction task to be tackled as a binary classification. The prediction of potential BGC regions is handled by TOUCAN, a supervised learning framework for which classification models are trained based on the proposed benchmark datasets. TOUCAN relies on a set of discriminative features (amino acid k-mers, Pfam protein domains, and Gene Ontology terms), and post-processing methods to identify candidate BGC regions in fungal genomes. Finally, a reinforcement learning approach is proposed as a way of optimizing potential BGC regions predicted by state-of-the-art BGC discovery tools. The reinforcement learning approach aims to improve the composition of candidate BGCs based on protein domain profiles found in BGC and non-BGC instances, and on functional annotations of known BGC components.

Keywords: machine learning, biosynthetic gene clusters, functional genomics

INTRODUCTION

The discovery of natural products, or secondary metabolites, has brought fundamental changes to society and promoted improvements in human health. From penicillin to caffeine, these compounds are so relevant as to the point that, a few decades ago, most medications in use were derived from secondary metabolite (SM) compounds (Pickens et al., 2011). These substances produced by fungi, bacteria and plants carry diverse chemical structures and high variability in their composition. Unlike primary metabolites, secondary metabolites are not required for an organism growth nor reproduction (Bills & Gloer, 2016). Instead they are known to be involved in a variety of biological activities, such as ones that allow specific reactions and adaptations to environmental factors, and also interaction with or protection against other organisms (Bills & Gloer, 2016; Keller, 2019). Filamentous fungi are a particular rich source of these bioactive compounds. Previous studies have claimed that the majority of fungal secondary metabolites revealed up until recently possesses antibacterial, antifungal or antitumor activities (Keller, 2019).

The search for fungal secondary metabolites is extremely valuable, since it could unveil novel compounds with pharmacological activities potentially leading to the development of medications. The potential of discovering relevant fungal secondary metabolites relies frequently on bioinformatics and computational biology approaches built to perform the first step of identifying the genes involved in metabolic pathways that synthesize these important compounds. These genes are often contiguously arranged in the genomes of producing organisms, and are known as biosynthetic gene clusters (BGCs) (Kautsar et al., 2020). Recent growth

in the availability of genomic and proteomic data has provided an unprecedented opportunity to search for these compounds in fungal genomes. At the same time, the demand for robust approaches to identify secondary metabolites is increasing, with new findings on possible applications of these compounds in medicine and industrial processes (Bills & Gloer, 2016).

Major challenges are associated with the discovery of fungal secondary metabolites. Bioinformatics and computational biology approaches dedicated to identify fungal BGCs generally lack scalability to be able to process in a generalized manner the exponentially growing, newly sequenced genomic data made publicly available. In addition, a common obstacle faced by these approaches is the re-discovery of known compounds, which contributes with minor progress to the field. Given the cost associated with the identification, curation, and experimental characterization of these complex compounds, the number of known fungal BGCs to date is scarce, which limits the amount of *a priori* knowledge available to support building powerful automatic discovery approaches. The validation steps within the BGC discovery process, which spans from accurate definition of BGC components to secondary metabolite compound production, could greatly benefit from predictions obtained by robust approaches that are able to follow the fast pace of newly sequenced fungal genomes made publicly available, as well as recent findings on secondary metabolite types and their applications.

The capability of machine learning algorithms to extrapolate and build experience from data makes them a possible solution to obtain high quality BGC predictions, thereby facilitating the discovery of novel compounds. In this thesis, a machine learning-based approach for BGC discovery is presented to support identification of novel secondary metabolites in fungi. While the main focus of this thesis is to identify candidate BGC regions in fungal genomes, its contributions also directly promote improvement of BGC data availability, and enhancement of candidate

BGC composition. In Chapter 1, the background context for this thesis is presented, where key concepts are introduced: definition of secondary metabolites, biosynthetic gene clusters, supervised learning, reinforcement learning, and biological features. In Chapter 2, the research problem is described, presenting the motivation, objectives and hypothesis addressed in this thesis. In Chapter 3, an overview of the state-of-the-art is presented, with a short description of previous works. Chapter 4 describes the first contribution of this thesis, which consists of the development of benchmark datasets to support BGC discovery approaches for fungi. Chapter 5 presents the second contribution of this thesis, the development of a supervised learning framework to discover fungal BGCs. Chapter 6 presents the third contribution of this thesis, a reinforcement learning approach integrating functional annotations to improve fungal BGC prediction.

This thesis includes the content of published and submitted articles which represent its main contributions.

- Chapter 4: Almeida, H., Tsang, A. and Diallo, A. B. (2019). "Supporting supervised learning in fungal Biosynthetic Gene Cluster discovery: new benchmark datasets", in IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2019, pp. 1280-1287, DOI 10.1109/BIBM47256.2019.8983041. *(article, published)*

- Chapter 5: Almeida, H., Palys, S., Tsang, A. and Diallo, A. B. (2020). "TOUCAN: a framework for fungal biosynthetic gene cluster discovery", in the journal NAR Genomics and Bioinformatics (NARGAB), 2020, Volume 2, Issue 4, December 2020, lqaa098, DOI 10.1093/nargab/lqaa098 *(article, published)*

  Almeida, H., Tsang, A. and Diallo, A. B. (2019). "Towards accurate identification of Biosynthetic Gene Clusters in fungi", in Machine Learning in

Computational and Systems Biology (MLCSB) COSI of the 27th Conference on Intelligent Systems for Molecular Biology (ISMB), 2019. *(poster, published)*

- Chapter 6: Almeida, H., Tsang, A. and Diallo, A. B. (2021). "Improving candidate Biosynthetic Gene Clusters in fungi through reinforcement learning", *(article, submitted)*
  Almeida, H., Tsang, A. and Diallo, A. B. (2021). "A reinforcement learning approach to improve fungal Biosynthetic Gene Cluster prediction", in the 25th international conference on Research in Computational Molecular Biology (RECOMB), 2021. *(poster, published)*

CHAPTER I

BACKGROUND

In this chapter, an overview of fundamental concepts is presented. These concepts include notions of the biological and machine learning backgrounds which are relevant to understanding the contributions of this thesis.

1.1 Secondary metabolites: concepts and definition

Fungi, bacteria and plants produce a variety of organic compounds that are not directly implicated in vital functions, such as organism development, reproduction or growth (Vining, 1990; Croteau et al., 2000). These compounds, known as secondary metabolites (SMs) or natural products, are classified into different types according to their chemical compositions and biosynthetic pathways. Secondary metabolite types include polyketides (PKS), non-ribosomal peptides (NRPS), terpenoids, ribosomally synthesized and post-translationally modified peptides (RiPPs), fatty acids, and alkaloids (Croteau et al., 2000; Bills & Gloer, 2016; Keller, 2019). These compounds take part in survival functions of their the producing organisms, potentially acting as defense against other microbes or environmental stress, such as exposure to UV radiation; favoring interspecies communication; or facilitating nutrient acquisition (Keller, 2015; Bills & Gloer, 2016).

The study of fungal SMs has been highly significant to human health, as they can be the source of both harmful and beneficial compounds (Keller, 2015; Bills & Gloer, 2016; Kjærbølling et al., 2019). Some fungal SMs are known for their toxic properties, while others for their pharmacological activities. Aflatoxin, sterigmatocystin, alternariol, altertoxin, trypacidin and gliotoxin are examples of known mycotoxins derived from fungal SMs, the latter being a potential anti-tumor agent in cancer treatment (Keller, 2015, 2019). Several compounds have antifungal properties, such as fumagillin, fetullamide, echinocandin, and aspterric acid (Keller, 2019). Cyclosporin and mycophenolic acid are used as immunos-

supressants (Keller, 2015; Bills & Gloer, 2016), and ergot alkaloids are used for migraine treatment (Keller, 2015). Statins, such as lovastatin and mevastatin, are used as cholesterol-lowering drugs (Bills & Gloer, 2016), while penicillin and oosporein are used as antibacterials (Keller, 2019).

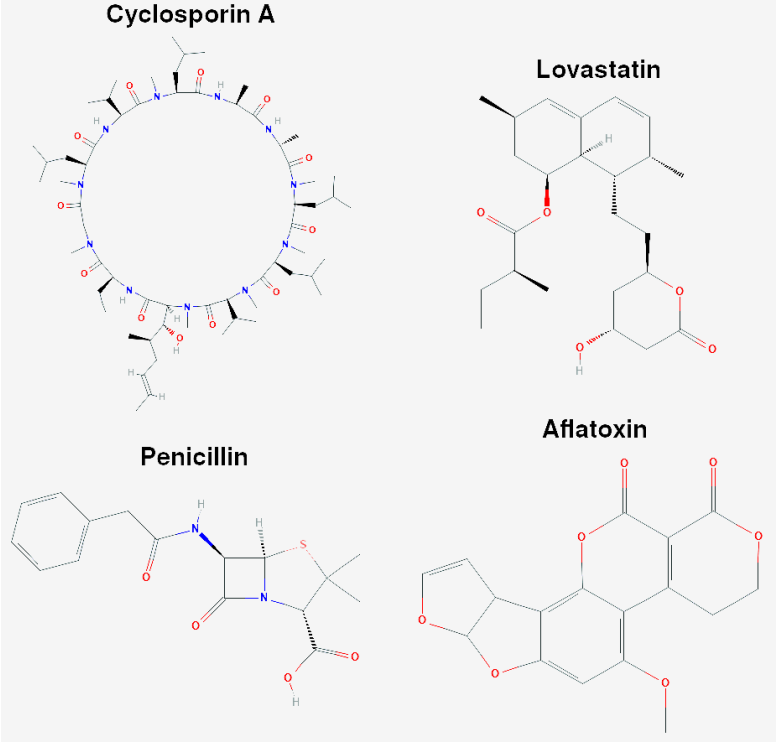


Figure 1.1: Examples of fungal secondary metabolites: cyclosporin A, used as an immunosuppressant medication; lovastatin, a cholesterol-lowering medication; penicillin, largely used as an antibiotic; and aflatoxin, a powerful carcinogenic toxin.

Filamentous fungi, more specifically the *Aspergillus* genus, are a rich source of SM compounds (Inglis et al., 2013; Kjærbølling et al., 2020). *Aspergillus niger* is a species of particular interest for SM research, given its ubiquitous presence, its relevance for industrial processes, and its capability of producing a variety of SM compounds (Aguilar-Pontes et al., 2018; Frisvad et al., 2018; Evdokias et al., 2021). Similarly, *Aspergillus nidulans* has also been a species of interest in SM research because it has been used for decades as a model organism in genetic and

cell biology studies (Kjærbølling et al., 2020; Drott et al., 2020). Filamentous fungi, including the mushrooms, are considered to hold strong potential to unveil a large variety of SM compounds (Bills & Gloer, 2016), consequently making these organisms suitable targets for SM research.

1.2 Biosynthetic gene clusters

Genes encoding biosynthetic pathways that produce SMs are often found to arrange contiguously or clustered in an organism genome. These clusters of genes are known as Biosynthetic Gene Clusters (BGCs). Biosynthetic Gene Clusters are minimally composed of at least a gene encoding a backbone enzyme, which defines the main SM compound produced by a cluster, and genes for tailoring enzymes, which are involved in the production of variants by modifying the produced core compound (Keller, 2015). For example, polyketides are made by the backbone enzyme polyketide synthase, and non-ribosomal peptides are made by the backbone enzyme non-ribosomal peptide synthetase. Tailoring enzymes such as methyltransferases add methyl groups to the core compounds to generate variants.

Apart from these minimal components, BGCs can also include cluster-specific transcription factors, transporters, and hypothetical (functionally uncharacterized) proteins (Keller, 2019), as shown in Figure 1.2. Genes encoding transcription factors in BGCs are known to play a role in the regulation of the in-cluster BGC genes. Genes encoding in-cluster transporters are involved in the export of metabolites to facilitate their function, or as a self-protection role by expelling toxins derived from the SM compound produced by the organism. Although hypothetical proteins found in BGCs may not immediately demonstrate an obvious role in the production of a compound as their biochemical function have not been characterized, are still of interest in SM research (Keller, 2015).

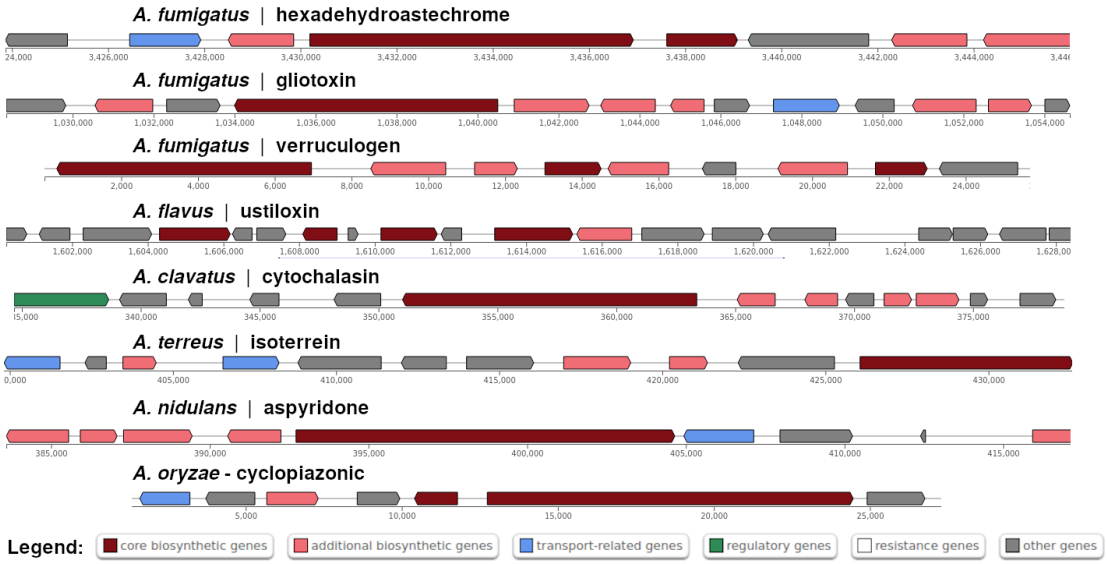


Figure 1.2: Examples of BGCs from *Aspergillus* species obtained from the public available BGC database MIBiG

Previously discovered BGCs can be found in publicly available BGC databases, such as ClusterMine360 (Conway & Boddy, 2012), antiSMASH database (Blin et al., 2018), and the Minimum Information about a Biosynthetic Gene cluster (MIBiG) database (Kautsar et al., 2019). However, the number of known compounds to date found in these databases is overwhelmingly larger for bacterial BGCs than it is for fungi, or even for plant BGCs. While ClusterMine360 is solely dedicated to bacterial BGCs, most entries in the antiSMASH and MIBiG databases are also for bacterial BGCs. From over ≈25,800 genomes in the antiSMASH database, less than 200 are fungal while more than 25,000 are bacterial (as of August 2021). Entries in the antiSMASH database consist of BGC predictions obtained with the antiSMASH tool (Blin et al., 2018) for genomes from the National Center for Biotechnology Information (NCBI).

MIBiG contains the largest number of publicly available, manually curated fungal BGCs (as of August 2021), and can be integrated with the analysis of existing BGC discovery tools, such as antiSMASH. A recent update to the MIBiG database

presented by Kautsar et al. (2019) revealed a total of 2,021 BGCs, with 1,670 from bacteria, 249 from fungi, 19 from plants and 83 unknown. The majority of BGC types in MIBiG entries across all taxonomic groups are for polyketide and non-ribosomal peptides, which corresponded to a total of 825 and 672 BGCs, respectively. While the most common genus in MIBiG is the bacteria *Streptomyces* with 568 entries, the second one is the filamentous fungi *Aspergillus*, with 79 entries (Kautsar et al., 2019).

1.3 Machine learning

Machine learning methods rely on algorithms that are capable of learning from experience and improve its own performance at a given task. These ML algorithms learn how to perform a given task T, normally by observing from data, and evaluate their performance P through specific measures, with the goal of enhancing their experience E (Mitchell, 1997). Murphy (2021) presented machine learning methods through a probabilistic perspective, and described it as decision making under certainty, where problems are viewed through random variables that hold probability distributions that indicate probable values of these variables.

Applications of machine learning methods are endless, and they have contributed to achieve outstanding advancements in a broad range of areas, from medicine to natural language to finance. Various disciplines form the field of machine learning, being supervised learning the most common in current research works. Reinforcement learning is another discipline that has recently grown among machine learning applications. Further descriptions of both disciplines are provided in the following Sections 1.4 and 1.5.

1.4 Supervised learning

In supervised learning, the objective is to learn a function $f : X \rightarrow Y$, from inputs $x \in X$ that maps it to outputs $y \in Y$ for a given task T. To learn this mapping, an algorithm samples experience E from a set of examples $D_{X,Y}$. The training dataset D is composed of input-output pairs, in which input x are features extracted from data and y are labels or categories. Data inputs in x are usually represented by a vector of fixed length. A learning algorithm is fitted to D by minimizing a loss function $\ell(y, y')$, where y' is a predicted label for an input x (Mitchell, 1997; Murphy, 2021).

Classification and regression Supervised learning tasks are most commonly represented as classification or regression problems. In regression, a learning algorithm outputs predictions of real value, such as $y \in \mathbb{R}$. In classification, a learning algorithm outputs predictions within a set of classes C, such as $C = C_i, ..., C_n$. When C represents a set of mutually exclusive n classes when $n = 2$, the problem is known as a binary classification task, whereas if $n > 2$ the problem is known as a multi-class classification (Goodfellow et al., 2016; Murphy, 2021).

Evaluation metrics Performance of supervised learning algorithms are often evaluated based on Precision (P), Recall (R), and F-measure (F-m). These metrics are specially valuable when the distribution of Y labels in a dataset D of a given task presents an imbalanced distribution, such as one of the classes C_i composing only a small fraction of the entirety of D instances. P, R and F-m are computed from the number of true positive (TP), false positive (FP), and false negative (FN) results within predictions outputted by a supervised learning model. Precision, also known as specificity, measures the number of correct predictions within all predictions, and is computed as $P = \frac{TP}{TP+FP}$. Recall, also known as sensitivity,

measures the number of correct results that were actually outputted, and is computed as $R = \frac{TP}{TP+FN}$. F-measure, or F-score, is the harmonic mean of Precision and Recall, and is computed as $F\text{-}m = \frac{2\times P\times R}{P+R}$.

Class imbalance The distribution of dataset labels may be imbalanced in certain classification tasks. The class imbalance problem occurs when the label of interest is scarce. Class imbalance is often a condition in classification tasks designed for biological datasets (Almeida et al., 2014). Data sampling is among the popular techniques to deal with class imbalance (He & Ma, 2013), due to its low computational cost and better performance compared to other methods, such as cost-sensitive techniques (Borrajo et al., 2011). Oversampling increases the number of instances in the minority class. Since new data might not be available, new instances could be artificially generated. Undersampling discards data instances belonging to the majority class to achieve a given distribution balance, but this technique might result in information loss.

Methods Logistic regression is a probabilistic, discriminative classification model that relies on a logistic function to estimate a dependent variable, which presents two possible values in a binary classification context (Murphy, 2021). A binary logistic regression predictor is defined as $p(y|\boldsymbol{x};\boldsymbol{\theta}) = (y|\boldsymbol{\sigma}(\boldsymbol{w^T x} + b))$, for a input vector $\boldsymbol{x}$, a set of class labels $y \in \{1, ..., C\}$, weights $\boldsymbol{w}$, a bias b, and a sigmoid or logistic function $\boldsymbol{\sigma}$.

A Support vector machine (SVM) is a non-probabilistic predictor trained to rely on a subset of training points known as "support vectors" at test time (Murphy, 2021). The support vectors are the closest data points found near a maximum margin from a separating hyperplane H. An SVM predictor is defined as $f(x) = \sum_{i=1}^{N} \alpha_i \mathcal{K}(x, \mathbf{x_i})$ for a weight vector α, a kernel function K, an instance to be

classified x, and support vectors $\mathbf{x_i}$.

Random forest is an ensemble classification model that relies on base decision tree learners trained on random data subsets and input variables. The model final output is obtained by an ensemble of decisions from the decision tree learners (Murphy, 2021). The ensemble Random forest predictor of a set of M trees is defined as $f(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{m=1}^{M} \beta_m F_m(\boldsymbol{x};\boldsymbol{\theta_m})$, for a input vector $\boldsymbol{x}$, a m^{th} decision tree F_m, and weights β_m.

Multilayer perceptron (MLP) is a feedforward neural network predictor, composed of stacked input, hidden and output perceptron layers, and nonlinear activation function (Murphy, 2021). A perceptron is a deterministic classifier defined as $f(\boldsymbol{x_n};\boldsymbol{\theta}) = (H\boldsymbol{w^T x_n} + b)$, that starts with random weights $\boldsymbol{w}$ which are then updated as in $\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \eta_t(\hat{y}_n - y_n)\boldsymbol{x}_n$, for a labeled input pair $(\boldsymbol{x_n}, y_n)$, and a learning rate η_t at iteration t. Backpropagation is used to compute the gradient of a loss function with regards to the weights of a given network output. While a perceptron relies on a linear threshold function $H(a)$, MLPs rely on differentiable activation functions, such as the rectified linear unit (ReLU) (Murphy, 2021).

1.5 Reinforcement learning

Reinforcement learning focuses on the problem of an autonomous agent that interacts with its environment to maximize a reward signal, through learning from its own experience (Mitchell, 1997; Sutton & Barto, 2018). A reinforcement learning agent is capable of operating under uncertainty about its environment, making these methods suited to handle interactive problems, optimization, planning, and real-time decision making (Sutton & Barto, 2018).

In reinforcement learning, an agent learns how to interact with an environment in order to maximize its perceived rewards and achieve a goal. The learning agent

has to decide between a set of available actions A to navigate different states S_i in the environment. The reinforcement learning agent core is represented in terms of a policy $\pi(x)$ which maps environment states to actions, and determines the agent behaviour. During its transition between environment states, the agent receives feedback in terms of rewards or penalties, which it uses to acquire experience and improve its performance over time.

At the same time a reinforcement learning agent attempts to maximize its reward, it has also to find a balance between exploration and exploitation. When exploiting, the agent decides to take a greedy action, meaning an action that provides maximum reward at a given point. When exploring, the agent opts for a non-greedy action, which allows it to better estimate the return value of other actions. While exploitation may offer the current maximum expected reward, exploration may offer a higher reward value over time (Russell & Norvig, 2002; Sutton & Barto, 2018).

Temporal-difference reinforcement learning methods do not require an environment model and are capable of learning after a single time step. Q-learning is a temporal-difference algorithm that learns an action-value function Q that approximates the optimal action-value function and computes expected rewards for a given state (Sutton & Barto, 2018). Apart from being a model-free reinforcement learning method, Q-learning is an incremental on-line algorithm, for which action-values are updated at each time step based on existing estimates. These aspects could indicate that Q-learning is a suitable reinforcement learning method to optimize the decision-making process in a variety of biological tasks, which can be environment-free context, and benefit from a continuous approach capable of providing on-line updates. A Q-learner is defined as $Q(S_t, A_t) = Q(S_t, A_t) + \alpha\,[\,R_{t+1} + \gamma \max_a Q(S_{t+1,a}) - Q(S_t, A_t)]$, for a set of actions A and states S, and respective rewards R at a timestep t. Q-learning considers a

α learning rate, and a γ discount-rate factor, while ϵ defines a probability for the algorithm exploration versus exploitation rate (Sutton & Barto, 2018).

1.6 Biological feature representation

Machine learning-based algorithms, including supervised and reinforcement learning, rely on data to acquire experience and produce outputs. Omics disciplines, including genomics, transcriptomics, proteomics, metabolomics and more, focus on the comprehensive study of biological molecules, often through statistical, bioinformatics and computational biology approaches (Hasin et al., 2017). Omics data are commonly represented through a variety of biological features to be processed as input by learning methods. Biological features are often sequence ones, which mainly extract patterns occurring in genome sequences; or functional ones, which focus on describing the function of protein products and genes encoded by a genome sequence (Pevsner, 2015).

Sequence features Sequence features are often extracted directly from DNA, RNA, or protein sequences, through intrinsic methods that rely on signals or sequence patterns (Pevsner, 2015). One example of sequence features are codons, a set of three nucleotides in a DNA or RNA sequence that codes for an amino acid or a stop signal to protein synthesis. Introns and exons are other examples of sequence features. While introns represent the non-coding parts of a DNA sequence, exons represent the coding parts of a sequence that are translated to proteins. Sequence motifs are also an example of features, representing a short recurring pattern of either nucleotides or amino acids that are assumed to have specific biological functions (Hashim et al., 2019), such as binding or structural properties. Finally, K-mers are also an example of sequence features. Widely used features in bioinformatics, k-mers are a sub-sequence of length k extracted

from biological sequences. Figure 1.3 shows an example of k-mer extraction from an amino acid sequence.

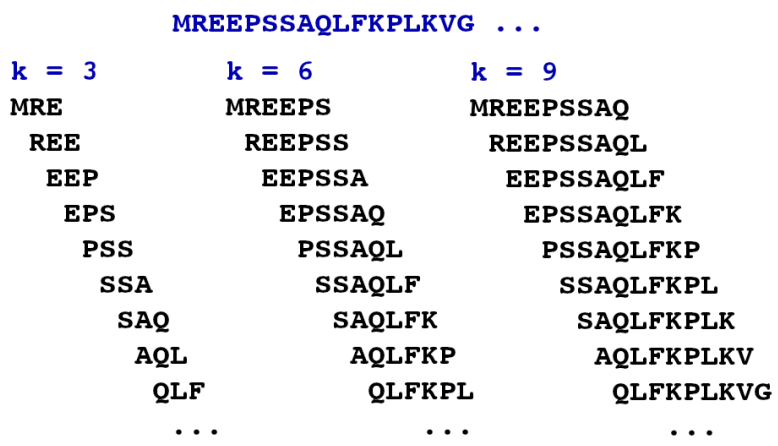


Figure 1.3: Example of k-mer extraction from an amino acid sequence

K-mer features can be representative of sequence motifs and conserved regions. Analysing the re-occurrence of motifs can point to relevant sequence regions that hold specific biological functions, such as transcription factor binding sites, elements involved in gene regulation, or the definition of protein secondary structure (Bailey et al., 2015; Hashim et al., 2019). Previous studies have evaluated the applicability of various k-mer lengths (Chor et al., 2009; Breitwieser et al., 2018; Kirk et al., 2018), however the most suitable value of k may vary widely according to the task at hand. Optimization of k values for a given problem will therefore likely to lead better performance. Generally, larger values of k will result in a more sparse vector space representation of the data, which may lead to overfitting (Wang et al., 2016).

Functional features Unlike sequence features, functional features are often obtained from external resources, such as curated biological databases, experimental approaches, or even manual annotation. Examples of functional features are conserved protein domains and families. Protein domains define the function or struc-

ture of a protein, and protein families determine a set of proteins evolutionarily related, with expressive sequence similarity. While a protein may present multiple domains with specific functions, similar domains may also be found in different proteins. The assignment of protein domains and families given an amino acid sequence normally relies on identifying protein signatures through computational approaches.

Pfam (Mistry et al., 2020) is one of the most common resources to obtain conserved protein domains and protein families for amino acid sequences. Pfam is built based on multiple sequence alignments and profile Hidden Markov Models (HMM). Protein sequence conservation is modeled using HMMs to represent a chain of states (match, delete, and insert), estimated from a seed alignment of representative of a protein family. A full alignment is built next, matching the HMMs derived from seed alignments to UniProtKB/Swiss-Prot (UniProt Consortium, 2020) sequences. Extracting Pfam domains from amino acid sequences provides information on common or shared functional profiles, making them relevant features to help identify targets for computational biology and bioinformatics tasks. Figure 1.4 shows an example of a few Pfam conserved protein domains extracted from amino acid sequences of fungal BGCs from MIBiG.

Another example of functional features are gene products, the biochemical materials resulting from gene expression (Pevsner, 2015). Gene Ontology (GO) (Ashburner et al., 2000; Gene Ontology Consortium, 2021) is an extensively used resource to obtain gene product annotations. Entries in the GO knowledge base are mostly curated from published data, and provide information about the function of genes and gene products across a variety of organisms. Annotations in GO represent a relationship between a specific gene and its function, supported by evidence. The ontology is constructed as a directed acyclic graph, a network in which GO terms are nodes connected to their ancestor and children terms.

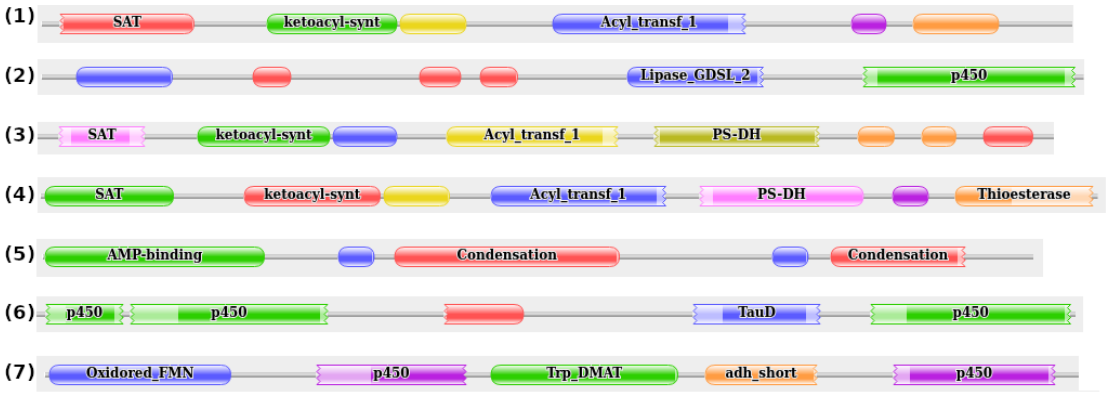


Figure 1.4: Example of Pfam domains extracted from MIBiG fungal BGCs. (1) Alternariol – *Aspergillus nidulans*; (2) lovastatin – *Aspergillus terreus*; (3) monascorubrin – *Talaromyces marneffei*; (4) naphthopyrone – *Aspergillus nidulans*; (5) ferrichrome – *Aspergillus oryzae*; (6) brassicicene C – *Alternaria brassicicola*; (7) fumigaclavine C – *Aspergillus fumigatus*

Figure 1.5 shows examples of GO terms ancestor charts for the synthesis of asperthecin (GO:0036184) and polyketide (GO:0030639), which are related to secondary metabolism. The charts were obtained with QuickGO (Binns et al., 2009), a GO browser.

There are three categories of GO terms: biological process, which involves a chemical or physical transformation to which a gene or gene product contributes; molecular function, which is the biochemical activity of a gene product; and cellular component, which indicates the cellular location where the gene product is active (Ashburner et al., 2000). Since gene products may be implicated in different processes and functions, the relationships between a gene product and GO categories are one-to-many. Similarly to Pfam domains, obtaining GO terms for genomic data helps identify a functional profile and potential targets for specific tasks, which also makes them relevant functional features.

Regarding the discovery of BGCs, both sequence and functional features are relevant to represent attributes of genomic sequences that will be processed with

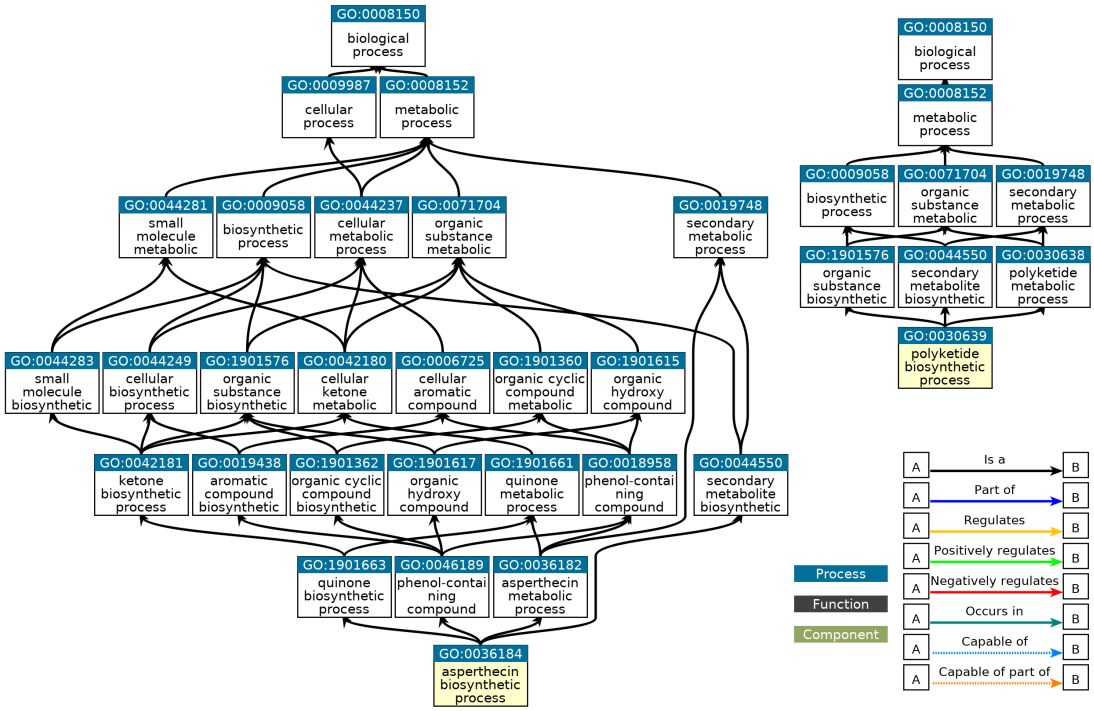


Figure 1.5: Examples of GO term annotations for the synthesis of asperthecin and polyketide

Feature	**Resource**	**Relevance to BGC discovery**	**Available at**
k-mers	-	recurrence of signature motifs	-
protein domains	Pfam	protein functions related to BGC composition and SM synthesis	`pfam.xfam.org`
gene products	GO	genes linked to secondary metabolism pathways	`geneontology.org`

Table 1.1: Biological features considered in the methodology of this thesis

bioinformatics or computational approaches to identify potential BGC candidate regions. These features can indicate the presence of patterns through the re-occurrence of sequence motifs, the presence of protein functions known to take part in BGCs and in the synthesis of SM compounds, and the appearance of processes known to belong to secondary metabolism pathways. A summary of the biological features applied in Chapters 4, 5, and 6 is presented in Table 1.1. The next chapter presents a global view of the BGC discovery research problem, describing main challenges associated with it, the hypothesis and main objectives of this thesis.

CHAPTER II

RESEARCH PROBLEM

2.1 Motivation

Secondary metabolites are bioactive compounds of diverse chemical structures produced primarily by bacteria, filamentous fungi and plants. These SM compounds were shown to provide fitness advantages and survival functions to the producing organism, for instance playing important roles in self-protection (Keller, 2015; Bills & Gloer, 2016; Drott et al., 2020). Fungal SMs have benefited human health due to their pharmaceutical properties, acting as cholesterol-lowering drugs, antifugal, immunossupressants, antibiotics, and anti-tumor agents in cancer treatment (Keller, 2015; Bills & Gloer, 2016; Chavali & Rhee, 2017). Genes implicated in the biosynthesis of SM compounds in fungi are usually co-located in an organism genome, and are known as Biosynthetic Gene Clusters (BGCs) (Kautsar et al., 2020).

Discovery of fungal BGCs could potentially lead to the identification of novel compounds relevant to the pharmaceutical and agricultural industries. With the growing amount of genome sequencing data that becomes available, the opportunities to identify novel SM compounds increase. Despite the large volume of genomic data available and previous effort put into developing approaches to identify BGCs, fungal BGC discovery remains a complex task.

One of the main reasons for the complexity in identifying fungal BGCs is due to their genomic diversity (Kjærbølling et al., 2019). In a comparative genomics analysis of *Aspergillus* species from section *Flavi*, Kjærbølling et al. (2020) demonstrated that more than half of fungal genomes varies across these related species, with a lower count of clade-specific protein families and a high count of species-specific genes. The authors also found that species-specific genes seemed to be over-represented in sub-telomeric regions, near the chromosome ends, locations previously known to be enriched with BGCs in some *Aspergillus* species. More-

over, an analysis of the species-specificity for which functional annotations were found showed their most common functions were transporters, transcriptional factors, methyltransferases and P450 enzymes suggesting their involvement in regulation and production of bioactive compounds, thus potentially also in SM synthesis. As a consequence, fungal BGC pathways of which synthesize the same or similar compounds are known to show noticeable variation in synteny among closely related species, or even between different strains of the same species (Kjærbølling et al., 2020). The diversity in fungal BGCs may be the result of the organisms evolution in response to environmental adaptation and survival (Bills & Gloer, 2016; Keller, 2019; Kjærbølling et al., 2020; Evdokias et al., 2021), providing them means to defend against other organisms, protect against UV exposure, or thrive in hostile environments.

However, the genomic diversity of fungal genomes represents a great challenge for accurately identifying fungal candidate BGC regions for *in silico* approaches, and significant curation effort is often needed to accurately identify true positive genes and reconstruct the metabolic pathway of a candidate BGC. Apart from the high genomic diversity in fungal genomes, as well as in the genes composing BGCs, the function of BGC neighboring genes can also be a source of ambiguity. For instance, certain BGC neighboring genes may show potentially relevant functional roles but not necessarily belong to a nearby candidate BGC, or take part in the metabolite production. At the same time, certain neighboring genes may encode proteins that seem inconsistent or superfluous to a nearby candidate BGC, but in reality play an important role in the cluster (Keller, 2015).

Typically, *in silico* candidate fungal BGCs obtained through different bioinformatics approaches and curated by experts are then experimentally characterized. Beyond the complexity of generating accurate *in silico* BGC predictions, experimental characterization and production of these SM compounds is also a chal-

lenging task. An important number of candidate BGC metabolic pathways are found to be silent or poorly expressed under laboratory conditions, which can prevent validation of *in silico* BGC predictions and ultimately the production of SM compounds (Montiel et al., 2015; Zhang et al., 2019). Metabolic engineering methods applied to chemically synthesize these silent or poorly expressed compounds, such as gene deletion (Gerke et al., 2012) or overexpression (Evdokias et al., 2021), promoter engineering (Montiel et al., 2015), and pathway refactoring (Zhang et al., 2019) can be complex and expensive (Pickens et al., 2011; Rahmat & Kang, 2020). The complexity associated with the experimental characterization of these compounds only reiterates the importance of generating accurate candidate BGC predictions through bioinformatics approaches, as a first step for identifying novel SM compounds. There are three main challenges that can be considered for identifying fungal BGCs: (1) data scarcity, (2) discovery of BGC regions, and (3) defining BGC composition and boundaries.

1 Data scarcity Due to all the challenging aspects of accurately identifying fungal BGCs, the number of compounds previously mapped and experimentally characterized is scarce. Among publicly available BGC databases, the number of known fungal BGCs is limited, especially when compared to the number of bacterial BGCs. The scarcity of known fungal BGCs can make it more challenging to build robust tools to support BGC discovery, given the restricted amount of curated data available to draw insights from and build useful genomic profiles for fungal BGCs. To illustrate the scarcity of fungal BGCs available in public databases, the MIBiG and antiSMASH databases hold (as of July 2021) respectively, a total of 15.7% and 1.9% of BGC entries from eukaryotes, while 84.3% and 98.1% BGC entries are from bacteria and archaea.

Not only the number of curated fungal BGCs available is rather small, BGC re-

gions themselves are scarce throughout fungal genomes. An analysis of fungal BGCs in the MIBiG database points to an estimate that, in average, BGC regions correspond to only 1% of the total genome length. Besides handling the genomic diversity of these clusters, approaches to discover BGCs must also aim to overcome the challenges of building robust methods despite the limited number of known fungal BGCs available from which to draw enough knowledge from, at the same time as handling the scarcity of the targeted and scarce BGC regions found within whole fungal genomes. Since fungal BGC regions are a small percentage of the entire genome sequence, it is also not evident how to determine non-BGC regions that could be relevant for the task and potentially support BGC discovery approaches. Creating novel datasets that represent an array of fungal BGCs, as well as a robust array of non-BGC regions could therefore be beneficial to the development of BGC discovery approaches.

2 Discovery of BGC regions Identifying candidate BGC regions is a vital step towards experimental characterization and reproduction of SM compounds. This task is normally performed with the support of bioinformatics and computational biology approaches developed for BGC discovery, often focused on specific organisms. The scarcity of fungal BGC data, as well as the characteristic genomic diversity shown by these clusters, can have a direct effect on the development of BGC discovery approaches, and therefore on the ability of such approaches to accurately identify candidate regions. Some aspects of the genomic diversity of these clusters can be observed in an analysis of the distribution fungal BGCs in the MIBiG database. The length of BGC regions varies from 617 up to 344,927 base pairs (bp), while the number of Pfam domains found in these BGCs can vary from 1 to 61, as shown in Table 2.1-A. Also Table 2.1-B shows how the BGC lengths and Pfam domain counts are spread across different size groups.

A

	bp length	**domains**
min	617	1
max	344,927	61
$\bar{x}$	31,231	18.8
σ	35,196	11.9

B

	kbp length	**domain count**
$x < 10$	22.3%	23.9%
$10 < x < 30$	40.8%	59.4%
$30 < x < 50$	22.3%	14.2 %
$x > 50$	14.5%	2.58 %

Table 2.1: **(A)** Minimum, maximum, arithmetic mean and standard deviation of base pair (bp) length and domain counts in MIBiG fungal BGCs. **(B)** Percentage of fungal BGCs in MIBiG per kilo base pair (kbp) length and domain counts.

The genomic diversity aspect can also be seen in a more detailed analysis of the Pfam domain distribution of MIBiG fungal BGCs, as shown in Figure 2.1. Pfam domains found in these clusters were manually curated by experts as *high* (usually present in BGCs) and *medium* (usually present, but not limited to BGCs). The presence of *high* and *medium* domains in MIBiG fungal BGCs is represented in each row, showing that while the clusters share a structural pattern concerning a set of specific domains, there is clearly a wide diversity of domains representing discriminative features for BGC discovery, an indication of the complexity of this task.

Previous approaches relying heavily on rule-based methods, or requiring manually curated data as input for example (Vesth et al., 2016), partially work around the data scarcity and genomic diversity aspects. But they can also result in approaches that overpredict genomic boundaries of BGCs (Khaldi et al., 2010; Blin et al., 2017), or that are not able to generalize well when predicting candidates in new genomes, performing well only for a limited number of organisms or BGC types (Khaldi et al., 2010; Takeda et al., 2014). The approach capability of generalizing when facing newly sequenced genomes is important to follow the exponential growth in genomic sequence data available, and support the identification of novel SM compounds. Additionally, overprediction of BGC boundaries that are carried over the experimental characterization phase may inflict extra

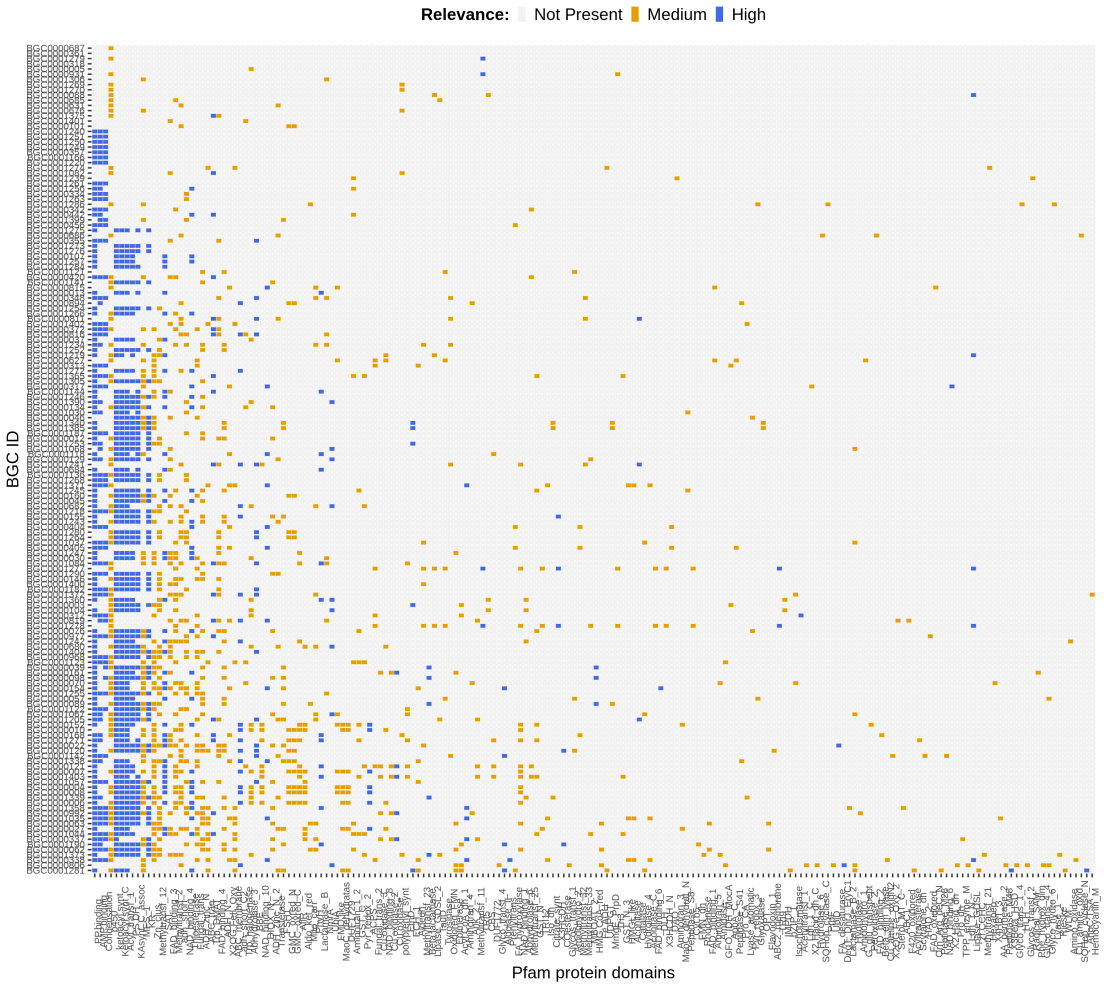


Figure 2.1: Distribution of manually curated *high* (usually present in BGCs – represented as blue points) and *medium* (usually present, but not limited to BGCs – represented as orange points) Pfam domains in MIBiG fungal BGCs, demonstrating one aspect of their genomic diversity

cost, require more expert curation and increase the process complexity.

Along with the genomic diversity and the data scarcity aspects of fungal BGCs, another concern is the re-discovery of known SMs versus identifying novel compounds (Keller, 2019). Approaches to discover BGCs often have to handle a balance between robustness, which may imply replicating the identification of previously known BGCs but missing unknown clusters; and novelty, which may imply discovering less common BGC structures, but missing known ones.

Machine learning approaches, which are able to generalize from data, might be suitable to handle fungal BGC discovery and overcome these challenges. These ML approaches have been applied more to bacteria compared to fungi, most likely due to the larger availability of bacteria BGC data.

3 Defining BGC composition and boundaries Biosynthetic gene clusters were shown to be generally composed of minimal building blocks (Keller, 2019), more specifically genes encoding a backbone enzyme and tailoring enzymes. However, BGC components are known to vary noticeably by presenting changes in the arrangement of genes involved in the metabolic pathway that synthesizes the SM compound. Cluster composition may vary at times due to the presence or absence of certain components, and at other times due to the location in which these components appear, such as in overlapping regions of neighboring BGCs, or spanning across multiple chromosomes.

While a backbone and tailoring enzymes are critical components for the synthesis of SM compounds, other components such as transcription factors, transporters, and hypothetical proteins may also be part of a BGC. The level of granularity in BGC components is often high since some components, such as transcription factors, frequently appear as cluster specific (Keller, 2019). These diverse compo-

nents could play relevant roles in the cluster biosynthetic pathway (Keller, 2015), and therefore their accurate identification is important for the BGC discovery process.

Figure 2.2: Comparison of MIBiG fungal BGCs from different species associated with the same SM compound

A comparison of varying cluster compositions among MIBiG fungal BGCs is shown in Figure 2.2, for clusters associated to the synthesis of the same SM compounds. The cyclopiazonic acid BGC in *Aspergillus flavus* spans from positions 2,000 to 16,000 containing 3 genes, and in *Aspergillus oryzae* it spans from positions 5,000 to 25,000 containing 7 genes. The cercosporin BGC appears in very different locations in *Cercospora beticola*, spanning from positions 2,255,000 to 2,295,000 corresponding to 16 genes, and in *Cercospora zeina*, spanning from positions 2,000 to 22,000 corresponding to 7 genes. Two transport-related genes and one core gene (backbone) appear in the monacolin K BGC found in *Aspergillus terreus*, while in *Monascus pilosus* the monacolin K BGC shows only one transport-related gene

and two core genes. The fumonisin BGC comprises one transport-related gene and four core genes in two *Fusarium* species, but is composed of 24 genes in *Fusarium verticillioides*, while in *Fusarium oxysporum* it is composed of 17 genes. Although the sterigmatocystin BGC spans from similar positions in two *Aspergillus* species, it is composed of 17 genes in *Aspergillus nidulans* and 26 genes in *Aspergillus ochraceoroseus*.

The variety in cluster composition is an important challenge in BGC discovery, making therefore the accurate identification of BGC components a difficult task. Correct definition of components is challenging even when BGC regions are manually curated or experimentally characterized (Kjærbølling et al., 2019). The BGC discovery process could benefit from approaches to optimize candidate BGC composition that help improve the quality of predicted BGC regions, and potentially provide better candidates for chemical synthesis of SM compounds.

2.2 Hypothesis

The main hypothesis of this thesis is that the discovery of fungal BGCs can be supported by machine learning approaches, and ultimately overcome the data scarcity, genomic diversity and accurate BGC component prediction challenges associated with this task. Firstly, machine learning approaches applied to tackle fungal BGC discovery should be able to rely on features extracted from fungal genome and proteome data, which is built as a robust representation of various fungal genomic profiles that relevant to the BGC discovery problem. Next, these machine learning-based approaches should be able to utilize relevant features to identify candidate genomic sequence regions that could potentially contain BGCs. Finally, the approaches developed should be able to improve the genomic regions identified as candidate BGCs based on patterns drawn from protein domain signatures of previously curated and experimentally characterized fungal BGCs.

Conceptualizing the main hypothesis depends on the exploitation of relevant data resources and the implementation of suitable machine learning methods. To model the discovery of fungal BGCs as a machine learning task, this thesis seeks to answer the following questions:

1. Could robust datasets encourage the development of new machine learning tools to support discovery of fungal BGCs? How to adequately portrait the diverse genomic profiles of fungi in such datasets? How to identify relevant, and discriminant non-BGC data to support development of supervised learning approaches?

2. Given a robust and representative dataset, could supervised learning approaches based on relevant features support BGC discovery in fungi, as it was previously applied in bacteria? How could the BGC discovery task be modeled as a supervised learning problem, suitable to overcome the fungal BGC data scarcity, as well as to handle the genomic diversity of these compounds?

3. Given predictions of candidate fungal BGC regions, could a reinforcement learning support optimization of cluster components? Moreover, could functional annotations provided by experts help improve the quality of predicted candidate BGCs? Could such an approach help overcome the overprediction of cluster boundaries that current state-of-the-art BGC discovery tools suffer from?

2.3 Objectives of this thesis

The global objective of this thesis is to propose a novel machine learning-based approach to identify candidate fungal BGCs. To achieve this main objective, concrete objectives are drawn here by concentrating on the hypothesis questions pre-

sented in Section 2.2, and addressed through the development of suitable methodologies which are further described in this thesis. In brief, this thesis addresses the following three concrete objectives:

1. build a set of publicly available benchmark datasets that contain a sound representation of fungal genomic profiles relevant to BGC discovery, improving BGC data availability and therefore enabling the task to be modeled as binary classification;

2. develop a robust method to identify fungal candidate BGCs based on supervised learning and utilizing the previously built benchmark datasets, that is capable of handling data from different organisms or SM types and generalizing better than previous methods for fungal BGC discovery mostly based on data-driven approaches, and that could facilitate newly sequenced, but not yet annotated, genomes to be potentially processed for BGC predictions;

3. develop a state-of-the-art method to enhance the quality of predicted fungal BGCs that is based on reinforcement learning, utilizing protein domain signatures of known and experimentally characterized fungal BGCs, exploiting functional annotations of BGC components when available to discriminate between true positive and false positive components within BGC predictions.

Following the hypothesis and objectives of this thesis, Chapter 3 describes previous tools developed to discover BGCs, presenting a comparison between different methods, target organisms and main approaches. The overview of related work in Chapter 3 illustrates the potential of applying machine learning methods to tackle fungal BGC discovery, since only a few tools use learning methods to identify BGCs, and they were mostly focused on bacteria.

CHAPTER III

RELATED WORK

Substantial research effort has been put towards developing approaches to discover BGCs. Medema & Fischbach (2015), Chavali & Rhee (2017), and Medema (2021) present reviews on various computational approaches to identify SMs, as well as new or recently updated support resources. A review of BGC discovery tools is presented in Section 3.1, with a short description of each system. Section 3.2 presents a review of approaches to perform activity and functional analysis of BGCs. In Section 3.3 a review of reinforcement learning approaches applied to biological data is presented.

3.1 BGC discovery tools

Previous BGC discovery approaches are presented and compared in this Section. Table 3.1 presents the list of studies relevant to this thesis, and a summary of their salient features. For a fair comparison, previous studies are divided into three main categories: data-driven approaches, probabilistic approaches, and machine learning approaches, presented in this order in Table 3.1 and followed by more detailed descriptions in Subsections 3.1.1, 3.1.2, and 3.1.3.

As shown in Table 3.1, most computational approaches for BGC discovery focus on bacteria, especially more recently published or updated studies. The volume of annotated BGC data available for bacteria is considerably larger than that of fungi, which may explain this aspect. The lack of annotated data for fungal BGCs, in part owing to the diversity in the organization of fungal BGCs and the scarcity of BGCs in fungal genomes, could also be a consequence of fungi-dedicated approaches being mostly data-driven as opposed to probabilistic or machine learning-based.

	Tool	Scope	Target SM	Input	Main approach	Publicly available	Last update
Data-driven	MIDDAS-M	Fungi	Unlimited	Genome sequence, transcriptome	Virtual clusters, gene expression levels	Yes	2013
	Takeda et al.	Fungi	Unlimited	Nucleotide, amino acid sequences	Comparative genomics (homologous genes)	-	2014
	FunGene ClusterS	Fungi	NRPS, PKS DMAT	Genome sequence, transcriptome,	Similar gene expression levels	Yes	2016
	CASSIS/ SMIPS	Fungi (mostly)	NRPS, PKS DMAT	Genome sequence. gene start-end, protein sequence (if SMIPS)	Motif co-occurrence in promoters around anchor genes	Yes	2016
	EvoMining	Bacteria	Unlimited	Genome sequence	Phylogenetic enzyme family analysis	Yes	2019
Probabilistic	SMURF	Fungi	NRPS, PKS DMAT	Genome sequence	Pfam HMM profiles, target domains	Yes	N/A
	ClusterFinder	Bacteria	Unlimited	Genome sequence	Pfam HMM profiles	Yes	2013
	PRISM 4	Bacteria	Unlimited	Genome sequence	Custom HMM library, rule-based	Yes	2020
	RRE-finder	Bacteria	RiPP	Amino acid sequence	Custom RiPP HMM profiles	Yes	2020
	CO-OCCUR	Fungi	Unlimited	Genome sequence	Pfam HMM profiles, comparative genomics	Yes	2020
	antiSMASH/ fungiSMASH	Bacteria, fungi	Unlimited	Genome sequence	Pfam HMM profiles, curated rules	Yes	2021
Machine learning	deepBGC	Bacteria	Unlimited	Amino acid sequence, Pfam domains	BiLSTM, Pfam2vec embeddings	Yes	2019
	NeuRiPP	Bacteria	RiPPs	Amino acid sequence	Five NN architectures	Yes	2019
	deepRiPP	Bacteria	RiPPs	Amino acid sequence	LSTM, comparative genomics and metabolomics	Yes	2020
	RiPPMiner-Genome	Bacteria	RiPPs	Genome sequence	Pfam HMM profiles, Supervised learning	Yes	2021

Table 3.1: Comparison of different methods for identifying BGCs

3.1.1 Data-driven approaches

Data-driven approaches are mostly based on genomic or phylogenetic analysis of the input data, focusing on information such as co-occurrence of gene expression levels, and sequence alignment. While data-driven approaches seem less dependent on previous curated data of known BGC structures, they might require more manual analysis effort and fine-parameter tuning.

MIDDAS-M Umemura et al. (2013) relies on extracting virtual clusters from an annotated genome sequence and checking for the co-occurence of gene expression levels among components of a BGC. Virtual clusters are obtained by extracting a sliding window of a given size, varying from 3 to 30 genes. Then, an induction ratio score is computed for each virtual cluster, by comparing its expression level in SM-producing versus non-SM-producing conditions. Virtual clusters for which genes are co-regulated with the expectation of candidate BGCs presenting a higher score, which is magnified from standard deviations of the normal distribution found in the transcriptome data. Candidate BGCs are defined as the virtual cluster size in a given region presenting the highest co-expression score.

Takeda et al. (2014) describes an approach based on comparative genomics, without relying on known BGC motifs. First, pairwise similarity search and alignment of homologous genes are performed for selected genomes. Next, scores are assigned to homologous genes, as an attempt to correct boundaries of the candidate BGC. The last step enriches the candidate BGC by clearing up syntenic blocks, meaning blocks of genes appearing across species. Parameter optimization for the algorithm is performed using *Aspergillus fumigatus* and *Aspergillus flavus* genomes.

FunGeneClusters Vesth et al. (2016) approach is based on gene expression levels among neighboring genes. A cluster score is computed for a given window of genes, based on a correlation coefficient. The authors noted that FunGeneClusters predicts co-regulated genes in general, and not specifically BGCs involved in SM synthesis. Parameters for the method are user-defined, such as the gene window size, choice of correlation coefficient, and a gene skipping threshold.

CASSIS/SMIPS Wolf et al. (2016) is based on identifying islands presenting higher concentration of cluster-specific transcription factor binding sites near potential backbone enzymes. SMIPS is applied to perform genome-wide identification of potential backbones, based on InterProScan Jones et al. (2014) protein domains. CASSIS is applied to identify biding site motifs within promoters of previously identified backbone enzymes. Motifs found are then analysed for their presence throughout the genome versus within backbone promoters. Candidate BGC boundaries are set according to the upstream and downstream occurrence of backbone promoter motifs.

EvoMining Sélem-Mojica et al. (2019) analyses enzyme expansion-and-recruitment events as indicators of enzyme association to specialized metabolism. Bacterial genomes are scanned for significant enzyme expansion-recruitment events, a process in which an enzyme performs a new function than its original one. Expanded enzymes identified are then evaluated against SM databases through BLAST, determining then possible enzyme recruitments into SM biosynthesis.

3.1.2 Probabilistic approaches

Probabilistic approaches used for BGC discovery are based on Hidden Markov Models (HMMs), and may or may not be combined with other methods. HMM

profiles in these approaches rely mostly on Pfam. Pfam HMM profiles describe sequence conservation in protein families. A chain of match, delete and insert states hold amino acid probabilities in a given state or their transition probabilities of being added or skipped in a given state, which are derived from seed and full sequence alignments of sequences annotated with specific protein families (Mistry et al., 2020).

SMURF Khaldi et al. (2010) relies on a HMM approach to identify conserved backbone protein domains, then scanning for decorating enzymes (transcription factors, transporters) in a window of ± 20 genes around the backbone enzyme. Genes within the window that present at least one SM-related protein domain are considered a positive hit. Cluster boundaries are defined by reaching either a threshold of consecutive negative hits, or a threshold in base pairs of intergenic (non-coding) region.

ClusterFinder Cimermancic et al. (2014) presents a HMM based approach, extracting contiguous Pfam protein domains from nucleotide sequences, and computing the probability of each domain belonging to BGCs. To assign probabilities for each domain, the authors rely on the protein domain frequency of training datasets, as well as in the identity of neighboring domains. The authors finally apply antiSMASH to annotate SM products from candidate BGCs.

PRISM 4 Skinnider et al. (2020) annotates genomic sequences based on a library of HMMs composed of SM related domains, and relies on a rule-based approach to identify candidate BGCs. The HMM library is built based on collections of Pfam conserved domains, curated BLAST databases, and selected multiple sequence alignment and phylogenetic analysis of SM related substrates. The

rule-based approach considers presence of minimum two SM related domains in proximity. A chemical structure analysis is then performed to predict the SM product associated with candidate BGCs found.

RRE-finder Kloosterman et al. (2020) relies on custom profile HMMs (pHMMs) designed to identify RiPP recognition elements (RREs), which are domains involved in the start of Ribosomally synthesized and Post-translationally modified Peptide (RiPP) biosynthesis. Custom HMMs are built based on sequence similarity of known RiPP classes and previously detected RREs, and based on seed BLAST alignment for unknown RiPP classes. To validate models, the authors evaluate custom HMMs of known RiPP classes against RiPP datasets or UniProtKB.

CO-OCCUR Gluck-Thaler et al. (2020) presents an approach based on candidate BGC regions predicted using SMURF (HMM). Candidate BGCs outputted by SMURF are expanded, then analysed for an unexpected co-occurrence of neighboring genes of interest among a set of fungal species to identify a conserved relationship to signature biosynthetic genes (backbones). Co-occurrence of neighboring genes is analysed within previously computed sets of *Dothideomycetes* orthologs and paralogs, and their presence in known BGCs. Predicted BGCs are assigned a SM type if there is at least 90% similarity to characterized BGC signature genes.

antiSMASH/fungiSMASH Blin et al. (2021) relies on HMMs from different databases, such as Pfam and TIGRFAM (Haft et al., 2012), and curated rules for 71 BGC types. After extracting conserved protein domains using the HMM models, it applies pre-defined cluster rules analysing presence of certain

domain families around the core gene (backbone). While expanding the prediction to neighboring genes around the backbone, candidate BGCs are checked for overlaps. The approach incorporates a set of support tools, such as RRE-finder, RODEO (Tietz et al., 2017), and CASSIS.

3.1.3 Machine learning-based approaches

Machine learning-based approaches presented in this review are mostly based on supervised learning, and may or may not be combined with other methods. Prihoda et al. (2021) presents an overview of recent advances in machine learning approaches applied to BGC discovery. All machine learning approaches addressed in this review are focused on bacteria, with most specifically designed to identify RiPPs, or applying deep learning methods. These supervised learning approaches rely on labeled training datasets composed of positive (target) and negative instances to train models, and validation and/or test datasets to evaluate the model predictive performance.

deepBGC Hannigan et al. (2019) presents an approach based on a BiLSTM model and Pfam protein domain embeddings (Pfam2vec) to predict BGC and non-BGC genome regions, and a random forest classifier to identify SM products for candidate BGCs. Pfam2vec embeddings are built using the skipgram architechture (Mikolov et al., 2013), embedding size of 100, and Pfam domains extracted from bacterial genomes in order of appearance. The classification model is composed of a BiLSTM layer of size 128, and a time-distributed dense layer with sigmoid activation.Positive instances in the training set are obtained from ClusterFinder, and negative instances generated similarly to ClusterFinder ones. The random forest classifier is built based on product class and product activity data extracted from MIBiG BGCs. Prediction scores obtained with the classification

model are averaged by gene, and consecutive regions are merged to form candidate BGCs, for which potential SM products are then also predicted.

NeuRiPP de Los Santos (2019) evaluates the application of five different neural network architectures to identify precursor peptides from RiPPs. Among the five architectures the authors listed bidirectional Long short-term memory (BiLSTM), Linear Convolutional Neural Network (CNN), Parallel CNN, Linear CNN + LSTM, and Parallel CNN + LSTM. LSTM layers were of size 60, while Linear CNNs are composed of three CNNs layers of varying sizes, and Parallel CNNs are composed of two parallel CNN layers of three different kernel sizes. As input, the neural networks receives fixed length sequences of 120 amino acids. Positive training instances are composed of precursor peptides from different RiPP classes previously identified with PRISM, RODEO, or antiSMASH, while negative instances are sequences identified as not precursors by RODEO. A undersampling method is applied to reduce the size of the negative set, which was seven times larger than the positive set. Test instances are pre-processed by antiSMASH, RODEO and Prodigal-short (Santos-Aberturas et al., 2019) before being submitted to NeuRiPP.

DeepRiPP Merwin et al. (2020) presents an approach formed by three components to identify precursor peptides from RiPPS. A NLPPrecursor component is adapted from natural language models, and relies on a three layer LSTM architecture to generate embeddings for amino acid tokens, and then to evaluate if a given input sequence is a RiPP precursor. Two LSTM layers in the embedder are of size 1140, while the last is of size 400, as well as the final embedding size. NLPPrecursor is trained on a set of precursor peptides and non-precursor peptides identified by PRISM. The approach is also composed of two other tools: BAR-

LEY, that relies on local alignment to handle dereplication of known products and promote identifying novel compounds; and CLAMS, that relies on comparative metabolomics to analyse candidate BGCs against databases of mass spectral data.

RiPPMiner-Genome Agrawal et al. (2021) introduces an approach that relied on HMM and supervised learning. Genomic regions of interest are obtained using Prodigal (Hyatt et al., 2010) for whole genomes, and analysed against Pfam HMM profiles of enzymes belonging to RiPP classes. Random forest classifiers are built for different RiPP classes, trained on previously identified precursor peptides and non-precursor peptides, in an attempt to improve the prediction obtained from the Pfam HMM analysis.

Although previous work has shown that machine learning methods are suited to handle BGC discovery, this approach has not been applied to fungal data as it has been to bacterial data. As addressed in Chapter 1, annotated fungal BGC data is scarce in comparison to bacteria, which could have affected the development of learning approaches dedicated to fungi, and consequently the discovery of novel BGCs in these organisms.

3.2 Improvement of BGC predictions with activity and functional analysis

Previous work on improving predicted BGCs mostly focused on identifying BGC main products, activities, or their chemical structures (Medema, 2021). Some studies rely on functional analysis of BGC components to improve their prediction. For instance, optimization of bacterial BGC boundaries in antiSMASH is supported by CASSIS (Wolf et al., 2016), which relies on binding sites of cluster-specific transcription factors to identify candidate BGC regions around backbone

enzymes (Blin et al., 2021).

Skinnider et al. (2020) presents an approach to predict chemical structures for bacterial secondary metabolites, within which machine learning classifiers were built to predict antibacterial, antitumor, immunomodulatory, antifungal, and/or antiviral activities. The classifiers are trained based on chemical fingerprints from structure predictions obtained by leveraging functional annotations of secondary metabolite tailoring enzyme reactions. Skinnider et al. (2020) reports results outperforming antiSMASH structure prediction, both in coverage and accuracy.

Hannigan et al. (2019) builds a Random Forest classifier to identify BGC activity, along with its BGC discovery approach. The model is built based on activity annotations extracted from MIBiG bacterial BGCs, and predicts activity within four types: antibacterial, cytotoxic, inhibitor, antifungal. Walker & Clardy (2021) also presents a machine learning approach to predict BGC activity, but with more activity types: antibacterial, anti-Gram-positive, anti-Gram-negative, antifungal, antitumor or cytotoxic.

Activity prediction in both Hannigan et al. (2019) and Walker & Clardy (2021) is based on bacterial BGCs only, obtained from MIBiG. While Hannigan et al. (2019) extracts chemical activity data from MIBiG entries, Walker & Clardy (2021) associates an activity to a BGC through evidence curated from literature review of previously reported specific activity associated to a BGC. The training dataset in Hannigan et al. (2019) contains 370 instances, and relies solely on Pfam domains as features. Walker & Clardy (2021) represents dataset instances with antiSMASH annotations containing secondary metabolite ortholog groups and motifs, as well as Pfam domains and Pfam subfamilies, which are generated with a sequence similarity network algorithm (Gerlt et al., 2015). The more robust feature representation in Walker & Clardy (2021) outperformed results obtained

previously by Hannigan et al. (2019). Both approaches have to handle an imbalanced dataset, and report an $\approx<$ 40% average F-m (Hannigan et al., 2019) and an $\approx$ 70% average balanced accuracy (Walker & Clardy, 2021) when predicting BGC activity.

Optimizing candidate BGCs based on different aspects, such as chemical structures, activity, or functional annotations is a relevant step on BGC discovery. It allows for potentially more accurate and richer BGC predictions, facilitating the experimental characterization process by denoting cluster components and boundaries more precisely, or even lead to the discovery of novel compounds.

3.3 Reinforcement learning approaches for biological data

Applications of reinforcement learning to process biological data are less common when compared to other machine learning methods such as supervised learning (Mahmud et al., 2018). Previous works rely on reinforcement learning to perform optimization tasks, such as metabolic engineering through bioretrosynthesis (Koch et al., 2019), multiple sequence alignment for nucleotide sequences (Mircea et al., 2018; Ramakrishnan et al., 2018; Song et al., 2021), control gene regulatory networks (Imani & Braga-Neto, 2018), as well as design biological sequences (Eastman et al., 2018; Angermueller et al., 2020) and *de novo* drug-like compounds (Gottipati et al., 2020; Jeon & Kim, 2020).

Koch et al. (2019) present an approach based on Monte Carlo Tree Search to explore metabolic pathways capable of *de novo* bioretrosynthesis reactions, which identifies molecules and enzymes that are capable to convert themselves into a given target. The sequential decision process starts from a state with a target compound to be produced, for which its molecules can be transformed through allowed actions that continue until a leaf node in the search tree is reached. A ranking scheme is established for possible compound transformations in given states,

providing rewards according to a biochemical score associated to the transformation.

Mircea et al. (2018) introduce a multiple sequence alignment Q-learning approach to progressively align a set on input sequences, based on local optimal alignments and a profile computed for input sequences, which determines their frequency for each nucleotide. Rewards for candidate alignments consider the match or mismatch between nucleotides (or gap) at a given position, and are computed based on the sum-of-pairs score.

In Ramakrishnan et al. (2018) the authors also describe a multiple sequence alignment approach, but based on a Asynchronous Advantage Actor Critic (A3C) model containing two separate neural networks, one considered as an actor and the other a critic, composed of convolutional and fully connected layers. The model is fed randomly aligned and generated sequences, and computes its reward based on the sum-of-pairs score for proposed alignments.

The multiple sequence alignment approach in Song et al. (2021) combines two deep (convolutional) Q-networks: a dueling deep-Q network, that learns scores separately for states and actions; and a double deep-Q network, that aims to avoid overestimation by converging towards the main network using a constant ratio. Input nucleotide sequences are processed as a window of sub-sequence pairs, for which smaller alignments are generated, shifting to subsequent windows in the environment through forward, insertion or deletion actions.

Imani & Braga-Neto (2018) propose a Bayesian inverse reinforcement learning approach to control gene regulatory networks. Gene expression measurements are represented based on a partially-observed boolean signal model. Q-learning is applied to simulate an expert intervention policy to estimate the pathway cost of activating or suppressing certain control genes given an input gene. The goal is to

accurately modify the gene network dynamics through gene up or down-regulation and achieve the desired effect, such as suppressing genes associated with cancer metastasis, or activate genes associated with tumor suppression.

Eastman et al. (2018) present an approach to perform inverse RNA folding, and design RNA sequences that will fold into a target input secondary structure. Given an input candidate RNA sequence of fixed length, the reinforcement learning agent is trained to modify it through actions so as to achieve a desired structure. Rewards are only assigned when the agent reaches the target structure. The approach policy network is based on the Asynchronous Advantage Actor-Critic (A3C) algorithm, and consists of a set of multiple convolutional layers.

Angermueller et al. (2020) also propose a model based reinforcement learning approach to perform sequence design using on proximal policy optimization. Actions determine the next character in a sequence string from left to right, while rewards are only assigned at the terminal state of each input sequence. The reward function is converged by fitting a supervised regression model on the possible sequence prefixes collected at a certain step, unless the reward model presents uncertainty above or accuracy below specific thresholds. To help increase sequence diversity and avoid redundancy, the approach processes input sequences in large batches of approximately 100 to 1000 instances, and when computing the terminal state reward, the authors verified for sequence similarity against already proposed sequences.

Jeon & Kim (2020) describe an approach based on double Q-learning and deep-Q networks to optimize molecules and design of drug-like compounds. Three dimensional structure of target proteins are taken as input. Actions in the reinforcement learning approach determine atom or bond addition or removal, under valid constraints, while the rewards are computed according to specific properties obtained

by the modified molecules.

Similarly to Ramakrishnan et al. (2018), Gottipati et al. (2020) also present an approach based on actor-critic networks that aims to identify most suitable chemical reactants to interact with a given molecule and achieve a specific reaction. In Gottipati et al. (2020), the actor networks based on fully connected layers select a chemical reactant and compute possible products to be generated, while the critic network, based on a double Q-learning, provided scores on the predicted products obtained, aiming to maximize its rewards.

As shown in Section 3.1, most previous works focusing on fungi are based on data-driven or probabilistic methods, thereby opening an opportunity for exploring machine learning methods to identify fungal BGCs.

CHAPTER IV

NEW BENCHMARK DATASETS FOR FUNGAL BGC DISCOVERY

A first step towards building a robust machine learning approach for fungal BGC discovery was designing benchmark datasets, to tackle the task of identifying candidate BGC regions as a supervised learning problem. This Chapter describes the methodology adopted to built new benchmark datasets to represent diverse fungal genomic profiles, providing a solid knowledge basis for learning methods. The study presented in this Chapter was published in the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) issue and presented at the Machine Learning and Artificial Intelligence in Bioinformatics and Medical Informatics (MABM 2019) in the same conference, under the title "Supporting supervised learning in fungal Biosynthetic Gene Cluster discovery: new benchmark datasets". Article writing, approach implementation, experimental design and execution were performed by Hayda Almeida under the supervision of professors Adrian Tsang and Abdoulaye Baniré Diallo. A printed version of this article is presented in the Appendix A.

4.1 Abstract

Fungal Biosynthetic Gene Clusters (BGCs) of secondary metabolites are clusters of genes capable of producing natural products, compounds that play an important role in the production of a wide variety of bioactive compounds, including antibiotics and pharmaceuticals. Identifying BGCs can lead to the discovery of novel natural products to benefit human health. Previous work has been focused on developing automatic tools to support BGC discovery in plants, fungi, and bacteria. Data-driven methods, as well as probabilistic and supervised learning methods have been explored in identifying BGCs. Most methods applied to identify fungal BGCs were data-driven and presented limited scope. Supervised learning methods have been shown to perform well at identifying BGCs in bacteria, and could be well suited to perform the same task in fungi.

But labeled data instances are needed to perform supervised learning. Openly accessible BGC databases contain only a very small portion of previously curated fungal BGCs. Making new fungal BGC datasets available could motivate the development of supervised learning methods for fungal BGCs and potentially improve prediction performance compared to data-driven methods. In this work we propose new publicly available fungal BGC datasets to support the BGC discovery task using supervised learning. These datasets are prepared to perform binary classification and predict candidate BGC regions in fungal genomes. In addition we analyse the performance of a well supported supervised learning tool developed to predict BGCs.

4.2 Introduction

Natural products (NPs) are specialized bioactive compounds primarily produced by plants, fungi and bacteria. NPs are a vital source for drugs: from anti-cancer,

anti-virus, and cholesterol-lowering medications to antibiotics, and immunosuppressants (Chaudhary et al., 2013; Medema & Fischbach, 2015; Chavali & Rhee, 2017). Unlike those in plants, genes involved in the biosynthesis of many NPs in bacteria and fungi are co-localized in the genome of organisms and usually organized as clusters of genes (Osbourn, 2010). Gene clusters capable of producing NPs are known as Biosynthetic Gene Clusters (BGC).

The task of identifying new BGCs could potentially lead to the discovery of novel NPs to benefit human health. However this task involves complex and costly processes, as well as the analysis of large amounts of biological data. Development of automatic tools that can support the identification of BGCs is therefore highly relevant. Various approaches have been used to develop such tools, such as data-driven methods, probabilistic methods, and supervised learning methods. In supervised learning the BGC discovery task can be represented as binary classification task. The goal in a binary classification task is to classify data instances as belonging to one out of two different categories. A binary classification BGC dataset would therefore be composed of positive and negative BGC instances.

Supervised learning has been previously used to predicting bacterial BGCs (Agrawal et al., 2017; Hannigan et al., 2019) and shown to perform well. Supervised learning methods however are developed primarily based on annotated datasets, for which all instances are labeled as belonging to a specific class. Unlike for bacteria, the number of known fungal BGC data previously validated by curators is rather limited. The Minimum Information about a Biosynthetic Gene cluster (MIBiG) (Medema et al., 2015)[1] repository is one of the largest freely available BGC databases.

As an example of the disparity between known available BGC from bacteria versus

[1] http://mibig.secondarymetabolites.org/

fungi that has been annotated by curators, MIBiG holds over 1,196 bacteria BGCs, while only 206 are fungal BGCs[2].

Generating fungal BGC datasets for supervised learning approaches imposes a few challenges. For instance, negative samples are needed for binary classification, and they are not directly provided by BGC databases just as annotaded BGC data. To be able to support a robust classification approach, fungal BGC datasets used as input should include a variety of organisms and BGC types to properly represent fungal genomic profiles.

The availability of fungal BGC datasets could leverage the development of new supervised learning approaches to tackle BGC discovery in fungi. This work presents new datasets prepared to tackle fungal BGC discovery as a binary classification task. These datasets are constructed in such way that they include most variety of BGC types from different organisms, attempting to represent fungal genomic profiles to better suit the fungal BGC classification task. Finally we also analyse the usage of fungal BGC datasets with one of the state-of-the-art supervised learning methods developed for BGC discovery, DeepBGC (Hannigan et al., 2019).

4.3 Previous work

In this section we present previous work on the availability of BGC data previously predicted or annotated by curators that can support BGC discovery, and previous work conducted towards developing automatic approaches to identify fungal BGCs. BGC databases and some of their characteristics are discussed in Section 4.3.1. Previous work on predicting BGCs in fungi is presented in Section 4.3.2.

---

[2]As of July 2019.

4.3.1 BGC Databases

Only a small number of open access BGC databases is currently available to support research on automatic tools to identify BGCs. The majority of entries in these databases corresponds to bacteria data, while only a small portion are fungal BGCs.[3] MIBiG is a BGC repository in which curated entries are submitted by curators, and added to the database in a format compliant with the Minimum Information about any Sequence (MIxS) framework data standard. It holds 206 fungi BGCs and 1,196 for bacteria. Clustermine360 (Conway & Boddy, 2012) contains microbial polyketide synthases (PKS) and non-ribosomal peptide synthetases (NRPS) biosynthesis. It holds a total of 29 fungal BGCs, while over 900 are from bacteria. Clustermine360 entries are curated and submitted by curators, enriched with information from the National Center for Biotechnology Information (NCBI)[4], and analysed with the antiSMASH (Blin et al., 2017) tool. The antiSMASH database (Blin et al., 2016) has 24,773 microbial BGCs predicted based on its homonymous tool. Unlike its bacteria version, the fungal version of antiSMASH does not provide a database of fungal BGCs to the best of our knowledge.

The Integrated Microbial Genomes: Atlas of Biosynthetic Gene Clusters (Hadjithomas et al., 2016) (IMG/ABC) database contains BGCs predicted using the ClusterFinder algorithm (Cimermancic et al., 2014). IMG/ABC holds 127 fungal BGCs and 1,025 from bacteria.

These databases are not connected. Since it is likely that there are overlaps among the different databases, the number of unique fungal BGCs could be even

---

[3]Number of entries for databases are reported as of July 2019.

[4]https://www.ncbi.nlm.nih.gov/

smaller. The small proportion of fungal BGCs across databases is an example of the challenges in developing automatic tools to tackle BGC discovery in fungi. This work proposes new publicly available datasets to be an input of supervised learning tools to predict fungal BGCs, based on MIBiG and orthologous genes. The details on our datasets and their analysis are discussed in Section 4.4.

4.3.2 BGC discovery in Fungi

Significant effort has been put towards developing approaches to discover BGCs (Medema & Fischbach, 2015; Chavali & Rhee, 2017). The majority of approaches focused on processing bacterial data, while some of them are specially focused on fungi. Identifying BGCs remains a challenging task specially in fungal genomes, due to the diversity of clusters (Kjærbølling et al., 2018).

Previous work on fungal BGC discovery made use mostly of data-driven methods, which are heavily based on the analysis of the input or output data and require fine parameter-tuning. These methods required as input the genome sequence combined with transcription data (Vesth et al., 2016; Umemura et al., 2013), or gene functional annotations (Wolf et al., 2016), as well as both nucleotide and amino acid sequences (Takeda et al., 2014). Vesth et al. (2016) and Umemura et al. (2013) focused on analysing similar gene expression levels, while Umemura et al. (2013) used virtual clusters. Vesth et al. (2016) looked at motif co-occurrence in promoters around anchor genes, and Takeda et al. (2014) analysed homologous genes through a comparative genomics approach.

Such data-driven methods are less dependent on curated BGC data, which are time consuming to obtain, but they all present limitations. Wolf et al. (2016) requires gene functional annotations, which may not be available, and Vesth et al. (2016) relies heavily on manual curation of output to achieve the expected results. A very limited BGC prediction scope is considered in Khaldi et al. (2010)

and Takeda et al. (2014). Both approaches are developed based on biological sequences from a single species, and they also require fine parameter-tuning. Such limitations of data-driven methods can restrict their ability to generalize to new data, and as a consequence compromise the discovery of novel BGCs.

Likely due to the larger availability of curated BGC data, probabilistic (Cimermancic et al., 2014; Skinnider et al., 2015; Blin et al., 2017) and machine learning approaches (Agrawal et al., 2017; Hannigan et al., 2019) have been more explored in bacteria compared to fungi, and shown to perform well. Probabilistic and machine learning approaches could be beneficial for BGC discovery, since by nature they are more capable of generalizing given new data, and will likely perform better at identifying data patterns and discovering novel BGCs, when compared to data-driven methods. In this study we also analyse the performance of a supervised learning approach developed to tackle BGC discovery using the fungal BGC datasets proposed by our work. The details on our experimental setup are further discussed in Section 4.4.

4.4 Methodology

Some of the challenges in generating fungal BGC datasets for binary classification are the need of negative instances, which are not directly provided in BGC databases; and accounting for a variety of organisms, BGC types, and also fungal genomic profiles. The availability of new fungal BGC datasets however could potentially motivate the development of supervised learning approaches to tackle fungal BGC discovery.

In this work we propose new publicly available fungal BGC datasets to support supervised learning approaches tackling BGC discovery as a binary classification task. We present here the methodology adopted to prepare fungal BGC datasets and their analysis using a supervised learning method, with the goal of analysing

the method performance in fungal BGC data.

Details on our proposed fungal BGC datasets are presented in Section 4.4.1. Section 4.4.2 presents the test datasets with which we analysed the performance of classification models built on fungal BGC datasets. In Section 4.4.3 we provide details on the parameters considered in our analysis based on a supervised learning method, as well as the classification models considered.

4.4.1 Proposed Datasets

Supervised learning was shown to perform well at BGC discovery in previous work that focused on handling bacteria data (Agrawal et al., 2017; Hannigan et al., 2019). Given that annotated data are needed to perform a supervised learning approach, we propose here fungal BGC datasets to support the development of this approach for fungi.

As mentioned in Section 4.2, positive and negative instances are needed to perform fungal BGC discovery as a binary classification task using supervised learning. To create our fungal BGC datasets, we extracted and filtered positive instances from the MIBiG (Medema et al., 2015) repository, previously presented in Section 4.3.1. MIBiG has the highest number of unique fungal BGCs among the BGC databases previously presented. Additionally, MIBiG BGCs were annotated and submitted by the research community, unlike BGCs in other databases that were automatically predicted.

From all MIBiG instances, we have selected only the fungal BGC subset, excluding BGCs belonging to *Aspergillus niger* (*A. niger*) to avoid overlaps during the test phase, resulting in a total of 200 positive instances.

We generated synthetic negative instances collecting and integrating orthologous

genes from OrthoDB[5] (Kriventseva et al., 2018). Orthologs are homologous genes descendants from a single gene of a last common ancestor. The OrthoDB database contains protein-coding genes that represent the last common ancestors given a specific phylogeny radiation of a species, and are therefore known to retain ancestral function (Kriventseva et al., 2018). Orthologs represent regions conserved across species. They can correspond to a relevant negative instances for BGC discovery. This is due to the fact that fungal BGCs are known to have opposite characteristics and show large genomic diversity even in otherwise closely-related or same genus species (Kjærbølling et al., 2018). Genes belonging to fungal BGCs have been previously referred to as “species-specific" (Vesth et al., 2018), unlike orthologs.

Orthologous genes have been previously used to discover BGCs in fungi. In Takeda et al. (2014), the authors presented an alignment-based approach focused on identifying syntenic block regions, which are more likely to contain orthologs and less likely to contain BGCs. Non-syntenic blocks were then used to search for candidate BGCs and to better define candidate cluster boundaries. The approach in Takeda et al. (2014) was explored in small set of 10 filamentous fungi. The results showed good performance, predicting correctly 21 out of 24 fungal BGCs.

In this study we selected the fungal OrthoDB subset to construct the synthetic negative BGC instances. The OrthoDB fungal subset contains a total of 5,083,652 non-redundant orthologs. To avoid potential overlaps, we performed a BLAST analysis between the fungal subsets of both OrthoDB and MIBiG. We discarded 11,000 ortholog matches found using the BLAST parameter *evalue* (expected value) set to $1e-60$.

To generate synthetic negative instances, we then concatenated the amino acid

[5] `http://orthodb.org/`

sequence of fungal orthologs using a fixed length of 7,000 amino acids to create synthetic gene clusters. The 7,000 amino acid length is chosen since it corresponds to the average length of fungal BGC amino acid sequences in MIBiG. Figure 4.1 shows an example of positive instances in our datasets and negative instances being generated from OrthoDB orthologs. After processing OrthoDB fungal orthologs a total of 693,195 synthetic negative clusters were generated.

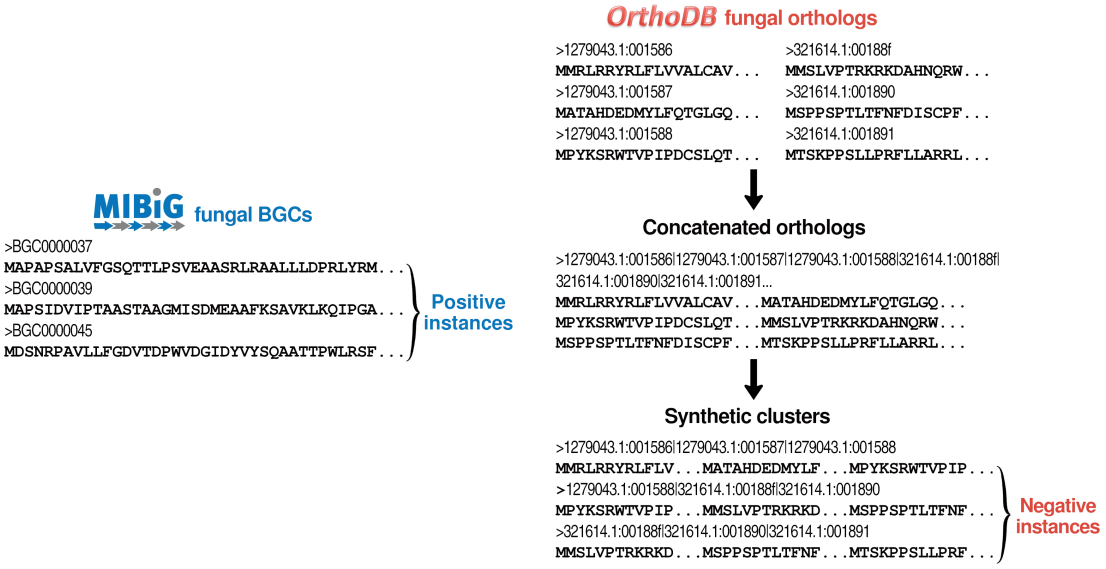


Figure 4.1: Example of positive instances and the process to generate synthetic negative instances from orthologs

The MIBiG fungal subset and the pool of OrthoDB synthetic negative clusters were then considered to generate fungal BGC datasets with different distributions of positive and negative instances. Among the MIBiG fungal subset the annotated BGC regions corresponded in average to ≈1% of the total genome length of an organism, which provides a hint on the imbalance in class distribution that can be seen in a real test case scenario. Due to the natural imbalance of BGC regions versus non-BGC regions in a genome, we are interested in analysing the performance of a supervised learning approach based on datasets with various distributions of positive and negative instances. To analyse this aspect, we generated fungal BGC

datasets with varying distributions by increasing the number of synthetic negative instances randomly selected from the OrthoDB synthetic negative clusters pool. Table 4.1 shows the positive vs. negative distributions in each dataset.

Table 4.1: Distribution of instances across fungal BGC datasets

	Train		**Validation**	
Dataset	**Pos**	**Neg**	**Pos**	**Neg**
50%-50%	160	160	40	40
40%-60%	160	240	40	60
30%-70%	160	373	40	93
20%-80%	160	640	40	160
10%-90%	160	1,440	40	360
05%-95%	160	3,040	40	760
01%-99%	160	15,840	40	3,960

To generate classification models based on a supervised learning method, we extracted Pfam (El-Gebali et al., 2019)[6] IDs from the positive and negative instances. All datasets were converted into `pfamtsv` format (Hannigan et al., 2019), which is required as input in the supervised learning approach applied in this work. For each dataset, 80% were randomly selected for the training phase, while 20% were held out for the validation phase, as shown in Table 4.1.

4.4.2 Test Datasets

To analyse the performance of classification models built based on fungal BGC datasets, we selected a fungal genome from the *Aspergillus* genus to represent a real test case scenario. *Aspergillus* is the most frequent genus among fungal species in MIBiG, together with *Penicillium*. For this evaluation we focused specifically on the *A. niger* species. *A. niger* is a genome of interest due to its biological diversity and major relevance to industrial processes (de Vries et al., 2017). In Inglis et al.

[6] `http://pfam.xfam.org`

(2013) the authors present manual annotation of BGCs in *Aspergilli*, among which a total of 79 BGCs are found in *A. niger*.

To generate candidate clusters for the test phase, we collected a manually curated *A. niger* genome sequence made publicly available through the Genozymes project[7]. We generated test candidate clusters by considering a sliding window of 30,000 nucleotides in the *A. niger* genome. The 30,000 sliding window length is defined based on the average length of the nucleotide sequence of MIBiG fungal BGCs. A similar approach was previously applied in fungal BGC discovery to generate virtual clusters (Umemura et al., 2013).

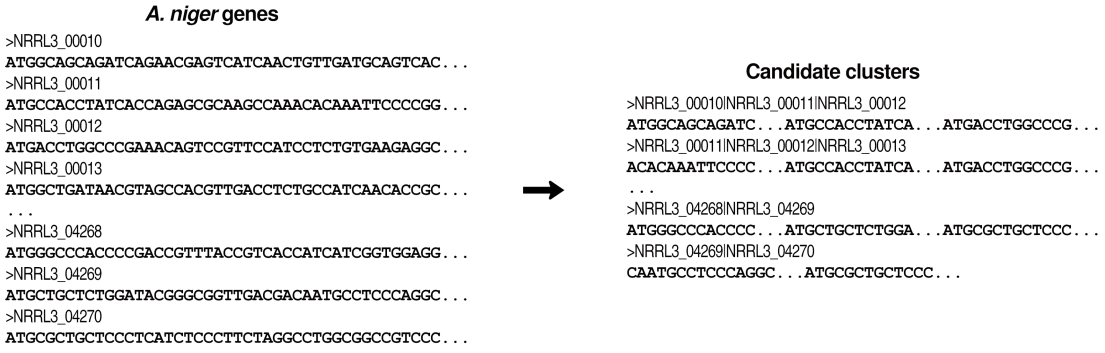


Figure 4.2: Example of *A. niger* candidate clusters generated for test phase

The 30,000 sliding window was shifted along the genome using either a 50% or a 30% overlap. The overlaps in a sliding window mean that each test candidate cluster will contain the last 15,000 nucleotides (if a 50% overlap) or the last 9,000 nucleotides (if a 30% overlap) of the immediate previous candidate cluster. With the strategy of generating candidate clusters using overlaps, we are more likely to cover regions in between two or more genes. Figure 4.2 shows an example of candidate clusters being generated from *A. niger* genes using overlaps. The test datasets based on a 50% overlap contains a total of 1,184 candidate clusters, while

[7]https://gb.fungalgenomics.ca/portal/

the test datasets based on a 30% overlap contains a total of 846 candidate clusters.

4.4.3 Classification Models

In this section we describe the methods applied to analyse the performance of a supervised learning approach using the fungal BGC datasets presented in Section 4.4.1 and the test data presented in Section 4.4.2. To generate classification models with our fungal BGC datases, we utilized the DeepBGC system (Hannigan et al., 2019). DeepBGC executable, source code and other resources are openly available[8]. Among these resources, there are pre-built BGC classification models and word2vec-based embeddings built using Pfam IDs, referred to as pfam2vec embeddings. In Hannigan et al. (2019) the authors explained that pfam2vec embeddings were trained based in a skipgram architecture with 100 dimensions and over 15,686 unique Pfam IDs. DeepBGC classification is based on a Bidirectional Long Short Term Memory (BiLSTM) neural network, for which the input are pfam2vec embeddings. In Hannigan et al. (2019) DeepBGC hyperparameters are described as a BiLSTM layer size of 128, dropout of 0.2, sigmoid activation, batch size of 64, 256 timestamps over 328 epochs, using Adam optimizer at a learning rate of 1e-4, with weighted binary cross-entropy loss. To generate classification models using fungal BGC datasets on the DeepBGC system we adopted the same hyperparameters described in Hannigan et al. (2019), as well as the pfam2vec embeddings as input for training. For each fungal BGC dataset, we have generated a different classification model using DeepBGC. Fungal BGC models are named by their positive instance percentage:

- `pos50` (50%-50%)
- `pos40` (40%-60%)

[8] https://github.com/Merck/deepbgc

- `pos30` (30%-70%)
- `pos20` (20%-80%)
- `pos10` (10%-90%)
- `pos05` (05%-95%)
- `pos01` (01%-99%)

To complement our analysis, we also analysed the performance of our test datasets using the four bacteria-based models made available at the DeepBGC repository:

- `deepbgc`
- `cf_o` (clusterfinder_original)
- `cf_r` (clusterfinder_retrained)
- `cf_g` (clusterfinder_geneborder)

According to the models description at the DeepBGC releases page[9] and Hannigan et al. (2019), the `deepbgc` model is based on the BiLSTM DeepBGC architecture and trained on a MIBiG dataset. The other models are built based on ClusterFinder (Cimermancic et al., 2014), which is a Hidden Markov Model (HMM). `cf_o` is a ClusterFinder HMM using original parameters; `cf_r` is also a ClusterFinder HMM but trained on a MIBiG dataset; and `cf_g` is a ClusterFinder HMM that switches stages only on gene borders, and trained on a MIBiG dataset.

[9] https://github.com/Merck/deepbgc/releases

4.5 Results and Discussion

We present here statistics and further details on the publicly available fungal BGC datasets proposed in this study. We also present results of validation and test phase obtained with classification models based on fungal BGC datasets and built using DeepBGC. Section 4.5.1 has further information and statistics on the fungal BGC datasets proposed in our work. In Section 4.5.2 we present results obtained at validation of training DeepBGC using the models `pos50`, `pos40`, `pos30`, `pos20`, `pos10`, `pos05`, and `pos01`. In Section 4.5.3 we present results obtained at test phase. For the sake of comparison, we also report the results on test data using BGC classification models provided by DeepBGC and built based on bacteria data, as listed in Section 4.4.3. All performance metrics are reported on the positive class only.

4.5.1 Fungal BGC datasets

The fungal BGC datasets proposed in this work are composed of positive and negative instances, as mentioned in Section 4.4.1. These datasets are suitable for performing binary classification to predict fungal BGCs, and are made publicly available at `https://github.com/bioinfoUQAM/fungalbgcdata`. The availability of such resource can potentially motivate the development of supervised learning approaches to tackle BGC discovery in fungi.

Positive instances in our datasets represent fungal BGCs from 52 different fungal genera. The variety of fungal genus is relevant to provide a large representation of BGC occurrence through different organisms. Additionally, the positive instances contain samples of over 10 different BGC types. Table 4.2 shows the different BGC types and a summary of fungal genera in our datasets. As the table shows, the most common BGC type is Polyketide synthase (PKS), followed by Non-ribosomal

peptide synthase (NRP) and Terpene synthase (TC). The presence of different fungal genus and BGC types in the datasets are important for representing a wide variety of BGC occurrences, and therefore contribute to building more robust supervised learning approaches.

BGC types	#
Alkaloid	3
Alkaloid/NRP	3
Alkaloid/TC	1
Alkaloid/NRP/TC	1
NRP	41
NRP/PKS	19
PKS	90
PKS/TC	5
RiPP	3
Saccharide	1
TC	23
Other	10
Total	**200**

BGC fungi genus	#	**BGC fungi genus**	#
Acremonium	1	Metacordyceps	1
Alternaria	5	Metarhizium	1
Armillaria	1	Monascus	3
Aspergillus	9	Mycosphaerella	1
Aureobasidium	1	Myrothecium	1
Beauveria	1	Neosartorya	1
Bipolaris	3	Neotyphodium	2
Botrytis	1	Nodulisporium	1
Byssochlamys	1	Paecilomyces	1
Cercospora	1	Parastagonospora	1
Chaetomium	2	Penicillium	13
Cladonia	2	Pestalotiopsis	1
Claviceps	2	Phoma	2
Diaporthe	1	Phomopsis	1
Elsinoe	1	Purpureocillium	1
Epichloe	2	Sarocladium	1
Fusarium	8	Shiraia	1
Glarea	1	Sordaria	1
Glycomyces	1	Sphaceloma	1
Hypholoma	1	Stachybotrys	1
Hypomyces	1	Starmerella	1
Isaria	1	Talaromyces	3
Lasiodiplodia	1	Tapinella	1
Lecanicillium	1	Tolypocladium	2
Leptosphaeria	1	Trichophyton	1
Malbranchea	1	Ustilago	1

Table 4.2: Fungal genera and BGC types in positive instances of datasets

Negative instances in our datasets represent synthetic gene clusters composed of fungal orthologs. By using fungal orthologs as source for the negative instances, we can generate synthetic gene clusters that depict the genomic profile of fungi. A total of 549 fungal species are present in orthologs composing our negative instances. The main fungal groups to which the orthologs belong to are shown in

Table 4.3, according to their taxonomy level. In this table we show the number of species clustered under different taxonomy levels (genus, family, order, or class), and the corresponding total of non-redundant orthologous genes for each group.

Group	**Taxonomy level**	**# Species**	**# Genes**
Aspergillus	Genus	30	309,629
Cryptococcus	Genus	7	44,028
Exophiala	Genus	7	67,291
Metarhizium	Genus	5	45,563
Penicilium	Genus	21	208,580
Phytophthora	Genus	6	89,378
Hypocreaceae	Family	7	66,815
Pleosporaceae	Family	9	94, 817
Polyporaceae	Family	6	61,584
Saprolegniaceae	Family	6	81,114
Trichocomaceae	Family	6	52,941
Agaricales	Order	25	293,149
Eurotiales	Order	60	608,401
Helotiales	Order	14	162,251
Hypocreales	Order	50	512,282
Mucorales	Order	15	164,081
Polyporales	Order	17	169,368
Sordariales	Order	8	66,549
Agaricomycetes	Class	77	912,187
Eurotiomycetes	Class	103	1,002,099
Microbotryomycetes	Class	9	59,326
Pucciniomycetes	Class	6	64,018
Saccharomycetes	Class	76	390,808
Tremellomycetes	Class	18	121,702
Ustilaginomycetes	Class	9	55,465

Table 4.3: Main fungal groups present in negative instances of datasets

The 52 fungal genera in positive instances together with the 549 fungal species in negative instance orthologs contribute to represent the genomic diversity in fungi, and therefore support the development of more robust classification models.

4.5.2 Validation performance

Table 4.4 shows validation metrics obtained with fungal BGC datasets. During training phase, all models using fungal BGC datasets had early stopping before completing the total 328 epochs. This could be a sign that the models were overfitting, a possible consequence due to the size of the datasets and the imbalanced distribution between the two classes.

The best performing model `pos50` is the one with the most balanced distribution of positive and negative instances. It yield Precision (P) of 0.598, Recall (R) of 0.995, and F-measure (F) of 0.747. Models `pos10`, `pos05`, and `pos01`, the ones with the most imbalanced distributions, had the lowest validation loss but also the lowest P, R and F.

Table 4.4: Validation performance using models built on proposed datasets

Model	**Epochs**	**Loss**	**P**	**R**	**F**
`pos50`	91	0.683	0.598	0.995	0.747
`pos40`	52	0.719	0.407	1	0.578
`pos30`	108	0.667	0.536	0.743	0.623
`pos20`	97	0.758	0.230	0.991	0.373
`pos10`	70	0.389	0	0	0
`pos05`	73	0.240	0	0	0
`pos01`	57	0.062	0	0	0

4.5.3 Test performance

The test phase show how the models would perform in a real case scenario, when a complete genome is being processed to predict candidate BGC regions. The dataset inputted in the test phase is composed of candidate clusters from the *A. niger* genome sequence, as described in Section 4.4.2. The performance on the test data is presented in two ways: gene metrics and cluster metrics. Gene metrics show P, R, and F for genes that belong to known BGCs. Cluster metrics

show P, R, and F for known BGCs where a minimum of one cluster gene must be correctly classified for the cluster to be predicted as positive. Tables 4.5 and 4.6 show the results on *A. niger* test datasets, with overlaps of respectively 50% and 30%. These results were obtained using classification models built with the fungal BGC datasets described in Section 4.4.1.

Table 4.5: Performance for *A. niger* test data using models built on fungal BGC datasets using 50% overlap

	Gene metrics			**Cluster metrics**		
Model	**P**	**R**	**F**	**P**	**R**	**F**
`pos50`	0.049	1.0	0.094	0.072	0.988	0.134
`pos40`	0.048	0.962	0.091	0.073	0.988	0.136
`pos30`	0.044	0.867	0.083	0.073	0.977	0.136
`pos20`	0.039	0.694	0.074	0.079	0.93	0.146
`pos10`	0	0	0	0	0	0
`pos05`	0	0	0	0	0	0
`pos01`	0	0	0	0	0	0

Table 4.6: Performance for *A. niger* test data using models built on fungal BGC datasets using 30% overlap

	Gene metrics			**Cluster metrics**		
Model	**P**	**R**	**F**	**P**	**R**	**F**
`pos50`	0.05	1.0	0.096	0.1	0.988	0.182
`pos40`	0.048	0.951	0.092	0.099	0.953	0.179
`pos30`	0.045	0.865	0.085	0.1	0.942	0.18
`pos20`	0.039	0.669	0.073	0.105	0.884	0.188
`pos10`	0	0	0	0	0	0
`pos05`	0	0	0	0	0	0
`pos01`	0	0	0	0	0	0

Results in the test phase show an important decrease in performance compared to the validation phase metrics. However the behaviors observed at the validation step also appear in test. Similarly to the validation phase, the more imbalanced models `pos10, pos05, pos01` did not predict any candidate cluster as positive.

This behavior happened with both test datasets of 50% or 30% overlap, and it could indicate that the model is sensitive to an imbalanced distribution of classes.

Also similarly to the validation phase the more balanced models `pos50, pos40, pos30, pos20` tended to predict most of candidate clusters as positives, leading to high recall but very low precision. Table 4.6 shows slightly better performance for P, R, and F compared to table 4.5. This behavior could indicate that using a 30% overlap in the test data is better suited for the task.

Following the results obtained with models based on fungal BGC datasets, we would like to also analyse the performance of DeepBGC models built using bacteria data on *A. niger* test datasets. Tables 4.7 and 4.8 show the results obtained on *A. niger* data with respectively 50% and 30% overlap using DeepBGC bacteria models.

Table 4.7: Performance for *A. niger* test data with 50% overlap using models provided by DeepBGC

	Gene metrics			**Cluster metrics**		
Model	**P**	**R**	**F**	**P**	**R**	**F**
`deepbgc`	0.074	0.972	0.138	0.114	0.988	0.205
`cf_o`	0.05	1.0	0.096	0.074	0.988	0.138
`cf_r`	0.056	0.997	0.106	0.083	0.988	0.153
`cf_g`	0.06	0.989	0.113	0.09	0.988	0.166

Table 4.8: Performance for *A. niger* test data with 30% overlap using models provided by DeepBGC

	Gene metrics			**Cluster metrics**		
Model	**P**	**R**	**F**	**P**	**R**	**F**
`deepbgc`	0.074	0.954	0.138	0.159	0.988	0.273
`cf_o`	0.051	0.984	0.096	0.103	0.988	0.187
`cf_r`	0.058	0.994	0.109	0.118	0.988	0.211
`cf_g`	0.061	0.992	0.116	0.126	0.988	0.223

Among all DeepBGC bacteria models, `deepbgc` performed best at both gene and cluster metrics, either using 30% or 50% overlap, with 0.273 F. The model `cf_o` showed the lowest performance, with 0.138 F. Models `cf_r` and `cf_g` showed in both cases better performance than `cf_o`. The results using DeepBGC trained models yield a similar tendency than that of the fungal BGC models: high recall but very low precision.

A loss in performance between validation and test results is evident, either when using fungal BGC based models or DeepBGC bacteria models.

As mentioned in Section 4.4.1, fungal BGCs seem to show larger genomic diversity, which possibly makes it more complex to perform BGC discovery in fungi if compared to bacteria. Therefore, performance is expected to be somehow affected by performing fungal BGC classification using bacteria-based models.

The dataset size at training time could also have had an impact on training `pos50, pos40, pos30, pos20, pos10, pos05` models, given DeepBGC classification approach. As the authors in Angermueller et al. (2016) explained, the suitability of deep learning approaches varies according to the problem at hand; and in cases when available data is limited conventional approaches could be relevant and more advantageous. As discussed in Section 4.4.1 the number of known fungal BGC data previously validated by curators is rather limited, which as a consequence will limit the size of fungal BGC datasets. It is possible and worth investigating that different classification methods, apart from a BiLSTM neural network as adopted in DeepBGC, will be better suited for handling fungal BGC discovery.

4.6 Conclusion

NPs are bioactive compounds that play a vital role in the production of a large variety of drugs, and the discovery of novel NPs can potentially benefit human health. Great effort has been put on identifying BGCs that are capable of producing NPs in plants, bacteria and fungi. Identifying BGCs is a challenging task, specially in fungi given the clusters genomic diversity.

Previous work on identifying BGCs in bacteria have resulted in a large variety of approaches and annotated data available. In fungi most previous approaches are based on data-driven methods and present a limited scope, such as covering only certain types of BGCs, or have been developed based on a single species data. The availability of new fungal BGC datasets could potentially motivate the development of new methods to identify BGCs in fungi. One example is supervised learning, a method that have shown to perform well in bacteria data.

In this work, we present new fungal BGC datasets to leverage supervised learning in the fungal BGC discovery task. These datasets are made publicly available at `https://github.com/bioinfoUQAM/fungalbgcdata`. The availability of such fungal BGC datasets can potentially motivate the development of binary classification approaches to tackle the BGC discovery task. We have shown results obtained on these fungal BGC datasets using a supervised learning approach developed for bacteria BGCs. We also analysed the performance of bacteria-based classification models applied on a fungal genome. The test performance on both fungal-based generated models or bacteria-based models was similar given precision (low) and recall (high) metrics using the same supervised learning method. This points to an opportunity to explore different supervised learning approaches than the one adopted by the DeepBGC system, that might be more suitable to handle fungal BGC datasets.

CHAPTER V

A SUPERVISED LEARNING FRAMEWORK FOR FUNGAL BGC DISCOVERY

The availability of benchmark datasets that represent diverse fungal genomic profiles relevant to BGC discovery allows the development of supervised learning approaches capable of identifying BGC regions in fungal genomes. This Chapter describes the methodology adopted to build TOUCAN, a supervised learning framework and post-processing methods to predict BGCs in fungi. The study presented in this Chapter was published in the NAR Genomics and Bioinformatics (NARGAB) journal under the title "TOUCAN: a framework for fungal biosynthetic gene cluster discovery". We note that a short version of this article was accepted at the 27th international conference in Intelligent Systems for Molecular Biology (ISMB), as a poster within the Machine Learning in Computational and Systems Biology (MLCSB) COSI, under the title "Towards accurate identification of Biosynthetic Gene Clusters in fungi". Article writing, approach implementation, experimental design and execution were performed by Hayda Almeida, under the supervision of professors Adrian Tsang and Abdoulaye Baniré Diallo. The annotation of high and medium MIBiG fungal protein domains was performed by Sylvester Palys, then PhD student under the supervision of professor Adrian Tsang. A printed version of this article is presented in the Appendix B.

5.1 Abstract

Fungal secondary metabolites (SMs) are an important source of numerous bioactive compounds largely applied in the pharmaceutical industry, as in the production of antibiotics and anticancer medications. The discovery of novel fungal SMs can potentially benefit human health. Identifying Biosynthetic Gene Clusters (BGCs) involved in the biosynthesis of SMs can be a costly and complex task, especially due to the genomic diversity of fungal BGCs. Previous studies on fungal BGC discovery present limited scope and can restrict the discovery of new BGCs. In this work we introduce TOUCAN, a supervised learning framework for fungal BGCs discovery. Unlike previous methods, TOUCAN is capable of predicting BGCs on amino acid sequences, facilitating its use on newly sequenced and not yet curated data. It relies on three main pillars: rigorous selection of datasets by BGC experts; combination of functional, evolutionary and compositional features coupled with outperforming classifiers; and robust post-processing methods. TOUCAN best performing model yield 0.982 F-m on BGCs regions in the *Aspergillus niger* genome. Overall results show that TOUCAN outperforms previous approaches. TOUCAN focuses on fungal BGCs but can be easily adapted to expand its scope to process other species or include new features.

5.2 Introduction

Secondary metabolites (SMs) are specialized bioactive compounds primarily produced by plants, fungi and bacteria. They represent a vital source for drug discovery: from anti-cancer, anti-viral, and cholesterol-lowering medications to antibiotics, and immunosuppressants (Chavali & Rhee, 2017). Genes involved in the biosynthesis of many SMs in fungi are co-localized in the genome, organized as clusters of genes (Kautsar et al., 2020), and known as Biosynthetic Gene Clusters (BGC). Typically BGCs are minimally composed of one or more synthase or syn-

thetase genes encoding backbone enzymes, which produce the core structure of the compound, and genes that encode tailoring enzymes, which modify the core compound to generate variants (Kjærbølling et al., 2020). Backbone enzymes determine the class of secondary metabolite produced by a BGC. Biosynthetic Gene Clusters may also contain other genes such as those encoding cluster-specific transcription factors, mitigating toxic properties, transporters, tailoring enzymes, and genes with hypothetical functions (Keller, 2019). Identifying new fungal BGCs can potentially lead to the discovery of new compounds that can serve as vital source for drug discovery (Macheleidt et al., 2016; de Vries et al., 2017). Despite the availability of a large volume of fungal genome sequence data, BGC discovery remains a challenging task (Chavali & Rhee, 2017) due to the diversity of fungal BGCs. Fungal BGCs have been shown to present noticeable differences in synteny and non-conservation of sequences even in related species or different strains of the same species (Kjærbølling et al., 2020), where clustered genes of the same SM can appear in different scaffolds among evolutionarily close species. Several studies have presented approaches to discover BGCs (Chavali & Rhee, 2017). Most approaches to identify fungal BGCs rely on probabilistic or data-driven methods, requiring as input genomic data (Takeda et al., 2014) combined with gene functional annotations (Wolf et al., 2016) and/or transcription data (Vesth et al., 2016; Umemura et al., 2013). Previous works also analysed fungal gene expression levels (Vesth et al., 2016), motif co-occurrence in promoters around anchor genes (containing backbone enzymes) (Wolf et al., 2016), compared expression levels of virtual gene clusters in conditions favourable to SM production (Umemura et al., 2013), and analysed homologous genes through sequence alignment and filtering syntenic blocks (Takeda et al., 2014). fungiSMASH (Blin et al., 2017) combines a probabilistic method (profile Hidden Markov Models (pHMMs) from proteins) and curated BGC detection rules, and can use tools such as Cluster Assignment by Islands of Sites (CASSIS) (Wolf et al., 2016) and ClusterFinder (Cimermancic

et al., 2014) to predict fungal BGC boundaries. These previous approaches present several limitations: overprediction of BGC length (Khaldi et al., 2010; Blin et al., 2017); dependence on manual curation (Vesth et al., 2016) which is expensive; or consider a very limited scope, potentially affecting the ability to process different BGC types or organisms (Khaldi et al., 2010; Takeda et al., 2014).

Approaches derived from supervised learning have shown to perform well when predicting bacterial BGCs (Agrawal et al., 2017; Hannigan et al., 2019). To our knowledge such methods have not been applied to identifying fungal BGCs. For instance RiPPMiner (Agrawal et al., 2017) based on Support Vector Machine (SVM) and Random Forest (RF) achieves 0.91 F-measure (F-m) in binary classification of ribosomally synthesized and post-translationally modified peptides. A recent approach, called DeepBGC, was designed to exploit Pfam (El-Gebali et al., 2019) domain embeddings to represent bacterial BGCs (Cimermancic et al., 2014) to feed a Bidirectional Long Short Term Memory (BiLSTM) neural network (Hannigan et al., 2019). DeepBGC relies also on post-processing methods such as merging consecutive BGC genes or filtering regions without known BGC protein domains. DeepBGC achieved a 0.923 Area Under the Curve (AUC) when predicting BGC positions in a set of 65 experimentally validated BGCs from six bacterial genomes, outperforming previous studies (Hannigan et al., 2019). When handling fungal BGC data, DeepBGC in its original version yield performance no higher than 0.2 F-m (Almeida et al., 2019), and when trained on fungal data underperformed previous methods such as fungiSMASH (Blin et al., 2017), as we show in Section 5.4. This could indicate that BGC discovery methods developed for bacteria may not be suitable for fungi due to the high diversity of fungal BGCs which are found to vary even among closely related species (Kjærbølling et al., 2020). Hence it is important to develop BGC discovery approaches dedicated to fungi, taking into account the specific characteristics of fungal BGCs, such as high diversity, BGC

components, as well as BGC and genome lengths which are usually longer than bacteria. Here we propose TOUCAN, a supervised learning framework to tackle BGC discovery in fungi that is based on a combination of heterogeneous biological feature types: k-mers, protein domains, and Gene Ontology (terms) to represent protein motifs and functions relevant to fungal BGCs.

5.3 Materials and methods

TOUCAN classification models were built based on a set of six open access fungal BGC datasets of varying distributions, a total of six classifiers, and two post-processing methods. In this Section we present the methodology adopted to develop TOUCAN models. TOUCAN predictions are validated based on a set of curated fungal BGCs.

5.3.1 Datasets

TOUCAN classification models were developed with comprehensive and exhaustive fungal BGC datasets presented in (Almeida et al., 2019) that are publicly available to support benchmarking of BGC discovery methods. The six fungal BGC training datasets are composed of different distributions of positive instances obtained from the Minimum Information about a Biosynthetic Gene cluster (MIBiG) (Kautsar et al., 2020) repository, and synthetic negative instances generated from OrthoDB (Kriventseva et al., 2018) orthologues. Fungal orthologous genes were previously applied in BGC discovery (Takeda et al., 2014). Orthologues can be a relevant source of negative instances since they represent conserved genes across species, while BGCs are known to show large genomic diversity even in closely-related species (Kjærbølling et al., 2020). To build negative instances, the amino acid sequences of OrthoDB fungal orthologous genes were concatenated using a fixed window size of 7,000 amino acids, which corresponds to the average

amino acid length of all positive instances from the fungal subset in MIBiG. This process generated a pool of training samples of 693,195 synthetic negative clusters (see (Almeida et al., 2019) for details). Studying datasets of various distributions could shed light on the impact of class imbalance in fungal BGC discovery which by nature presents a highly imbalanced scenario where only a small fraction of fungal genomes actually corresponded to BGCs (Almeida et al., 2019). To account for genomic diversity in fungi, positive instances in the six datasets represent more than 10 different BGC types and more than 100 fungal species. While negative instances were generated from a pool of orthologous genes representing $\approx$ 300 fungal species. To build and validate our models, we performed a random fixed split in each training dataset for which 80% of instances are dedicated to train and 20% for validation. Supplementary Table 1 shows the positive vs. negative distribution, and the train and validation splits in the six training fungal BGC datasets. A random fixed split allows us to evaluate the performances of the same train and validation sets under different parameters.

In the test phase we evaluated our classification models with six test datasets, generated similarly to Almeida et al. (2019), from a manually curated genome sequence of *Aspergillus niger NRRL3* (*A. niger*), available at `https://gb.fungalgenomics.ca/portal`. *Aspergillus niger* is an organism of interest for BGC discovery due to its relevance to industrial processes, and its ubiquitous distribution (de Vries et al., 2017). In this work 85 manually curated BGCs (Inglis et al., 2013) in *A. niger* will be considered as gold standard. Test candidate BGCs are generated by sequentially extracted genomic regions of *A. niger* with a sliding window of 5,000, 7,000 or 10,000 amino acids, with a 30% or a 50% overlap. The overlap of genomic regions allows us to cover BGC fragmented by the sliding windows. Multiple test datasets allow to analyse the impact of window lengths and overlaps when handling input data of test organisms, helping to determine

recommended parameters to obtain BGC predictions in new genome sequences. By generating test candidates based on a fixed sliding window length, new sequence data can be processed without requiring curation, genome annotation or gene models as input, unlike that of other BGC discovery tools. In Section 5.4, we report the performance obtained by the models using different window lengths and overlaps.

5.3.2 Features

To represent the fungal BGC dataset instances as feature vectors, we relied on heterogeneous biological features extracted from the protein sequences of dataset instances: k-mers, Pfam protein domains, and GO terms. Several feature types are combined to better represent the diverse genomic profiles in fungal BGCs and help build relevant discriminative models. Feature vectors are composed of number of occurrences of features per training instance. K-mers (a contiguous number of K amino acids appearing sequentially) are common features in genomic classification tasks (Vinje et al., 2015). We have extracted k-mers with varying lengths of $3 \leq K \leq 9$. K-mers appearing less than three times were discarded to reduce feature dimensionality, because presence of rare features could introduce bias (Yang & Pedersen, 1997). K-mer lengths were evaluated separately using validation sets to identify the K value yielding the best performance. Further details on validation of K values are provided in Section 5.4.

Pfam protein domains were previously applied in BGC discovery both in fungi (Khaldi et al., 2010) and in bacteria (Cimermancic et al., 2014; Hannigan et al., 2019). Protein domains are relevant features for BGC classification and can indicate the presence of backbone enzymes, a key component of BGCs (Inglis et al., 2013; Kjærbølling et al., 2020). We performed an analysis of protein domain distribution among positive instances in our datasets to understand their relevance as

features. In our analysis, Pfam protein domains extracted from positive instances were manually labeled by us as *high* (corresponding to a domain usually only present in BGCs) and *medium* (a domain usually present in, but not limited to BGCs). The complete list of *medium* and *high* annotated Pfam domains are presented in Supplementary Tables 2 and 3. Then we analysed all positive instance datasets for the presence or absence of such domains, shown in Supplementary Figure 1. This analysis highlights two important aspects: first the protein domain diversity in fungal BGCs; and second the presence of *high* domains shared by most BGCs suggesting that they share a structural pattern, most likely related to the presence of a backbone enzyme. The structural pattern yielded by the distribution of manually annotated protein domains in positive instances suggests that this feature type might carry an important discriminating power. Pfam domain features were extracted from our training datasets using the Pfam database.

GO term annotations were also modeled as features and obtained from our training instances using Swiss-Prot (UniProt Consortium, 2019). To identify corresponding GO terms, we performed a BLAST analysis of amino acid sequences from our dataset instances against the Swiss-Prot database composed of 560,292 reviewed entries (as of June 2019). BLAST parameters considered were *evalue* (expected value) $\leq 1e-4$ and *qcovs* (query coverage per subject) ≥ 50. A *qcovs* ≥ 50 could indicate relevant sequence similarity (Rost, 1999), since the alignment length would correspond to at least 50% of 7,000 amino acids for each match. We considered GO terms from all classes. GO term matches found were filtered for duplicates, and only unique GO terms were kept to represent dataset instances. Supplementary Table 4 shows the number of unique features per type, extracted from each training dataset and used to build our classification models. At this point, extracted features were all kept (except for K-mers that occur less than three times in a dataset), without relying on feature selection methods.

The feature order is not necessarily conserved during classification, and it is by all purposes processed in a Bag-Of-Words (BOW) manner. Considering that all extracted features can be relevant at this point since the experiments performed in our work are still a learning space of suitable parameters to tackle BGC discovery. Feature selection could therefore limit the exploration of potentially relevant attributes or combinations of features, but it might be valuable as a next step.

5.3.3 Classification methods

TOUCAN classification models were built with a total of six classifiers. We performed experiments with different classification algorithms to assess the performance of heterogeneous features and post-processing methods, and then identified the best configuration to tackle the BGC discovery task. Three classifiers were Support Vector Machine (SVM) classifiers: C-Support Vector (`svc`), Linear Support Vector (`lsvc`), and Nu-Support Vector (`nusvc`) classifiers. SVM classifiers were previously applied in BGC discovery (Agrawal et al., 2017). Default parameters were used for `svc` and `lsvc` during experiments, while for the `nusvc` classifier the `nu` parameter was adjusted in connection with the percentage of positive instances *pos* in a given dataset:

$$\texttt{nu}= \begin{cases} 0.5, & \text{if } pos \geq 30\% \\ \dfrac{pos}{100}, & \text{otherwise} \end{cases} \tag{5.1}$$

The other three classifiers were a Multilayer Perceptron (`mlp`), Logistic Regression (`logit`) and Random Forest (`randomf`). While `logit` classifier can provide a baseline model for the task, neural networks (Hannigan et al., 2019) and `randomf` (Agrawal et al., 2017) were also previously applied in BGC discovery. Also for `mlp`, `logit`, and `randomf` default parameters were kept but could however be optimized to suit specific experiments if needed. These six classifiers were

evaluated independently during our experiments.

5.3.4 Post-processing methods

Predictions of candidate BGCs outputted by TOUCAN are post-processed to improve output precision. Post-processing methods adopted in our work were greedy approaches, such as in PRISM (Skinnider et al., 2015) which identifies bond-forming domains and expands cluster boundaries on either ends of such domains. Unlike PRISM, TOUCAN does not require curation as input, and relies on classification models to identify potential BGC regions in which post-processing methods can be applied, facilitating its use on newly sequenced or not yet annotated genomes. The post-processing methods `succ` and `merge` are shown in Algorithm SuccessiveMerge, and aim to address potential cluster boundary limitations (over or under estimation) common in previous approaches (Khaldi et al., 2010; Blin et al., 2017).

BGC region length can vary greatly among fungal MIBIG BGCs: for an x number of amino acids, x can vary such as $195 \leq x \leq 62,079$, with a standard deviation $\sigma(x) \approx 6013.73$ and mean $\overline{x} \approx 7,033$. In this work, a fixed amino acid length to generate test candidate instances from an organism genome is applied. Both `succ` and `merge` post-processing help to overcome the shortcoming in cases where cluster regions have limited boundaries. The `succ` post-processing gives to a $nbSucc$ of successive predictions the same $confidence$ prediction score of a positive prediction ($confidence \geq threshold$). The `merge` post-processing merges a $nbSucc$ of successive predictions of a positive prediction ($confidence \geq threshold$) into a single positive prediction. For both `succ` and `merge` we considered $0 \leq nbSucc \leq 3$, set as an arbitrary parameter for a first evaluation of post-processing methods. Both post-processing methods were applied only if $nbSucc$ successive predictions were also not positive.

Algorithm 1 - SuccessiveMerge: Compute successive or merged positives

Data: P, list of predictions.

Result: P', list of post-processed predictions.

```
begin
    nbSucc;          // number of successive predictions
    doMerge;         // boolean, True to perform merge
    threshold;       // confidence threshold, default 0.5
    merges;          // list of merged predictions

    for i ∈ P do
        count = 1
        if i.confidence ≥ threshold then
            merged.ID = i.ID
            while count ≤ nbSucc do
                successor = P[i + count]
                if doMerge then
                    merged.ID += successor.ID
                else if successor.confidence < threshold then
                    successor.confidence = i.confidence
                    P.update(successor)
                count++
            if doMerge then
                merged.confidence = i.confidence
                merges.add(merged)
        else
            merges.add(i)
    if doMerge then
        return merges
    return P
```

5.3.5 Evaluation metrics

TOUCAN classification models were assessed in terms of Precision (P), Recall (R), F-measure (F-m) and a *clusterScore* metric. To compute P, R and F-m, we considered as True Positives (TP) BGC candidates predicted as positive that have at least one gene that matches a gold standard BGC. The *clusterScore* represents the coverage of expected gold standard BGC genes within a candidate BGC, where $0 \leq clusterScore \leq 1$, and was computed for each BGC canidate predicted positive. To compute the *clusterScore* for a BGC candidate C and its gold standard BGC match M, we first counted the number of $geneMatches$ in C, meaning the number of M genes in C. We then computed a similarity value `sim` between all pairs of genes in the disjunctive union $C \triangle M$, and add to the *clusterScore* the best `sim` obtained for the unmatched $M - C$ genes. Computing the `sim` value allows us to account for the possible presence of gold standard orthologues among unmatched genes in a BGC candidate predicted positive. The `sim` value represents a percent identity *pident* obtained through a local alignment with BLAST between two genes, using cutoffs of minimum $pident \geq 20$ and minimum query coverage $qcov \geq 10$. The *clusterScore* for a BGC candidate C was normalized by the number of genes in its gold standard BGC match M. Algorithm SIMILARITY shows the computation of `sim` scores, while Algorithm CLUSTERSCORE shows the computation of *clusterScores*. We analysed the *clusterScore* of TOUCAN predicted positives compared to state-of-the-art methods in Section 5.4.

Algorithm 2 - SIMILARITY: Compute similarity score of genes

Data: *pred.genes*, genes in a positive predicted cluster

gold.genes, genes in a gold cluster

genesFound, predicted genes matching gold genes

similarities, list of similarities between genes

Result: *similarityScore*, similarity score

```
begin
    pairs = ∅                          // best scores for pairs of genes
    pairedGenes = ∅                    // predicted and gold genes paired
    pred.genes = pred.genes − genesFound
    gold.genes = gold.genes − genesFound
    for i ∈ gold.genes do
        totalScore = 0
        for j ∈ pred.genes do
            score = similarities.get(i, j)
            if score ≥ totalScore then
                pair.gene1 = i
                pair.gene2 = j
                pair.score = score
                pairs.add (pair)
                totalScore = score
    // Sort pairs by score in descending order
    pairs = sortDescScore(pairs)
    for pair ∈ pairs do
        if pair.gene2 not ∈ pairedGenes then
            similarityScore+ = pair.score
            pairedGenes.add (pair.gene2)
    return similarityScore
```

Algorithm 3 - ClusterScore: Compute *clusterScore* evaluation metric

Data: P, list of prediction tuples $(geneID, clusterID, confidenceVal)$

G, list of gold tuples $(goldGeneID, goldClusterID)$

`useSimilarity`, boolean value for considering similarity scores

Result: *clusterScores* for a list of predictions.

begin

- $threshold$; `// confidence threshold, default 0.5`
- $geneMatches$; `// count of gene matches`
- $geneMatches^*$; `// count gene matches with similarity scores`
- $clustersFound$; `// list of gold clusters found`
- `// Compute successive or merged positives`
- $P =$ `computeSuccOrMerge`(P)
- `// Retrieve list of only positive predictions`
- $posGenes, posClusters =$ `unfoldPredictions`(P)
- **for** $gold \in G$ **do**
  - **for** $gene \in gold.genes$ **do**
    - `// Find occurrences of gene in predictions`
    - $pred =$ `getAllOccurrences`$(gene, P)$
    - **if** $pred \in posGenes$ **then**
      - $geneMatches+ = 1$
      - $clustersFound$.`add`$(pred)$
  - **if** $gold.confidence \geq threshold$ *&* `useSimilarity` **then**
    - $geneMatches^* = geneMatches+$`similarity`$(gold.genes, pred.genes, genesFound)$
  - $clusterScore = \frac{geneMatches^*}{\texttt{length}(gold.genes)}$
  - $clusterScores$.`add`$(gold.ID, clusterScore)$
- **return** *clusterScores*

5.3.6 State-of-the-art performance comparison

The performance of TOUCAN models was compared to results obtained by two state-of-the-art tools: fungiSMASH (Blin et al., 2017) and DeepBGC (Hannigan et al., 2019) (version 0.1.18 and models as of February, 2020 available at `https://github.com/Merck/deepbgc`). The experiments with fungiSMASM were

performed with its three strictness levels: relaxed, strict, and loose, and with default parameters for its extra feature options (as of January, 2020): "KnownClusterBlast", "ActiveSiteFinder", and "SubClusterBlast". DeepBGC focuses on bacterial data and is based on a BiLSTM neural network and Pfam domain embeddings. A total of three DeepBGC classification models are applied in this work: one with original DeepBGC training dataset and hyperparameters, as in (Hannigan et al., 2019); one built with DeepBGC original hyperparameters and our best performing training dataset; and one built with our best performing training dataset and fungal optimized hyperparameters (thanks to the authors) (see Supplementary Table 7 for original and fungal optimized hyperparameters).

5.4 Results

We present here results obtained with TOUCAN, a supervised learning framework to discover fungal BGCs. To identify the best configuration to tackle BGC discovery in fungi, we designed, trained and assessed several classification models combining heterogeneous biological features, datasets of various distributions, classifiers, and post-processing methods, as described in Section 5.3. Validation results are drawn on held-out training instances corresponding to 20% of each training dataset. The performance of TOUCAN was assessed on test datasets of a gold standard of 85 manually annotated *A. niger* BGCs (Inglis et al., 2013). The focus here is BGC discovery, hence, the model is optimized to correctly identify positive instances, rather than the negative ones. Thus results were reported for the positive class.

Feature importance and performance on validation datasets To identify the most suitable K for k-mer features within $3 \leq K \leq 9$ we performed a set of experiments on all six datasets and six classifiers, as presented in Sections 5.3.1

and 5.3.3. Performance of k-mer models on our validation sets is shown in Supplementary Figure 2. In general better performance was achieved with $K = 6$, which was thus the K value considered for our following experiments. We also performed an analysis of feature importance across training datasets, obtained with a `randomf` classifier, with default parameters. Table 5.1 shows the top 15 ranked features across training datasets.

Table 5.1: Top 15 features ranked by importance for each training dataset, from completely balanced (50% positive, 50% negative) to most imbalanced (05% positive, 95% negative). Highlighted features appeared in multiple datasets.

Training dataset distribution					
50 - 50%	**40 - 60%**	**30 - 70%**	**20 - 80%**	**10 - 90%**	**05 - 95%**
PF00698.21	**PF00698.21**	GO:0008168	**HGTGTQ**	PF00109.26	**TACSSS**
PF00668.20	**HGTGTQ**	**HGTGTQ**	GO:0008152	GO:0044550	GTGTQA
ADGYCR	**GO:0031177**	**GQGAQW**	PF00550.25	LYRTGD	GYARGE
GO:0016491	**GAGTGG**	GYCRAD	IDTACS	**VFTGQG**	**GO:0046148**
FDGYRF	VEMHGT	**GAGTGG**	PF02458.15	NFSAAG	TGDLAR
GO:0016740	**VFTGQG**	QQRLLL	DTACSS	VEAHGT	SINSFG
MHGTGT	**PF00668.20**	**TACSSS**	VTLSGD	GO:0043041	DPQQRL
DTACSS	GO:0016874	PF02801.22	FTGQGA	GHSLGE	LFTSGS
GO:1900557	YKTGDL	**GO:0009058**	PF08242.12	AYEALE	NSFGFG
GO:0009058	GO:0019184	**GO:0046148**	**AYGPTE**	**GO:0016491**	CDTAVA
GRFFAA	GO:0043042	GO:0047462	GO:0004315	TQVKIR	FDASFF
PF14765.6	PGRFFA	GEYAAL	**GO:0031177**	GO:0046500	**AYGPTE**
MDPQQR	**MHGTGT**	GO:0005829	KLRGFR	**DTACSS**	YILFTS
FTSGST	GO:1900790	PFAFHS	GO:0016021	GO:0032259	AIVLAG
GQGAQW	VEIGPH	LHSLEA	PF00067.22	DTFVRC	AVVGHS

Features appearing on the top 15 of multiple datasets are highlighted. Protein domain feature names start with *PF*, GO term feature names start with *GO*, and the other features are 6-mers. We can observed that every protein domain feature appearing among the top ranked of all datasets belonged either to the *high* or *medium* manually annotated domains, even though (non-*high* and non-*medium* domain features are also included in our feature set. Moreover, while GO terms represent $\approx$ 30% of all top 15 ranked features, they make up for at most 0.7%

of total features. This possibly indicates their strong discriminating power in the task. After evaluating feature importance, we trained several classification models combining the feature types for each classifier and training dataset distribution. For each training dataset distribution, a random fixed split, designating 80% of its instances were selected for train and 20% for validation, as mentioned in Section 5.3.1. The top F-m performances on validation sets per training dataset are shown in Supplementary Table 5. During validation we noted that models built with three feature types outperformed models using one feature type, such as the ones built when evaluating the most suitable K-mer length.

Validation performance seems to be overall affected by the instance distribution: more imbalanced datasets show lower F-m compared to more balanced ones. When analysing MIBiG fungal BGCs, only ≈1% of a genome sequence would correspond to cluster regions (Almeida et al., 2019), so utilising more balanced training data could provide better performance than using real case scenario distributions. We selected the dataset with the best F-m average performance, which was the most balanced (50-50%), to perform further evaluation with hyperparameter optimization through a grid search, followed by cross validation (CV) classification for all six classifiers. Best performing hyperparameters to maximize F-m for each classifier were listed in Supplementary Table 8. A 5-fold CV was performed with optimized hyperparameters on the 50%-50% dataset instances, randomly split between train and validation at each fold. Supplementary Table 6 shows the average performances on the 5-fold CV for each classifier.

Table 5.2: TOUCAN best performing models per test set sliding windows and overlaps in *A. niger*

Sliding window	**Overlap**	**Training set**	**Classifier**	**Post-process**	**P**	**R**	**F-m**
10,000	50%	50-50%	mlp	merge3	1	0.871	**0.931**
10,000	50%	40-60%	mlp	merge3	1	0.753	0.859
10,000	50%	30-70%	mlp	merge2	1	0.706	0.828
10,000	50%	20-80%	mlp	merge2	1	0.706	0.828
10,000	50%	10-90%	mlp	merge3	1	0.647	0.786
10,000	50%	05-95%	mlp	merge3	1	0.447	0.618
7,000	50%	50-50%	logit	merge3	0.929	0.765	0.839
7,000	50%	40-60%	logit	merge3	1	0.741	0.851
7,000	50%	30-70%	mlp	merge3	0.969	0.729	0.832
7,000	50%	20-80%	mlp	merge3	1	0.741	0.851
7,000	50%	10-90%	mlp	merge3	1	0.694	0.819
7,000	50%	05-95%	mlp	merge3	1	0.647	0.786
5,000	50%	50-50%	logit	merge3	0.817	0.788	0.802
5,000	50%	40-60%	logit	merge3	0.914	0.753	0.826
5,000	50%	30-70%	logit	merge3	0.953	0.718	0.819
5,000	50%	20-80%	logit	merge3	1	0.718	0.836
5,000	50%	10-90%	mlp	merge3	0.913	0.741	0.818
5,000	50%	05-95%	mlp	merge3	0.923	0.706	0.800
10,000	30%	50-50%	mlp	merge3	1	0.847	0.917
10,000	30%	40-60%	mlp	merge3	1	0.741	0.851
10,000	30%	30-70%	mlp	merge2	1	0.694	0.819
10,000	30%	20-80%	mlp	merge2	1	0.671	0.803
10,000	30%	10-90%	mlp	merge3	1	0.6	0.750
10,000	30%	05-95%	mlp	merge3	1	0.459	0.629
7,000	30%	50-50%	mlp	merge3	0.95	0.906	0.928
7,000	30%	40-60%	mlp	merge3	1	0.824	0.903
7,000	30%	30-70%	mlp	merge2	1	0.741	0.851
7,000	30%	20-80%	mlp	merge3	1	0.741	0.851
7,000	30%	10-90%	lsvc	merge3	1	0.553	0.712
7,000	30%	05-95%	mlp	merge3	1	0.635	0.777
5,000	30%	50-50%	logit	merge3	0.908	0.812	0.857
5,000	30%	40-60%	logit	merge3	0.985	0.788	0.876
5,000	30%	30-70%	logit	merge3	1	0.753	0.859
5,000	30%	20-80%	mlp	merge3	0.985	0.776	0.868
5,000	30%	10-90%	mlp	merge3	0.984	0.729	0.838
5,000	30%	05-95%	mlp	merge3	1	0.706	0.828

TOUCAN performance on test datasets We assessed TOUCAN models on six test datasets with amino acids sliding window lengths of 5,000, 7,000, 10,000, with overlaps of 50% and 30%, as described in Section 5.3.1. Candidate BGC predictions on the test data were obtained with TOUCAN classification models built using the six training dataset distributions with fixed train and validation splits, three feature types, and six classifiers. We then processed TOUCAN predicted candidate BGCs with post-processing methods `succ` and `merge`, considering $0 \leq nbSucc \leq 3$.

Table 5.2 shows for the positive class the best F-m obtained for each test dataset among all training dataset distributions. The highest 0.931 F-m was obtained by a model built with a 50-50% distributed training set, a `mlp` classifier, and a `merge3` post-processing. The best F-m was achieved with 10,000 amino acid sliding window test datasets. Regarding classifiers, `mlp` and `logit` yielded best performance followed less often by `lsvc`. As mentioned in Section 5.3.3, default parameters were used when performing our experiments. Tuning the classifier parameters may impact on the performance, but this is not the focus of this study. Overall results showed that a 30% overlap seems to be more advantageous for all sliding window lengths, even though the best F-m was achieved with test candidates generated based on a 50% overlap. The training set distribution seemed to have little influence on test candidates with a sliding window length of 5,000 amino acids, showing only a small variation on F-m for both 30% and 50% overlap. Less balanced training distribution seemed to affect performance more for candidates with a sliding window length of 10,000 amino acids, with a F-m varying from 0.618 F-m to 0.931 F-m when using 50% overlap, and from 0.629 F-m to 0.917 F-m when using 30% overlap.

We selected the best performing test datasets (10,000 amino acid sliding window) to carry an evaluation using 5-fold CV classification models based on the best per-

forming training set (50%-50%). The predicted BGC candidates obtained with CV classification models were also processed with TOUCAN post-processing methods `succ` and `merge`, in the same manner as the models presented in Table 5.2. The best performance results obtained with the 10,000 amino acid sliding window test data among all 5-fold CV classification models are shown in Table 5.3.

Table 5.3: TOUCAN best performances for the completely balanced (50% positive, 50% negative) CV models on *A. niger* test sets generated with a 10,000 amino acid sliding window.

Training set	**Sliding window**	**Overlap**	**Classifier**	**Post-process**	**P**	**R**	**F-m**
50-50%	10,000	50%	`svc`	`merge3`	0.941	0.941	0.941
50-50%	10,000	30%	`randomf`	`merge3`	1	0.965	**0.982**

As shown in Table 5.3, the 5-fold CV classification models improved to a 0.982 F-m from the previously best 0.931 F-m achieved with models based on fixed train and validation splits. Performance results in Tables 5.2 and 5.3 show TOUCAN models discriminative power to identify candidate BGC regions from non-BGC regions. Our results also demonstrate TOUCAN models capacity of obtaining relevant BGC predictions on new or non-annotated genomes in test dataset instances generated solely based on sliding windows of fixed amino acid length. This aspect distinguishes TOUCAN from previous approaches that rely on gene models and other genomic annotations as input (Agrawal et al., 2017; Hannigan et al., 2019).

Performance comparison with DeepBGC We compared the performance of three DeepBGC classification models using the 10,000 amino acid sliding window test datasets, which yield the best F-m with TOUCAN. As mentioned in Section 5.3, two out of the three DeepBGC models were trained using the best performing constructed training dataset(50%-50% dataset in this case). The Deep-

BGC hyperparameters applied in this comparison are also listed in Section 5.3. As shown in Almeida et al. (2019), during validation phase the DeepBGC model trained using the original hyperparameters and the 50%-50% training dataset had early stopping at epoch 109, from the original total of 328 epochs, as applied in Hannigan et al. (2019).

Table 5.4 shows P, R, and F-m performances of the three DeepBGC models for the positive class on the test dataset with a 50% or 30% overlap. DeepBGC models built with original hyperparameters yielded high recall but very low precision, consequently leading to F-m metrics lower than 0.3 for either models based on the 50%-50% training set or based on DeepBGC original data. Models built with fungal optimized hyperparameters yielded a noticeable performance improvement, with a 0.627 F-m.

Table 5.4: Performance metrics of DeepBGC models for *A. niger* test sets generated with 10,000 amino acid sliding window.

Training dataset	**DeepBGC model**	**Sliding window**	**Overlap**	**P**	**R**	**F-m**
DeepBGC	original	10,000	50%	0.114	1	0.205
DeepBGC	original	10,000	30%	0.159	1	0.274
50%-50%	original	10,000	50%	0.075	1	0.140
50%-50%	original	10,000	30%	0.105	1	0.191
50%-50%	fungal	10,000	50%	0.464	0.765	0.578
50%-50%	fungal	10,000	30%	0.580	0.682	**0.627**

For each of the three DeepBGC models, the test sets using a 30% overlap resulted in better performance than the ones using a 50% overlap. DeepBGC performance on predicting fungal BGCs shows high recall but very low precision, which consequently lead to F-m metrics lower than 0.2. The most imbalanced models classified all test candidates as negative, which could be a sign of the model

trying to optimize accuracy towards the majority class. Originally, DeepBGC was developed to predict bacterial BGCs, for which much more data is available compared to fungal BGCs. The larger amount of bacterial BGC data available benefits the development of supervised learning approaches. Fungal BGC data is more scarce, which makes it challenging to build robust classification models. Supervised learning approaches that fit bacteria may not be suitable to discover BGCs in fungi (Almeida et al., 2019).

Performance comparison with fungiSMASH We compared the performance of fungiSMASH on the same 10,000 amino acid sliding window test datasets used to compare with DeepBGC. The fungiSMASH parameters considered in this comparison are described in Section 5.3. fungiSMASH predictions are also assessed in terms of P, R, and F-m which are shown for the positive class in Table 5.5. fungiSMASH best performance yielded a 0.571 F-m when using a 50% overlap and 0.692 F-m when using a 30% overlap, both under relaxed strictness. As expected, loose strictness results in higher recall and lower precision, while a strict parameter results in higher precision but lower recall.

Table 5.5: Performance metrics of fungiSMASH models for *A. niger* test sets generated with 10,000 amino acids sliding window.

fungiSMASH strictness	**Sliding window**	**Overlap**	**P**	**R**	**F-m**	**Overlap**	**P**	**R**	**F-m**
relaxed (default)	10,000	50%	0.470	0.729	0.571	30%	0.649	0.741	**0.692**
strict	10,000	50%	0.471	0.576	0.519	30%	0.671	0.600	0.634
loose	10,000	50%	0.435	0.788	0.561	30%	0.591	0.800	0.68

Similar to TOUCAN models, fungiSMASH seems to yield generally better performance on 30% overlap test candidates. fungiSMASH showed in general a more stable performance predicting fungal BGCs compared to DeepBGC. Apart from being based on a different approach than DeepBGC, fungiSMASH was developed

focusing on fungal organisms. The difference in performance between the bacteria-focused approach of DeepBGC and the fungal-focused approach of fungiSMASH may be another indication that BGC discovery is a complex task, and can benefit from approaches built to target related organisms.

TOUCAN yields reproducible performance on *Aspergillus nidulans*

To assess TOUCAN reproducibility we assessed the performance of its models in the *A. nidulans* genome. As in *A. niger*, *A. nidulans* is a species known as an important source of BGCs (Inglis et al., 2013; Kjærbølling et al., 2020). Previous work on manual annotation of BGCs in Aspergilli (Inglis et al., 2013) identified a total of 70 BGCs in *A. nidulans*, which are considered as gold standard for this analysis. To obtain candidate BGCs for testing, *A. nidulans* genome sequence was processed in the same manner as *A. niger*. Test candidate BGCs for *A. nidulans* were obtained by extracting genomic regions sequentially from its genome, using amino acid sliding windows of 10,000 amino acids that overlap by 30% and 50%. The analysis on *A. nidulans* used the best performing model parameters previously established in *A. niger*: 50%-50% dataset, hyperparameter optimization, and 5-fold CV.

Table 5.6: Best performances per overlap of TOUCAN compared to fungiSMASH and DeepBGC for *A. nidulans* test sets generated with 10,000 amino acid sliding window.

System	Model	Sliding window	Overlap	P	R	F-m
TOUCAN	`lsvc` + merge3	10,000	50%	0.919	0.814	0.864
TOUCAN	`svc` + merge3	10,000	30%	0.953	0.871	**0.910**
fungiSMASH	relaxed (default)	10,000	50%	0.550	0.786	0.647
fungiSMASH	relaxed (default)	10,000	30%	0.775	0.786	**0.780**
DeepBGC	50%-50% fungal	10,000	50%	0.473	0.629	0.540
DeepBGC	50%-50% fungal	10,000	30%	0.631	0.586	**0.607**

Table 5.6 shows TOUCAN best performance results among all six classifiers and post-processing methods for the *A. nidulans* 10,000 amino acid sliding window test sets. For comparison, we evaluated *A. nidulans* BGC predictions obtained with the best fungiSMASH and DeepBGC models on the same test sets, for which the results are also shown in Table 5.6. We observed that similar F-m performance metrics were achieved for *A. nidulans* and *A. niger*. TOUCAN and DeepBGC, both based on supervised learning, yielded the least F-m variation on the results obtained for the two *Aspergillus* species, suggesting that due to their generalization ability, supervised learning approaches may be a suitable approach to tackle BGC discovery.

TOUCAN True Positive predictions improve coverage of BGC genes

We compared TP predictions (BGC candidate predicted positives that have at least one gene matching a gold standard BGC) obtained from best performing models in *A. niger* and *A. nidulans* for TOUCAN (0.982 F-m and 0.910 F-m, respectively) versus fungiSMASH (0.692 F-m and 0.780 F-m, respectively), and DeepBGC (0.620 F-m and 0.607 F-m, respectively). First we analysed the distribution of *clusterScores* computed for each BGC candidate predicted positive. Figure 5.1 shows the *clusterScore* distribution in *A. niger* and *A. nidulans* TP predictions obtained with TOUCAN, DeepBGC and fungiSMASH best models. We observed that compared to the other tools, TOUCAN TP predictions more often present a $clusterScore = 1$, meaning that TOUCAN predictions better encompass genes matching gold standard BGCs, possibly as a result of TOUCAN `merge` post-processing. Although `merge` post-processing leads to more comprehensive predictions, it could result in overprediction of cluster boundaries.

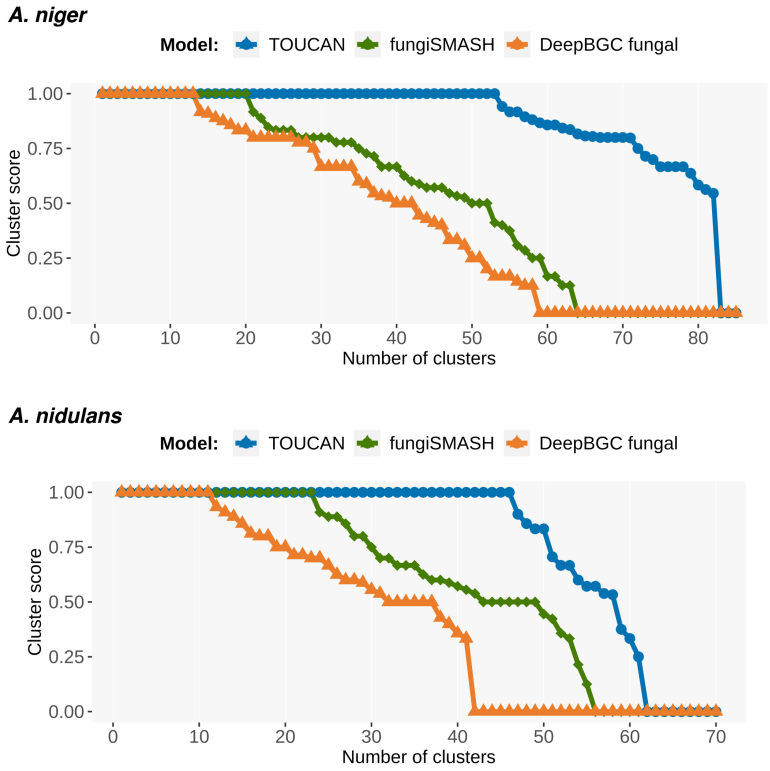


Figure 5.1: Distribution of *clusterScores* among True Positive predictions in *A. niger* and *A. nidulans* genomes. *clusterScore* distribution was computed for best performing models of each system (*A.niger*: TOUCAN: 0.982 F-m, DeepBGC: 0.627 F-m, fungiSMASH: 0.692 F-m; *A. nidulans*: TOUCAN: 0.910 F-m, DeepBGC: 0.607 F-m, fungiSMASH: 0.780 F-m).

To mitigate, filtering methods could be applied to refine candidate cluster regions, and also as an opportunity to fine-tune TOUCAN predictions to specific genus or species of interest. One possible way to apply targeted filtering is to rely on manually curated annotations of relevant features, such as the annotated *high* and *medium* Pfam protein domains shown in Section 5.3.2. We also analysed the presence of backbone enzymes within genes of TP predictions. Backbone enzymes are considered as the BGC core (Kjærbølling et al., 2020), playing a key role on its biosynthesis and defining the BGC compound to be produced (Inglis et al., 2013). We mapped the presence and absence of backbone genes among

TOUCAN, DeepBGC and fungiSMASH best models TP predictions.

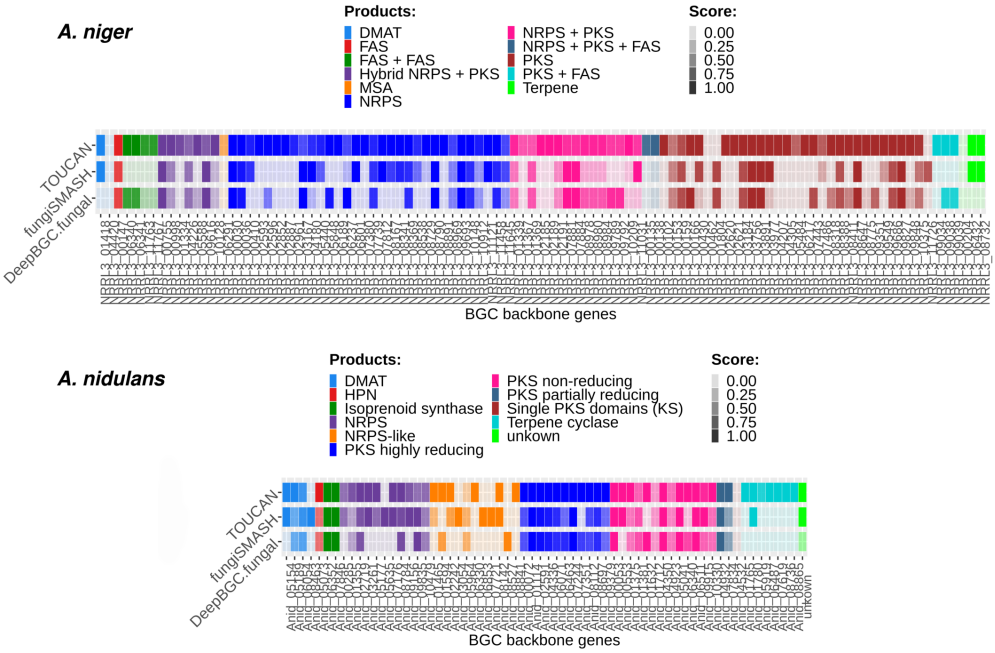


Figure 5.2: Presence of backbone enzymes among positive predictions in *A. niger* and *A. nidulans* genomes. Each backbone enzyme is shown per the gene ID it is associated with and the *clusterScore* assigned to the candidate predicted BGC.

Figure 5.2 shows backbone genes and product types found in *A. niger* and *A. nidulans*, respectively. Scores in Figure 5.2 (or the color intensity) correspond to the *clusterScore* computed for the predicted BGC. Backbone enzyme genes were present in 86.6% of all TOUCAN TP predictions for *A. niger*, versus 76.2% in fungiSMASH and 75.9% in DeepBGC. For *A. nidulans* 93.5% of TOUCAN TP predictions found backbone enzymes, versus 89% for fungiSMASH and 82.9% for DeepBGC.

5.5 Discussion

Secondary metabolites are bioactive compounds that play a vital role in the production of various drugs. Discovery of novel fungal BGCs can potentially benefit human health. In this work we presented TOUCAN, a supervised learning frame-

work for fungal BGC discovery. We evaluated classification models based on fungal BGC datasets of various distributions, six classifiers, heterogeneous biological features, and three post-processing methods. TOUCAN best performing model achieved 0.982 F-m in *A. niger* and 0.910 f-m in *A. nidulans*, outperforming previous methods. The results obtained with TOUCAN models could indicate that standard supervised learning approaches are suitable to tackle BGC discovery. TOUCAN outperformance is possibly due to a combination of factors: combining feature types, evaluating the impact of different class distributions during training, and post-processing candidate BGC predictions. `merge` post-processing can help identify regions that might have been missed, but in certain cases it may potentially lead to overestimation of predicted cluster boundaries.

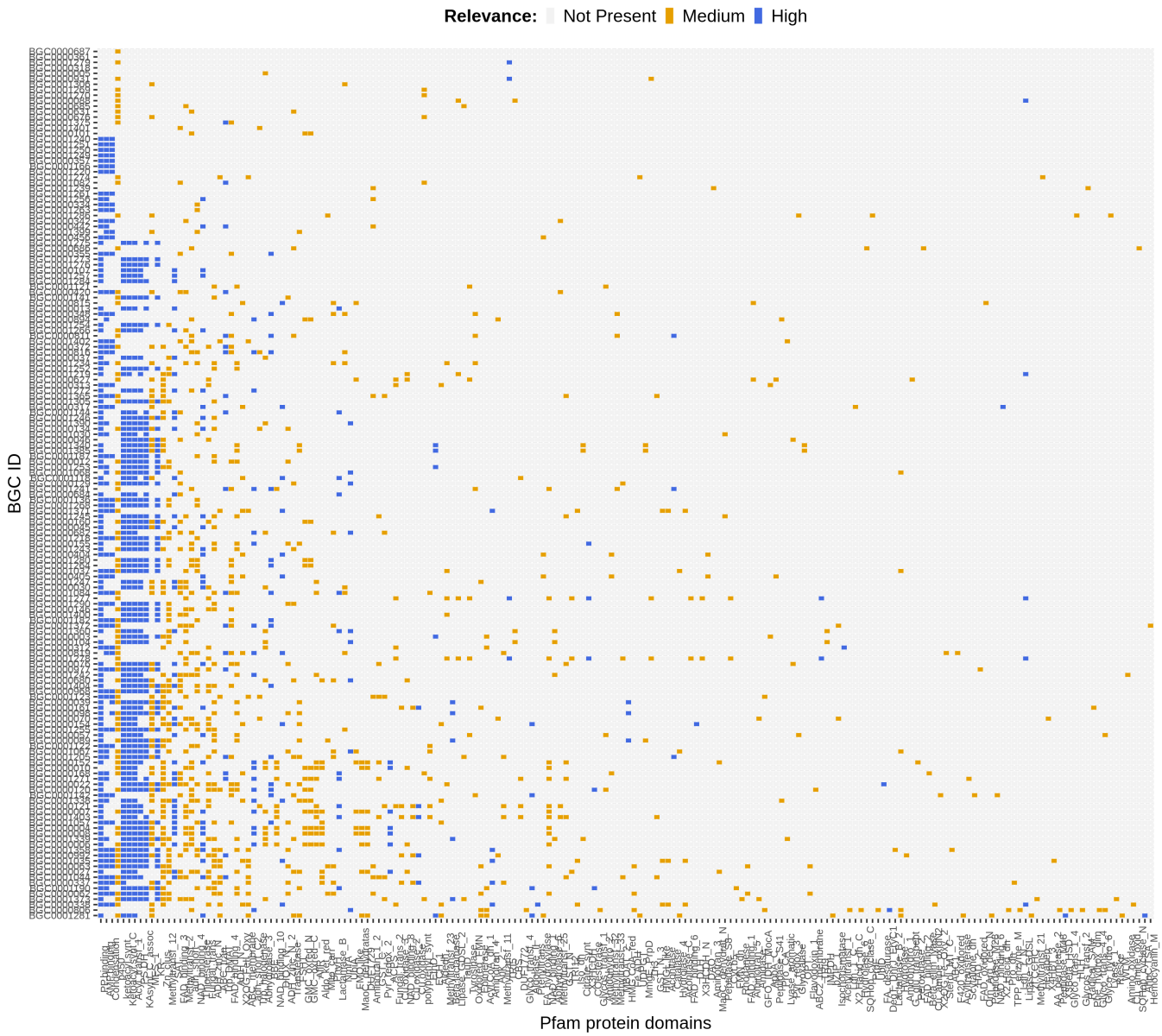
The performance of TOUCAN models was compared to two BGC discovery state-of-the-art approaches: DeepBGC, based on deep learning, and fungiSMASH, based on probabilistic methods. TOUCAN models showed better F-m when predicting BGCs in *A. niger* and *A. nidulans* compared to DeepBGC and fungiSMASH. TOUCAN also yielded more comprehensive coverage of gold standard BGC genes within predicted clusters, and was able to identify backbone enzyme genes more often in its TP predictions compared to the other methods. The presence of backbone enzymes can be a crucial aspect in determining the presence of a BGC in a given genomic region. The results obtained by TOUCAN, as well as the performance of DeepBGC models, demonstrate the potential of exploring supervised learning approaches for BGC discovery, and relevance of developing BGC prediction tools focused on fungal organisms. Fungi were shown to be an important source for bioactive compounds (Macheleidt et al., 2016; de Vries et al., 2017) used in the pharmaceutical industry, but fungal BGC data available in open access databases are scarce compared to bacteria. The availability of more annotated fungal BGCs is hence an important aspect to promote development and

improvement of existing and new fungal BGC discovery approaches. Previous BGC discovery tools require curated data to identify candidate BGC regions in an organism, which may not be available or is expensive to obtain. Unlike previous approaches, TOUCAN is capable of outputting BGC predictions from amino acid sequences without requiring further data curation as input. This aspect can facilitate TOUCAN usage and its application on newly sequenced genomes, promoting the discovery of novel candidate BGC regions and potentially novel drugs, such as antibiotics, immunosuppressants, and anti-cancer medications.

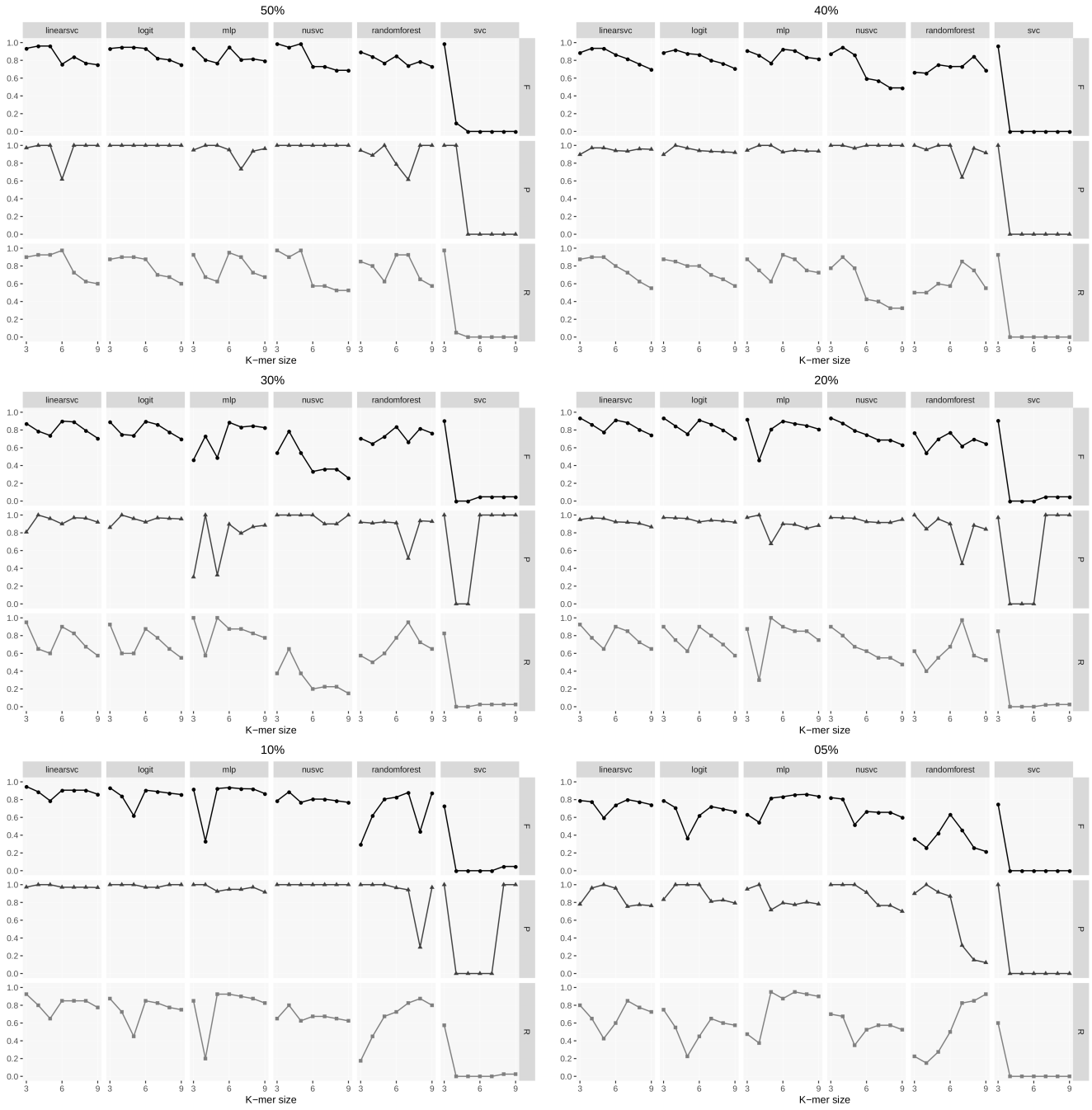
5.6 Data availability

TOUCAN source code as well as all datasets applied in our experiments are made publicly available at `http://github.com/bioinfoUQAM/TOUCAN`. TOUCAN source code is available under the MIT permissive software license. The datasets used in this work were obtained from open access databases, which are available under the Creative Commons Attribution 4.0 international license.

Supplementary Figure 1: Presence of Pfam protein domains annotated as *high* (usually present in BGCs) and *medium* (usually present, but not limited to BGCs) in our datatset positive instances. Each positive instance in our datasets is represented in a row. The columns represent the absence or presence of a *high* or *medium* Pfam protein domains, sorted by occurrence. The distribution of *high* and *medium* protein domains among positive instances shows that a structural pattern is shared by different BGC IDs.



Supplementary Figure 2: P, R, and F-m for classifiers on each validation set for $3 \leq K \leq 9$



Supplementary Table 1: Distribution of positive and negative instances across fungal BGC datasets, from completely balanced (50% positive, 50% negative) to most imbalanced (05% positive, 95% negative). Each dataset was split between train and validation subsets during the training phase.

Dataset	**Train**		**Validation**		**Total**	
distribution	**Pos**	**Neg**	**Pos**	**Neg**	**Pos**	**Neg**
50% - 50%	160	160	40	40	200	200
40% - 60%	160	240	40	60	200	300
30% - 70%	160	373	40	93	200	466
20% - 80%	160	640	40	160	200	800
10% - 90%	160	1,440	40	360	200	1,800
05% - 95%	160	3,040	40	760	200	3,800

Supplementary Table 2: Pfam domains annotated as *high* (usually present in BGCs) in our dataset positive instances.

Pfam ID	**Domain**	**Pfam ID**	**Domain**
PF00389	2-Hacid_dh	PF00378	ECH_1
PF01073	3Beta_HSD	PF00487	FA_desaturase
PF00725	3HCDH	PF00551	Formyl_trans_N
PF00583	Acetyltransf_1	PF00368	HMG-CoA_red
PF01648	ACPS	PF16197	KAsynt_C_assoc
PF00698	Acyl_transf_1	PF00109	ketoacyl-synt
PF13561	adh_short_C2	PF02801	Ketoacyl-synt_C
PF00578	AhpC-TSA	PF08659	KR
PF00596	Aldolase_II	PF00753	Lactamase_B
PF01063	Aminotran_4	PF00657	Lipase_GDSL
PF00501	AMP-binding	PF12013	OrsD
PF08031	BBE	PF00550	PP-binding
PF00144	Beta-lactamase	PF00432	Prenyltrans
PF00199	Catalase	PF14765	PS-DH
PF00135	COesterase	PF16073	SAT
PF00668	Condensation	PF00975	Thioesterase
PF00394	Cu-oxidase	PF06330	TRI5
PF01041	DegT_DnrJ_EryC1	PF08195	TRI9
PF14226	DIOX_N	PF11991	Trp_DMAT
PF01738	DLH	PF01040	UbiA

Supplementary Table 3: Pfam domains annotated as *medium* (usually present, but not limited to BGCs) in our datatset positive instances.

Pfam ID	Domain	Pfam ID	Domain	Pfam ID	Domain
PF02826	2-Hacid_dh_C	PF00970	FAD_binding_6	PF00891	Methyltransf_2
PF10014	2OG-Fe_Oxy_2	PF12831	FAD_oxidored	PF05050	Methyltransf_21
PF03171	2OG-FeII_Oxy	PF18325	Fas_alpha_ACP	PF13489	Methyltransf_23
PF02737	3HCDH_N	PF18314	FAS_I_H	PF13649	Methyltransf_25
PF13622	4HBT_3	PF17951	FAS_meander	PF13679	Methyltransf_32
PF13520	AA_permease_2	PF17828	FAS_N	PF10017	Methyltransf_33
PF00664	ABC_membrane	PF00465	Fe-ADH	PF07690	MFS_1
PF00005	ABC_tran	PF01613	Flavin_Reduct	PF00153	Mito_carr
PF03109	ABC1	PF00258	Flavodoxin_1	PF03972	MmgE_PrpD
PF01061	ABC2_membrane	PF01070	FMN_dh	PF00175	NAD_binding_1
PF07859	Abhydrolase_3	PF00743	FMO-like	PF13460	NAD_binding_10
PF08386	Abhydrolase_4	PF03959	FSH1	PF07993	NAD_binding_4
PF12697	Abhydrolase_6	PF04082	Fungal_trans	PF08030	NAD_binding_6
PF00330	Aconitase	PF11951	Fungal_trans_2	PF13450	NAD_binding_8
PF00694	Aconitase_C	PF01019	G_glu_transpept	PF05368	NmrA
PF00441	Acyl-CoA_dh_1	PF00117	GATase	PF03169	OPT
PF01553	Acyltransferase	PF01408	GFO_IDH_MocA	PF02784	Orn_Arg_deC_N
PF08240	ADH_N	PF01341	Glyco_hydro_6	PF00724	Oxidored_FMN
PF00106	adh_short	PF13692	Glyco_trans_1_4	PF00067	p450
PF00107	ADH_zinc_N	PF13632	Glyco_trans_2_3	PF04389	Peptidase_M28
PF13602	ADH_zinc_N_2	PF13579	Glyco_trans_4_4	PF01432	Peptidase_M3
PF08493	AflR	PF13439	Glyco_transf_4	PF01435	Peptidase_M48
PF00171	Aldedh	PF00534	Glycos_transf_1	PF02129	Peptidase_S15
PF00248	Aldo_ket_red	PF00535	Glycos_transf_2	PF03572	Peptidase_S41
PF01425	Amidase	PF00903	Glyoxalase	PF00082	Peptidase_S8
PF01979	Amidohydro_1	PF05199	GMC_oxred_C	PF08530	PepX_C
PF04909	Amidohydro_2	PF00732	GMC_oxred_N	PF01328	Peroxidase_2
PF01593	Amino_oxidase	PF00043	GST_C	PF07976	Phe_hydrox_dim
PF00155	Aminotran_1_2	PF02798	GST_N	PF05721	PhyH
PF00202	Aminotran_3	PF13417	GST_N_3	PF00348	polyprenyl_synt
PF00266	Aminotran_5	PF08759	GT-D	PF00484	Pro_CA
PF12796	Ank_2	PF13419	HAD_2	PF01619	Pro_dh
PF08546	ApbA_C	PF00372	Hemocyanin_M	PF04303	PrpF
PF00026	Asp	PF00132	Hexapep	PF07992	Pyr_redox_2
PF01212	Beta_elim_lyase	PF00010	HLH	PF13738	Pyr_redox_3
PF00170	bZIP_1	PF00682	HMGL-like	PF14027	Questin_oxidase
PF00571	CBS	PF18558	HTH_51	PF04055	Radical_SAM
PF00285	Citrate_synt	PF00702	Hydrolase	PF00581	Rhodanese
PF01179	Cu_amine_oxid	PF12146	Hydrolase_4	PF00355	Rieske
PF02727	Cu_amine_oxidN2	PF13344	Hydrolase_6	PF02982	Scytalone_dh
PF07731	Cu-oxidase_2	PF01231	IDO	PF13243	SQHop_cyclase_C
PF07732	Cu-oxidase_3	PF00478	IMPDH	PF13249	SQHop_cyclase_N
PF00173	Cyt-b5	PF00180	Iso_dh	PF08498	Sterol_MT_C
PF01266	DAO	PF00857	Isochorismatase	PF02668	TauD
PF01323	DSBA	PF12706	Lactamase_B_2	PF00205	TPP_enzyme_M
PF08354	DUF1729	PF02866	Ldh_1_C	PF02458	Transferase
PF08592	DUF1772	PF00056	Ldh_1_N	PF06609	TRI12
PF06441	EHN	PF02900	LigB	PF07428	Tri3
PF01370	Epimerase	PF13472	Lipase_GDSL_2	PF04820	Trp_halogenase
PF07110	EthD	PF00206	Lyase_1	PF00264	Tyrosinase
PF03807	F420_oxidored	PF00221	Lyase_aromatic	PF01977	UbiD
PF04116	FA_hydroxylase	PF13452	MaoC_dehydrat_N	PF00201	UDPGT
PF00667	FAD_binding_1	PF01575	MaoC_dehydratas	PF08325	WLM
PF00890	FAD_binding_2	PF13813	MBOAT_2	PF00096	zf-C2H2
PF01494	FAD_binding_3	PF08241	Methyltransf_11	PF00098	zf-CCHC
PF01565	FAD_binding_4	PF08242	Methyltransf_12	PF001728	Zn_clus

Supplementary Table 4: Unique features per training dataset distribution from completely balanced (50% positive, 50% negative) to most imbalanced (05% positive, 95% negative). Number of unique features (#) and feature percentage (%) is shown per each feature type for the total number of features in each dataset. K-mer features are shown for $K = 6$, the best performing K value in our study.

Dataset distribution	**K-mers (K=6)**		**Pfam domains**		**GO terms**		**Total**
	#	%	#	%	#	%	#
50% - 50%	45,874	(95.41)	1,866	(3.88)	341	(0.71)	48,081
40% - 60%	59,040	(96.59)	2,370	(3.87)	286	(0.46)	61,124
30% - 70%	80,604	(96.17)	2,885	(3.44)	323	(0.38)	83,812
20% - 80%	160,750	(97.38)	3,975	(2.41)	340	(0.20)	165,065
10% - 90%	559,708	(98.61)	7,524	(1.33)	354	(0.06)	567,586
05% - 95%	1,826,067	(98.97)	18,307	(0.99)	562	(0.03)	1,844,936

Supplementary Table 5: Validation performance on fixed train and validation sets per classifier. Models were built using all feature types combined.

Dataset	**Classifier**	**P**	**R**	**F-m**	**Average F-m**
50-50%	`lsvc`	1	0.925	0.961	**0.755**
50-50%	`logit`	1	0.925	0.961	**0.755**
40-60%	`mlp`	0.951	0.975	0.962	0.715
30-70%	`logit`	0.947	0.9	0.923	0.693
20-80%	`lsvc`	0.925	0.925	0.925	0.732
20-80%	`mlp`	0.925	0.925	0.925	0.732
10-90%	`mlp`	0.948	0.925	0.936	0.738
05-95%	`lsvc`	0.941	0.8	0.864	0.655

Supplementary Table 6: Validation performance on 5-fold CV per classifier on the completely balanced (50% positive, 50% negative) dataset. Models were built using all feature types combined.

Dataset	**Classifier**	**P**	**R**	**F-m**
50-50%	`lsvc`	0.934	0.925	0.929
50-50%	`logit`	0.922	0.935	0.928
50-50%	`mlp`	0.948	0.910	0.928
50-50%	`nusvc`	0.708	0.750	0.723
50-50%	`randomf`	0.944	0.900	0.919
50-50%	`svc`	0.911	0.900	0.904

Supplementary Table 7: DeepBGC original and fungal optimized hyperparameters applied during evaluation

Parameter	**Original**	**Fungal**
batch_size	64	16
hidden_size	128	128
timesteps	256	256
num_epochs	328	50
dropout	0.2	0.2
optimizer	adam	adam
learning_rate	1e-4	1e-4
loss	weighted binary cross-entropy	weighted binary cross-entropy

Supplementary Table 8: TOUCAN best performing hyperparameters to maximize F-m for each classifier.

`lsvc`	C = 0.01, loss = squared_hinge, penalty = l2
`logit`	penalty = l1, C=10, solver = saga
`mlp`	activation = relu, batch_size = 256, hidden_layer_sizes= 256, learning_rate = 'adaptive', solver = 'adam'
`nusvc`	coef0 = 0.01, gamma = 0.01, kernel = sigmoid
`randomf`	bootstrap = False, criterion = entropy, max_features= log2, n_estimators = 1000
`svc`	C = 100, gamma = 0.001, kernel = rbf

CHAPTER VI

IMPROVING BGC PREDICTION THROUGH REINFORCEMENT LEARNING AND FUNCTIONAL ANNOTATIONS

The results obtained with TOUCAN, a supervised learning framework to identify fungal BGCs, yield high F-measure when predicting cluster regions, outperforming previous tools. However due to the post-processing methods, TOUCAN predicted BGC regions were prone to overestimation of cluster boundaries, a common issue among previous BGC discovery tools. This Chapter describes the methodology adopted to build a reinforcement learning method to improve BGC predictions outputted by state-of-the-art tools. The approach aims to optimize composition of candidate BGCs, optionally integrating functional annotations of BGC components to improve performance. The study presented in this Chapter was submitted to the Bioinformatics journal, under the title "Improving candidate Biosynthetic Gene Clusters in fungi through reinforcement learning". We note that a short version of this article was accepted at the 25th international conference on Research in Computational Molecular Biology (RECOMB), as a poster under the title "A reinforcement learning approach to improve fungal Biosynthetic Gene Cluster prediction". Article writing, approach implementation, experimental design and execution were performed by Hayda Almeida, under the supervision of professors Adrian Tsang and Abdoulaye Baniré Diallo. A printed version of this article is presented in the Appendix C.

6.1 Abstract

Motivation: Precise identification of Biosynthetic Gene Clusters (BGCs) is a challenging task. Performance of BGC discovery tools is limited by their capacity to accurately predict components belonging to candidate BGCs, often overestimating cluster boundaries. To support optimizing the composition and boundaries of candidate BGCs, we propose reinforcement learning approach relying on protein domains and functional annotations from expert curated BGCs.

Results: The proposed reinforcement learning method aims to improve candidate BGCs obtained with state-of-the-art tools. It was evaluated on candidate BGCs obtained for two fungal genomes, *Aspergillus niger* and *Aspergillus nidulans*. The results highlight an improvement of the gene precision by above 15% for TOUCAN, fungiSMASH and DeepBGC; and cluster precision by above 25% for fungiSMASH and DeepBCG, allowing these tools to obtain almost perfect precision in cluster prediction. This can pave the way of optimizing current prediction of candidate BGCs in fungi, while minimizing the curation effort required by domain experts.

Availability and Implementation:

`https://github.com/bioinfoUQAM/RL-bgc-components`

Contact: diallo.abdoulaye@uqam.ca

Supplementary information: Supplementary data is available at Bioinformatics online.

6.2 Introduction

Filamentous fungi produce a large array of Secondary Metabolites (SM) which play an important role in the survival and development of producing organisms (Keller, 2015). Identifying novel fungal SMs is a field of high interest, given the relevance of these compounds particularly in the pharmaceutical industry for production of

various medications (Chavali & Rhee, 2017; Kjærbølling et al., 2019). Biosynthetic pathways that produce SM compounds are encoded by clusters of genes often appearing contiguously in an organism genome, known as Biosynthetic Gene Clusters (BGCs) (Keller, 2019; Kautsar et al., 2020). The genomic diversity of fungal genomes makes accurate identification of BGCs in fungi a highly challenging task for dedicated state-of-the-art tools, and even for manual curation or experimental characterization performed by experts (Kjærbølling et al., 2019). BGCs generally contain minimal components: backbone enzymes, defining the core chemical compound to be produced; and tailoring enzymes, capable of generating variants by modifying the cluster core compound (Keller, 2019). They may also present other components, such as cluster-specific transcription factors, transporters, and hypothetical proteins (Keller, 2015). Fungal BGCs are known to vary considerably in composition (similar clusters with different components), and location (cluster regions overlapping or spanning multiple chromosomes) even among closely related species (Keller, 2019; Kjærbølling et al., 2020; Evdokias et al., 2021).

Various approaches to obtain candidate BGCs (potential sequence regions encoding biosynthesis of SMs) were previously presented (Chavali & Rhee, 2017), such as fungiSMASH (Blin et al., 2021), DeepBGC (Hannigan et al., 2019), and TOUCAN (Almeida et al., 2020). However these approaches show limitations when it comes to the identification of components and boundaries of candidate BGCs, often overpredicting candidate regions. fungiSMASH offers the option to integrate CASSIS (Wolf et al., 2016) to improve cluster boundary prediction. Apart from being a potentially time-consuming option, CASSIS requires curated input, such as gene start and end positions and a reference anchor (backbone) gene, which may not be readily available and therefore limit its stand-alone application to other state-of-the-art BGC discovery approaches.

Obtaining accurate candidate BGCs is a critical step towards chemical synthesis

of SM compounds, which can be a complex and costly process as many of these metabolic pathways are silent or poorly expressed (Montiel et al., 2015; Zhang et al., 2019). In this work, we propose a reinforcement learning approach based on protein family domains from Pfam (El-Gebali et al., 2019) and functional annotations to support optimizing the boundaries and composition of candidate BGCs obtained with state-of-the-art tools, therefore potentially facilitating validation and experimental characterization of SM compounds. Protein domains were previously used in approaches to identify BGCs (Khaldi et al., 2010; Hannigan et al., 2019), and are used here to represent common or shared functional profiles among BGCs, such as presence of relevant components. Reinforcement learning methods are capable of adapting dynamically given feedback received (Neftci & Averbeck, 2019), and therefore might be suitable to handle the overestimation of candidate BGC boundaries, as well as the intrinsic diversity of fungal BGC components, potentially favoring the discovery of novel compounds.

In reinforcement learning, a learning agent interacts directly with an environment through actions in a goal-oriented manner, attempting to maximize its task reward and find an optimal solution (Sutton & Barto, 2018). The agent actions are assigned rewards or penalties, computed based on a given function and according to environment states reached (Sutton & Barto, 2018). When optimizing candidate BGCs, rewards could be assigned for when the agent identifies correct components and properly defines cluster boundaries, while penalties could be given when the agent disregards relevant components from a candidate BGC. While navigating through the environment, the learning agent tries to balance exploitation (acquired knowledge of best actions taken) and exploration (choose actions not tried previously) (Sutton & Barto, 2018). Reinforcement learning approaches had limited applications in biological contexts so far (Mahmud et al., 2018), however results show they generated robust policies and outperformed previous methods in tasks

performing multiple sequence alignment (Mircea et al., 2018), controlling gene regulatory networks (Imani & Braga-Neto, 2018), optimizing DNA and protein sequences (Angermueller et al., 2020), and performing *de novo* drug design (Gottipati et al., 2020). Our reinforcement learning approach relies on protein domains and functional annotations of BGC components to optimize candidate BGCs obtained with state-of-the-art tools, which often overestimate cluster boundaries.

6.3 Methods

The reinforcement learning approach presented here relies on Q-learning (Watkins & Dayan, 1992), a off-policy temporal difference algorithm, which is capable of learning directly from interacting with the environment, without relying on an environment model nor on a long-term value. Rather, a Q-learner uses the next step reward and estimates its gain for the following update and learns from each state transition (Sutton & Barto, 2018). To model a reinforcement learner agent, Pfam protein domains were extracted from curated BGC instances and synthetic non-BGC instances, as described in Section 6.3.1. Specific rewards were computed for protein domains according to their occurrence in cluster regions of BGC and synthetic non-BGCs, as described in Section 6.3.2. Test candidate BGCs were then submitted to the reinforcement learning agent to decide on potential BGC components to keep or skip. As a final step, the agent decisions could then be further enhanced by strategies developed based on curated functional annotations of BGC components, as described in Section 6.3.3. Overall performance is evaluated based on cluster and gene metrics, as described in Section 6.3.4.

6.3.1 Datasets

Publicly available fungal BGC benchmark datasets (Almeida et al., 2019) were applied to develop the reinforcement learning approach presented here. Both

training and test data are represented through the occurrence of Pfam protein domain features in curated BGC regions, non-BGC regions, and test candidate BGC regions. Previous work has shown the relevance of Pfam domains as features for BGC analysis (Inglis et al., 2013; Kjærbølling et al., 2020) and discovery (Hannigan et al., 2019; Almeida et al., 2020). Pfam domains can indicate the presence of key BGC components as discussed in Section 6.2, such as polyketide synthase or non-ribosomal peptide synthetase genes encoding backbone enzymes, genes encoding tailoring enzymes, transcription factors or transporters. Genes (or genomic regions, if gene annotations are not available) composing BGCs may contain none to multiple relevant Pfam domains.

Training Publicly available training datasets are presented in Almeida et al. (2019). These training datasets are composed of curated fungal BGC instances obtained from MIBiG (Minimum Information about a Biosynthetic Gene cluster) (Kautsar et al., 2020) repository, and synthetic non-BGC instances created from OrthoDB (Kriventseva et al., 2018) fungal orthologous genes. Training datasets of various distributions were generated through sampling of orthologous synthetic non-BGC instances, combined with curated fungal BGC instances (see Almeida et al. (2019) for details). Previous work has shown the relevance of orthologous genes in BGC discovery as they indicate conserved genomic regions (Takeda et al., 2014; Almeida et al., 2020), while BGC regions tend to present high genomic diversity even among closely related species (Kjærbølling et al., 2020). Publicly available training datasets of various distributions were previously evaluated in Almeida et al. (2020), identifying the most balanced one (50% BGC and 50% non-BGC instances) as the dataset yielding the best performance. For comparison purposes, this is therefore the training dataset applied in our approach.

Testing The decisions taken by the reinforcement learning agent are evaluated on candidate BGCs obtained for the *Aspergillus niger* NRRL3 genomic sequence (publicly available at `https://gb.fungalgenomics.ca/portal`) by three tools: TOUCAN (Almeida et al., 2020), fungiSMASH (Blin et al., 2021), and DeepBGC (Hannigan et al., 2019). *Aspergillus niger* is an organism of interest given its ubiquitous presence, and its importance for industrial processes and biotechnology, which makes it a relevant species in the study of BGC discovery (de Vries et al., 2017; Aguilar-Pontes et al., 2018; Evdokias et al., 2021). To obtain test candidate BGCs from *A. niger* amino acid sequence, we extracted sequentially sliding windows of fixed 10,000 amino acid length with a 30% window overlap (see Almeida et al. (2020) for details). *Aspergillus niger* candidate BGCs were then obtained from each BGC discovery tool, based on the same sequentially sliding windows to allow candidate predictions to be compared across the three tools. Before being processed by the proposed reinforcement learning agent, candidate BGCs obtained by all three tools were pre-processed using a majority vote strategy.

Candidate BGC pre-processing − Majority vote: Candidate BGCs contain a set of genomic region identifiers (such as gene names), as well as their corresponding Pfam protein domains. Examples of candidate BGCs are shown in Figure 6.1. For our experiments, candidate BGCs were obtained based on a test set of *A. niger* genomic regions of 10,000 amino acid sliding windows with a 30% overlap.

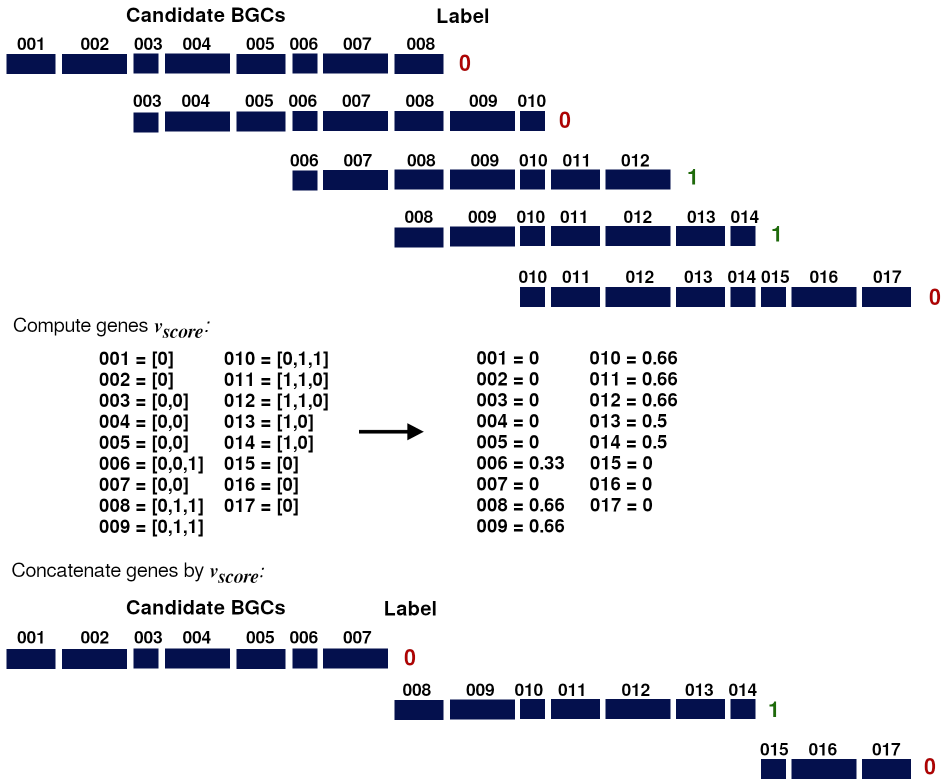


Figure 6.1: Computation of majority vote pre-processing for candidate BGCs: regions are merged according to the average score of predicted labels

On one hand, overlapping regions allow for covering potential BGC fragmentation due to fixed length sliding windows. On the other hand it will also generate repeated regions in candidate BGCs. The majority vote strategy, shown in Figure 6.1, therefore handles duplicated regions based on a local consensus. It works as follows: each gene g in a candidate BGC is represented by a label vector $L = l_0, l_1, ..., l_m$ where m is the number of candidate BGCs in which g appears and l_i the candidate BGC label (0 for predicted as non-BGC and 1 for predicted as BGC). The majority vote score v_{score} for a gene g is therefore the average value of its predicted labels $\overline{L}$. Sequential genes presenting a $v_{score} \geq 0.5$ are therefore concatenated as positive candidate BGCs, while the other genes with a $v_{score} < 0.5$ are concatenated as negative candidate BGCs, up to a limit of 10,000 amino acids

per cluster. In our experiments, *A. niger* gene models were used as reference points, however in the lack of gene models, regions of fixed smaller size than the sliding window length could be considered instead.

6.3.2 Reinforcement learning method

The proposed reinforcement learning approach is based on the temporal-difference and off-policy algorithm Q-learning (Watkins & Dayan, 1992; Sutton & Barto, 2018). In Q-learning, the action-value function Q converges towards an optimal policy, and allows the reinforcement learning agent to decide on the next step. The Q function provides the expected value of an action a, given a state s, and it is dynamically updated during the agent experience of interacting with the environment. Given a set of actions A, a set of states S and respective rewards R at a timestep t, the Q function is computed as:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha\,[\,R_{t+1} \;+\; \gamma \max_a Q(S_{t+1,a}) - Q(S_t, A_t)]$$

where α is the learning rate, and γ the discount-rate factor. Additionally, a probability ϵ defines the algorithm exploration versus exploitation rate (Sutton & Barto, 2018). In the context of optimizing BGC components, the reinforcement learning agent chooses the most suitable action within the set of actions $A = keep, skip$ for a candidate BGC, which is a set of states represented by Pfam domains within each gene. At the training phase state rewards were computed by extracting Pfam protein domains from the selected training dataset, as described in Section 6.3.1. Each protein domain d is represented by an occurrence vector $C = c_0, c_1, ...c_n$, where n is the number training dataset instances, and c_i the domain occurrence per training instance ($c_i > 0$ if a curated BGC instance, and $c_i < 0$ otherwise). To determine the rewards per action R_{keep} and R_{skip} of a domain d, we first compute a score s as follows:

$$s_{keep} = \sum_{x \in C} \frac{x}{|C|} \qquad s_{skip} = |1 - s_{keep}|$$

After computing both s_{keep} and s_{skip}, a $keepSkip$ threshold is applied to finally determine the rewards R_{keep} and R_{skip} for domain d, as in:

$$R_{keep}, R_{skip} = \begin{cases} s_{keep}, -s_{keep} & \text{if } s_{keep} > (s_{skip} * keepSkip) \\ -s_{skip}, s_{skip} & \text{otherwise.} \end{cases}$$

The agent is assigned a penalty for each step it receives a negative reward $R < 0$, with a total penalty computed per episode. An episode is completed when the agent has gone through the entire training dataset.

In the testing phase, the reinforcement learning agent is evaluated by the *keep* or *skip* actions it decides on for genes in candidate BGCs. Pfam domains are therefore extracted per gene (or per fixed size region, in case gene models are not available) in candidate BGCs. The optimal action for a gene g containing a set of domains $D = d_0, d_1, ..., d_n$, where n is the number of domains found in g is computed as follows:

$$g_a = argmax(\sum_{i=0}^{n} d_i(R_{keep}), \sum_{i=0}^{n} d_i(R_{skip}))$$

Genes for which $R_{skip} > R_{keep}$ are assigned the action $g_a = skip$, otherwise they are assigned a $g_a = keep$. Only genes assigned a $g_a = keep$ action will be maintained in a given candidate BGC.

6.3.3 Integrating functional annotations

Biosynthetic gene clusters are generally formed by components that play different roles in the cluster, such as backbone and tailoring enzymes, transcription factors, transporters, and hypothetical proteins, as discussed in Section 6.2. Backbone and tailoring enzymes for instance are considered essential BGC building blocks for the biosynthesis of SM compounds (Keller, 2019). A total of 85 *A. niger* BGCs (In-

glis et al., 2013) were used as our gold standard. To define these BGCs, Inglis et al. (2013) described obtaining *in silico* BGCs from state-of-the-art tools, and refining their boundaries based on published experimental data, synteny between BGC genes across multiple species, assignment of experimentally based GO terms, intergenic distance between boundary and adjacent genes. These 85 gold standard *A. niger* BGCs were then manually curated with their functional annotation within clusters. Pfam protein domains were then extracted from functionally annotated BGC gold-standard genes, and associated with a BGC component role. A list of all Pfam domains associated with each annotated BGC component is shown in Supplementary Table 1.

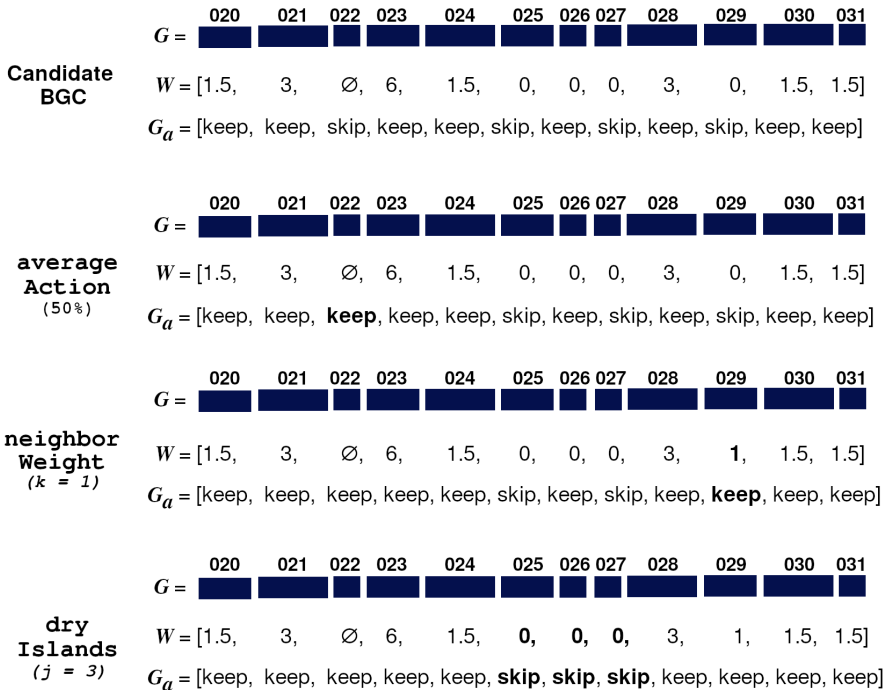


Figure 6.2: Example of functional annotation strategies applied to a candidate BGC

To integrate the functional annotation of BGC components, three strategies were developed based on Pfam domains associated to component roles. The three

strategies are applied to enhance the reinforcement learning agent decisions. The `averageAction` strategy handle genes lacking Pfam domains; the `neighborWeight` strategy handles presence of annotations in neighboring genes; and the `dryIslands` strategy handles absence of annotations in contiguous neighboring genes.

Various gold-standard BGC genes, mostly annotated as hypothetical proteins, simply do not contain any Pfam domain annotations and therefore may be directly assigned an action $g_a = skip$. BGC components considered hypothetical proteins may play a relevant role in the cluster (Keller, 2015). However they become challenging components to identify due to their lack of features, which makes them harder to distinguish from the noise within non-relevant components. With the `averageAction` strategy, if the reinforcement learning agent assigns an action $g_a = keep$ for a minimum gene threshold in a candidate BGC G, then genes in G that do not contain protein domains ($D = \varnothing$) will also be assigned an action $g_a = keep$. Optimization of the minimum threshold ([25%, 50%, 75%]) has yielded 50% as the most suitable value.

To implement the `neighborWeight` and `dryIslands` strategies, a candidate BGC G is assigned a weight vector W, where for each gene g in G a weight w is computed as follows:

$$w = \sum_{i=0}^{n} h_i \qquad\qquad h_i = \begin{cases} \beta & \text{if backbone,} \\ \lambda & \text{if other annotation,} \\ \sigma & \text{otherwise.} \end{cases}$$

where n is the number of domains found in g, and h the score associated with the BGC component functional annotation. For the sake of the experiments described in Section 6.4, we have set the following values: $\beta = 2$ if backbone, $\lambda = 1.5$ if other annotation, and $\sigma = 0$ otherwise. For the `neighborWeight` strategy, if a k number of surrounding neighbors of a gene g present a $\sum_{i=0}^{k} w_i > 1$, then the

gene weight $g_w = 1$ and the gene action $g_a = keep$. Optimization of the number of neighbor genes $k = [1, 2, 3]$ has yielded the most suitable $k = 1$. For the `dryIslands` strategy, if $\sum_{i=0}^{j} g_w = 0$ for j sequential genes in G, then the gene action $g_a = skip$. Optimization of the dry island size $j = [3, 4, 5]$ has yielded the most suitable $j = 3$. Figure 6.2 shows an example of how the reinforcement learning agent decisions are adjusted by the `neighborWeight` and `dryIslands` strategies. Functional annotations of BGC components provide expert domain knowledge and could potentially improve the actions chosen by the reinforcement learning agent, therefore improving precision of candidate BGC components.

6.3.4 Evaluation metrics

The performance of the reinforcement learning approach proposed here is evaluated in terms of *gene metrics* and *cluster metrics*, for which precision (P), recall (R), F-measure (F-m) are computed. *Cluster metrics* show the performance on identifying cluster regions, and considers as true positives (TPs) candidate BGCs G that have at least one gene g that belongs to the set of gold-standard BGC genes. *Gene metrics* shows the performance on matching genes in candidate BGCs with the complete set of gold-standard BGC genes, and considers as true positives (TPs) the candidate BGC genes that are identical or similar gene matches to gold-standard BGC genes. The similarity between candidate and gold-standard BGC genes is obtained through local BLAST alignment, with minimum thresholds of percent identity $pident \geq 20$ and query coverage $qcov \geq 10$. We also compute the average F-m between *cluster* and *gene metrics* F-m.

6.4 Results

The reinforcement learning approach proposed here is evaluated on candidate BGCs obtained with three BGC discovery tools: TOUCAN (Almeida et al.,

2020), fungiSMASH (Blin et al., 2021) independently and also combined with CASSIS (Wolf et al., 2016), and DeepBGC (Hannigan et al., 2019) for the *A. niger* genome. A total of 85 *A. niger* BGCs (Inglis et al., 2013) were manually curated and are considered as gold standard to evaluate the performance of our reinforcement learning approach on selecting BGC components from candidate BGCs. In Section 6.4.1 we present an overview of the distribution of genes presenting protein domains associated to functional annotations in the training and test data. Section 6.4.2 presents the results obtained by the reinforcement learning approach on candidate BGCs from the three tools, and Section 6.4.3 shows an analysis of reproducibility of the reinforcement learning approach in a second fungal genome, *Aspergillus nidulans*.

6.4.1 Distribution of domains linked to BGC components

We performed an analysis of the presence of protein domains associated with BGC component roles in genes belonging to the training and test datasets. The distribution of genes that present protein domains associated with BGC component types is shown in Table 6.1. A protein domain may be associated with multiple component roles if it was found to be present in genes annotated with different components.

Table 6.1: Domains linked to *A. niger* BGC components in dataset genes

Component type	**Training**		**Test**	
	BGCs	**non-BGCs**	**gold BGCs**	**non-gold BGCs**
Backbones	17.0%	2.0%	15.9%	2.2%
Tailoring enzymes	30.5%	7.8%	9.9%	11.9%
Transcription factors	4.8%	2.1%	5.9%	4.3%
Transporters	5.6%	2.8%	7.4%	4.6%
Non-component domains	44.7%	46.93%	49.3%	58.9%
No domains	14.6%	41.15%	15.5%	23.2%
Total # genes	2833	1781	624	11239

It is noticeable from Table 6.1 that protein domains appearing in BGC components are mostly found among genes in BGCs and gold BGCs instances. Genes that do not contain any protein domains are mostly found among non-BGCs and non-gold

BGCs instances. The percentage of genes without any encoded protein domains is higher than that of genes with encoded domains associated to transcription factors and transporters among BGCs and gold BGC genes.

The distribution of genes encoding protein domains associated with backbones in the training data is similar to the that of the test data. Genes without any encoded protein domains also yield a similar distribution among BGCs (14.6%) and gold BGCs (15.5%) genes. Among non-gold-standard BGC genes, more than half encode protein domains that are not associated to any component role. Overall the percentages in Table 6.1 demonstrate how the presence of protein domains associated to BGC components is ubiquitous both in BGCs and non-BGC regions, which makes correctly identifying BGC components a challenging task.

6.4.2 Reinforcement learning improves candidate BGCs

We present here the results obtained by the proposed reinforcement learning approach on candidate BGCs obtained with three BGC discovery tools: TOUCAN, fungiSMASH (fungiSMASH/C combined with CASSIS), and DeepBGC. Previously to processing candidate BGCs, we optimized the following reinforcement learning agent parameters: learning rate α, discount-rate factor γ, exploration-exploitation probability ϵ, and the *keepSkip* threshold, as described in Section 6.3.2, over a set of 500 episodes on the training data evaluating both fixed and incremental parameter values. The parameters $\alpha = 0.01, \gamma = 0.01, \epsilon = 0.01, keepSkip = 0.5$ yielded the smallest average penalty over 500 episodes. Supplementary Tables 2 and 3 show a summary of the parameter optimization. In this Section, we refer here to `TOUCAN`, `fungiSMASH`, `fungiSMASH/C` and `DeepBGC` as the candidate BGCs directly outputted by each tool; `TOUCAN-Q`, `fungiSMASH-Q`, `fungiSMASH/C-Q` and `DeepBGC-Q` as the candidate BGCs processed by the proposed reinforcement learning approach; and `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/C-Q-all` and

DeepBGC-Q-all as the candidate BGCs processed by the reinforcement learning approach combined with functional annotation strategies.

Table 6.2: Performance on *A. niger* candidate BGCs from TOUCAN, fungiSMASH and DeepBGC

	gene metrics			cluster metrics			average	% gold-std. genes	
model	P	R	F-m	P	R	F-m	F-m	negative	skipped
TOUCAN	0.269	0.906	0.414	0.963	0.929	0.946	0.68	12.6%	-
TOUCAN-Q	0.402	0.68	0.506	0.963	0.929	0.946	0.726	12.6%	26.4%
TOUCAN-Q-all	0.409	0.74	0.527	0.963	0.929	0.946	0.737	12.6%	16.2%
fungiSMASH	0.341	0.665	0.451	0.649	0.741	0.692	0.571	33.2%	-
fungiSMASH-Q	0.521	0.516	0.519	1	0.741	0.851	0.685	33.2%	22.3%
fungiSMASH-Q-all	0.495	0.575	0.532	1	0.741	0.851	0.691	33.2%	13.8%
fungiSMASH/C	0.371	0.713	0.488	1	0.729	0.844	0.666	34.13%	-
fungiSMASH/C-Q	0.523	0.508	0.515	1	0.729	0.844	0.680	34.13%	22.11%
fungiSMASH/C-Q-all	0.523	0.508	0.515	1	0.729	0.844	0.680	34.13%	22.11%
DeepBGC	0.351	0.481	0.406	0.732	0.612	0.667	0.536	52.4%	-
DeepBGC-Q	0.574	0.42	0.485	1	0.612	0.759	0.622	52.4%	12.2%
DeepBGC-Q-all	0.538	0.46	0.496	1	0.612	0.759	0.627	52.4%	7.1%

Table 6.2 shows the results obtained by the reinforcement learning agent on candidate BGCs for all three tools. As discussed in Section 6.3.4, cluster metrics show the approach performance on identifying cluster regions, while gene metrics show the performance on matching candidate and gold-standard genes within a BGC. The average F-m shows the overall performance, considering both cluster F-m and gene F-m. The proposed reinforcement learning approach improved gene metrics, more noticeably gene precision in candidate BGCs outputted by all three tools: an increase of 14%, 15.4%, 15.2%, and 18.7% achieved by TOUCAN-Q-all, fungiSMASH-Q-all, fungiSMASH/C-Q-all and DeepBGC-Q-all respectively. For TOUCAN-Q-all and fungiSMASH/C-Q-all, gene metrics were improved without harming cluster metrics, while for fungiSMASH-Q-all and DeepBGC-Q-all cluster metrics were also improved considerably, with an F-m increase of 15.9% and 9.2% for fungiSMASH-Q-all and DeepBGC-Q-all respectively. This indicates that

the reinforcement learning agent was capable of improving the precision of candidate BGC components without discarding correctly predicted candidate BGCs, and improving coverage of true positive BGC regions and properly targeting false positive ones predicted by both `fungiSMASH` and `DeepBGC`. The average F-m of all three tools also improved when applying the reinforcement learning agent combined with the functional annotation strategies. An increase in average F-m of 5.7%, 12%, 1.4%, and 9.1% was shown for `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/C-Q-all` and `DeepBGC-Q-all` respectively. Apart from improving gene precision, all candidate BGCs processed by the reinforcement learning agent combined with functional annotation strategies (`Q-all`) yielded a smaller percentage of gold-standard genes skipped, except for `fungiSMASH/C-Q-all`, which yield the same performance for `Q` and `Q-all` models. This suggests that BGC functional annotations can be relevant features to support improving precision of predicted BGCs, and better determine their structure.

Candidate BGCs shown in Figure 6.3 demonstrate the changes in cluster composition before and after applying the presented reinforcement learning method. A comparison between gold-standard and candidate BGCs in Figure 6.3-A shows how the reinforcement learning agent improved candidate BGCs from all three tools by correctly skipping non-BGC genes (in blue). Certain cases however are more complex for the agent, given the ambiguity of protein domains in candidate BGC genes. As the examples in Figure 6.3-B show, more non-BGC genes were kept by the agent, which can lead to processed candidate BGCs to be somehow overpredicted. This behavior could be caused by the fact that domains found in non-BGC genes in Figure 6.3-B also appear in true positive BGC genes, as opposed to Figure 6.3-A for which most domains in non-BGC genes were not present in any true positive BGC genes. Among protein domains of non-BGC genes (blue) in Figure 6.3-B, more than 50% are associated to BGC component

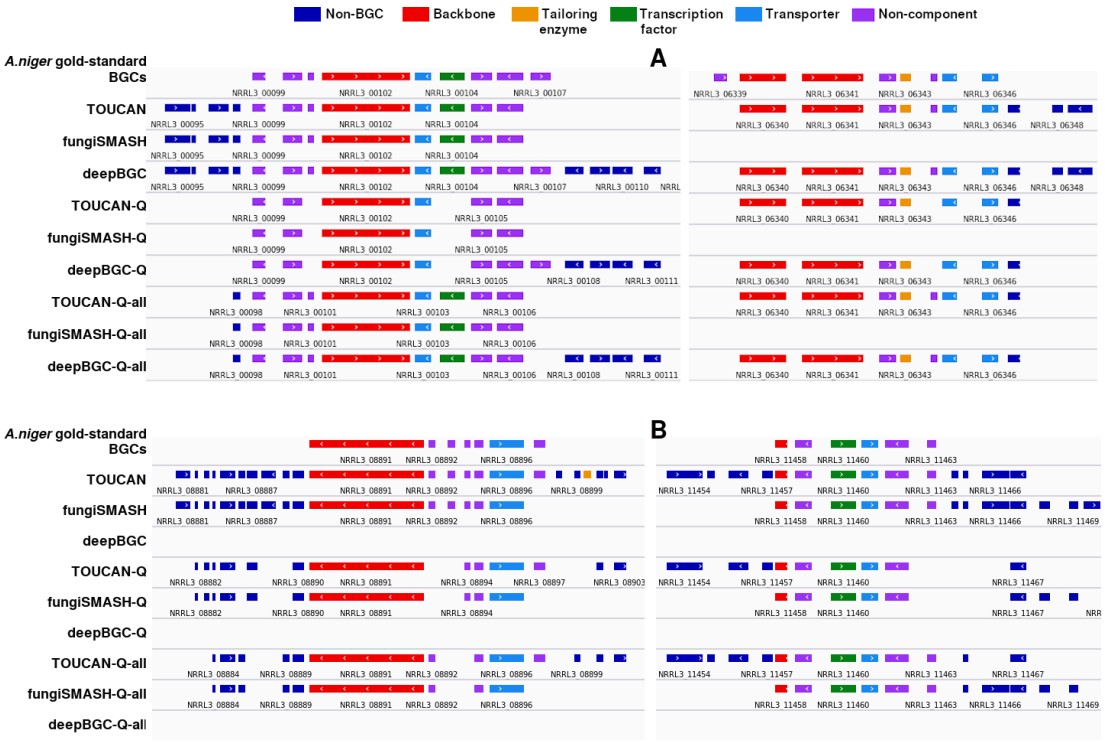


Figure 6.3: Comparison between gold-standard and candidate BGC composition for four *A. niger* clusters. Non-BGC genes are shown in dark blue. (**A**) Candidate BGCs for which the reinforcement learning agent correctly skipped most non-BGC genes compared to their polyketide (left) and fatty acid (right) gold standard BGCs. (**B**) Candidate BGCs for which the agent kept most non-BGC genes compared to their two non-ribosomal peptide gold standard BGCs, possibly due to their ambiguous protein domains, which more than half are associated to BGC component roles but do not belong to neighboring clusters.

roles, and found immediately after true positive BGC genes. Non-BGC genes shown in Figure 6.3-A presented only 20% of domains linked to BGC component roles. This demonstrates how ambiguous domains in candidate BGCs or their neighboring genes, along with the genomic diversity of these clusters, may increase the complexity of accurately identifying BGC components and boundaries.

Properly identifying BGC components is a challenging task not only for computational approaches that attempt to do so, but even for synthetic approaches that try to express genes composing candidate BGCs (Keller, 2019). Supplementary Table 4 shows an analysis of *A. niger* BGC component types found in gold-standard BGC genes and components found in candidate BGCs, before and after applying the reinforcement learning approach proposed here. As discussed in Section 6.3.3, gold BGC genes may contain none to multiple domains, therefore they may present none to multiple functional annotations. Candidate BGCs outputted by `fungiSMASH` and `DeepBGC` presented a smaller number of true positives, and consequently a smaller number of components was found compared to `TOUCAN` candidates, as shown in Supplementary Table 4.

The reinforcement learning agent aims to improve precision of candidate BGC components by removing potentially non-relevant regions. At the same time, the agent has to handle ambiguous genes that map to protein domains, normally found in both BGC and non-BGC instances. The number of backbone genes properly identified by `TOUCAN` (92.9%), `fungiSMASH` (70.7%), `fungiSMASH/C` (69.7%) and `DeepBGC` (64.6%) remains the same even after processing by the reinforcement learning agent for all three tools. This could indicate that the reinforcement learning agent was capable of learning correctly the relevance of regions encoding such enzymes. Backbone enzymes are vital components of BGCs (Kjærbølling et al., 2020), and their accurate identification could demonstrate the robustness of a BGC discovery method. Transcription factors and transporters in `DeepBGC`

candidate BGCs were maintained by the reinforcement learning agent, however the overall percentage of these components remains lower than the percentage identified by `TOUCAN` and `fungiSMASH`.

Some BGC genes are not associated to any component role, and often do not even contain any Pfam protein domains, as discussed in Section 6.3.3. Usually considered as hypothetical proteins, these genes pose a challenge on correctly identifying BGC components, and could be overlooked by BGC discovery approaches since their computational representation will likely be more analogous to non-BGC regions. These hypothetical proteins can seem to diverge from other BGC components but they may play important self-protection roles for the organism producing a SM compound (Keller, 2019). As shown in Supplementary Table 4, genes without any domains were the most missed by the reinforcement learning approach (`Q`) among candidate BGCs from all three tools. The `averageAction` strategy aims to address this issue by keeping candidate BGC genes without domains when at least a minimum 50% threshold of genes within a candidate BGC are assigned the action *keep*. A more lenient threshold was experimented with for `averageAction` strategy, however it can lead to the agent identifying a higher number false positives − genes without protein domains and often associated with non-relevant BGC regions − resulting in a decrease in precision.

6.4.3 Reproducibility in *Aspergillus nidulans* candidate BGCs

Similarly to *A. niger*, *A. nidulans* is a source of highly useful SMs compounds which are also largely utilized in the pharmaceutical industry (Inglis et al., 2013; Drott et al., 2020). To further evaluate the reproducibility of the proposed reinforcement learning approach, we processed the *A. nidulans* genome considering as gold standard a total of 72 gold standard BGCs presented in Drott et al. (2020). Assignment of functional annotations to BGC components is a costly and

time-consuming process. Since manually curated component annotations were not available for *A. nidulans* gold-standard BGCs, we generated pseudo-annotations by assigning potential component types to gold-standard BGC genes based on similar keywords found in their protein domain descriptions matching annotated BGC components in *A. niger*.

For instance, backbone pseudo-annotations were assigned to genes containing similar descriptions to the annotated backbone genes in *A. niger*, such as polyketide synthases, non-ribosomal peptide synthetases, dimethylallyltryptophan synthases and terpene synthases. Tailoring enzymes pseudo-annotations were considered as genes containing similar descriptions of *A. niger* tailoring enzymes, such as methyltransferases, monooxygenases, and oxidoreductases. Transcription factor and transporter pseudo-annotations were assigned to genes presenting domains described as presenting these functions. A list of all Pfam domains associated with a pseudo-functional annotation is shown in Supplementary Table 5. The distribution of component pseudo-annotations found in the training data and gold-standard genes for *A. nidulans* is shown in Table 6.3.

Table 6.3: Domains linked to *A. nidulans* pseudo BGC components dataset genes

Pseudo	**Training**		**Test**	
component type	**BGCs**	**non-BGCs**	**gold BGCs**	**non-gold BGCs**
Backbones	17.5%	2.13%	20%	2.45%
Tailoring enzymes	36%	3.70%	31.63%	4.5%
Transcription factors	4.83%	2.35%	5.92%	3.92%
Transporters	5.82%	3.65%	7.55%	5.2%
Non-component domains	33.15%	48.28%	35.3%	62.12%
No domains	14.6%	41.15%	12.65%	22.8%
Total # genes	2833	1781	490	10002

Candidate BGCs for *A. nidulans* were obtained from TOUCAN, fungiSMASH, fungiSMASH combined with CASSIS, and DeepBGC in the same manner as candidates were obtained for *A. niger*, performing the test set pre-processing using a majority vote of overlapping sliding windows of fixed 10,000 amino acids as de-

scribed in Section 6.3.1 by the reinforcement learning agent on TOUCAN, fungiSMASH, and DeepBGC candidate BGCs for *A. nidulans* are shown in Table 6.4.

Table 6.4: Performance on *A. nidulans* candidate BGCs from the three tools

	gene metrics			**cluster metrics**			**average**	**% gold genes**	
model	P	R	F-m	P	R	F-m	F-m	negative	skipped
`TOUCAN`	0.272	0.681	0.389	1	0.685	0.813	0.601	32.24%	-
`TOUCAN-Q`	0.441	0.591	0.505	1	0.681	0.810	0.657	32.24%	13.47%
`TOUCAN-Q-all`	0.402	0.646	0.495	1	0.681	0.810	0.653	32.24%	7.55%
`fungiSMASH`	0.319	0.727	0.443	0.817	0.795	0.806	0.624	30.61%	-
`fungiSMASH-Q`	0.479	0.592	0.53	1	0.781	0.877	0.703	30.61%	15.92%
`fungiSMASH-Q-all`	0.469	0.605	0.529	1	0.736	0.848	0.688	30.61%	13.88%
`fungiSMASH/C`	0.318	0.762	0.449	1	0.792	0.884	0.666	28.16%	-
`fungiSMASH/C-Q`	0.484	0.581	0.528	1	0.778	0.875	0.702	28.16%	19.18%
`fungiSMASH/C-Q-all`	0.484	0.581	0.528	1	0.778	0.875	0.702	28.16%	19.18%
`DeepBGC`	0.328	0.493	0.394	0.723	0.466	0.567	0.480	50.61%	-
`DeepBGC-Q`	0.491	0.441	0.465	1	0.466	0.636	0.550	50.61%	8.57%
`DeepBGC-Q-all`	0.473	0.492	0.482	1	0.472	0.642	0.562	50.61%	2.86%

Like in *A. niger*, the reinforcement learning approach improved gene precision in candidate BGCs outputted by all three tools: an increase of 13%, 15%, 16.6%, and 14.5% is seen for `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/C-Q-all` and `DeepBGC-Q-all` respectively. Gene metrics also yield improvement in *A. nidulans* without harming the cluster metrics for `TOUCAN-Q-all`, while improving it for `fungiSMASH-Q-all` and `DeepBGC-Q-all`, and only showing a less than 1% difference for `fungiSMASH/C-Q-all`. As previously mentioned, this indicates that the reinforcement learning agent was able to improve the precision of candidate BGC components without discarding correctly predicted candidate BGC regions. Average F-m performance also showed improvement for all three tools when compared to their original candidate BGCs, with an increase of 5.2%, 6.4%, 3.6%, and 8.2% for `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/c-Q-all` and `DeepBGC-Q-all`. When comparing the models relying on the reinforcement learn-

ing agent only (`Q`) versus the ones relying on both the agent and the functional annotation strategies (`Q-all`) we can observe improvements on gene recall and the percentage of gold-standard genes skipped, but a small drop on gene precision, with the exception of `fungiSMASH/C` models that yield similar performance for `Q` and `Q-all` models. Likely, the usage of *A. nidulans* pseudo-annotations resulted in a slight increase of false positive components. However it might be an useful alternative when manually curated functional annotations are not available, or also when wanting to favor recall over precision.

Candidate BGC composition before and after applying the reinforcement learning agent is shown in Supplementary Figure 1. Similarly to *A. niger*, Supplementary Figure 1-A demonstrates improvements in candidate BGCs achieved by the agent by skipping non-BGC genes (in blue). When handling more complex cases, as shown in Supplementary Figure 1-B, the agent kept most non-BGC genes, potentially resulting in overpredicted boundaries. Approximately 50% of protein domains from non-BGC genes in Supplementary Figure 1-B were associated to pseudo-functional annotations in *A. nidulans*, while only 20% of domains from non-BGC genes in Supplementary Figure 1-A were associated to any annotation.

6.5 Discussion and Conclusion

Secondary metabolites are a crucial source of compounds that benefit human health. Identifying BGCs responsible for synthesizing these compounds in fungi may lead to the discovery of new natural products, and potentially novel drugs. State-of-the-art tools for BGC discovery often overpredict BGC boundaries and components. In fungi BGCs are typically encoded by a high diversity of components, known to vary even among evolutionary closely related species. Precise identification of BGC components is therefore a challenging task, and can facilitate the validation and experimental characterization of SM compounds. In this work

we presented a reinforcement learning method and functional annotation strategies to support optimizing fungal candidate BGCs obtained with state-of-the-art tools. We evaluated our proposed approach on candidate BGCs obtained for *A. niger* and *A. nidulans* by three BGC discovery tools: TOUCAN, based on supervised learning; fungiSMASH, based on probabilistic and rule-based methods, as well as a version of fungiSMASH combined with CASSIS for cluster border prediction; and DeepBGC, based on deep learning. The results obtained by our reinforcement learning approach yield improvement of cluster and gene precision of BGC candidates obtained from all three tools, without affecting correctly predicted BGC regions.

Overall, best average F-m performances obtained for *A. niger* relied on the combination of the reinforcement learning method and functional annotation strategies based on expert curation. In *A. nidulans*, even pseudo-functional annotations were able to improve BGC gene recall, and reduce the number of gold-standard genes being skipped by the reinforcement learning agent. This indicates that, when available, integrating functional annotations further advances the approach capabilities. Functional annotations may however not always be publicly available, since they can be time-consuming to obtain. The results have shown however that the reinforcement learning approach alone, based solely on Pfam protein domains, improved average F-m of candidate BGCs in average by 7% in *A. niger* and 5.8% in *A. nidulans*. The performance of the reinforcement learning approach indicates its ability to identify the relevance of certain protein domain profiles associated with fungal BGCs, supporting previous findings of these as relevant features in the context of BGC discovery (Khaldi et al., 2010; Cimermancic et al., 2014; Hannigan et al., 2019).

The results achieved through reinforcement learning in candidate BGCs from both fungal genomes evaluated are indicative of the method generalization power and

robustness by handling candidate BGCs from different organisms. Additionally a preliminary analysis, shown in Supplementary Figure 2, was performed by processing completely annotated MIBiG BGCs from three fungal species using the proposed reinforcement learning method. The fact that the completely annotated BGCs were kept almost intact by the reinforcement learning method, with or without functional annotation strategies is another indication of its potential robustness on properly identifying essential BGC components for the SM biosynthesis.

As discussed in Section 6.2, properly identifying BGC components can be a great challenge, given the underlying high diversity of BGCs. Moreover, another important challenge related to the scarcity of validated fungal BGC data are potential biases, both of cluster boundary definition, as well as of BGC composition, since most MIBiG fungal BGCs composing the training dataset are polyketide synthases. While reported as manually curated (Kautsar et al., 2020), most MIBiG fungal BGCs in the training dataset are partially annotated, and Inglis et al. (2013) presented limited experimental characterization evidence for the annotated *Aspergillus* BGCs considered as gold standard BGCs in this work. While the number of completely or partially annotated fungal BGCs is scarce, the number of experimentally characterized clusters is even smaller. This only highlights that improving the availability of validated and experimentally characterized fungal BGC data can be a fundamental step towards supporting the development of robust *in silico* approaches for fungal BGC discovery.

Data availability The source code as well as all datasets applied in our experiments are made publicly available at `https://github.com/bioinfoUQAM/RL-bgc-components`. All material is available under the MIT software license. The datasets used in this work were obtained from open access databases, which are

Acknowledgements

We acknowledge the bioinformatics teams at CSFG and UQAM; Sylvester Palys, Marie Beigas, and Isabelle Benoit for helping with manual curation of functional annotations; and Maria Victoria Aguilar and Amine Remita for helpful discussions and brainstorming.

Funding

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) and the Fonds de recherche du Québec – Nature et technologies (FRQNT).

Supplementary Table 1: Pfam protein domains associated with annotations of BGC components in *A. niger*

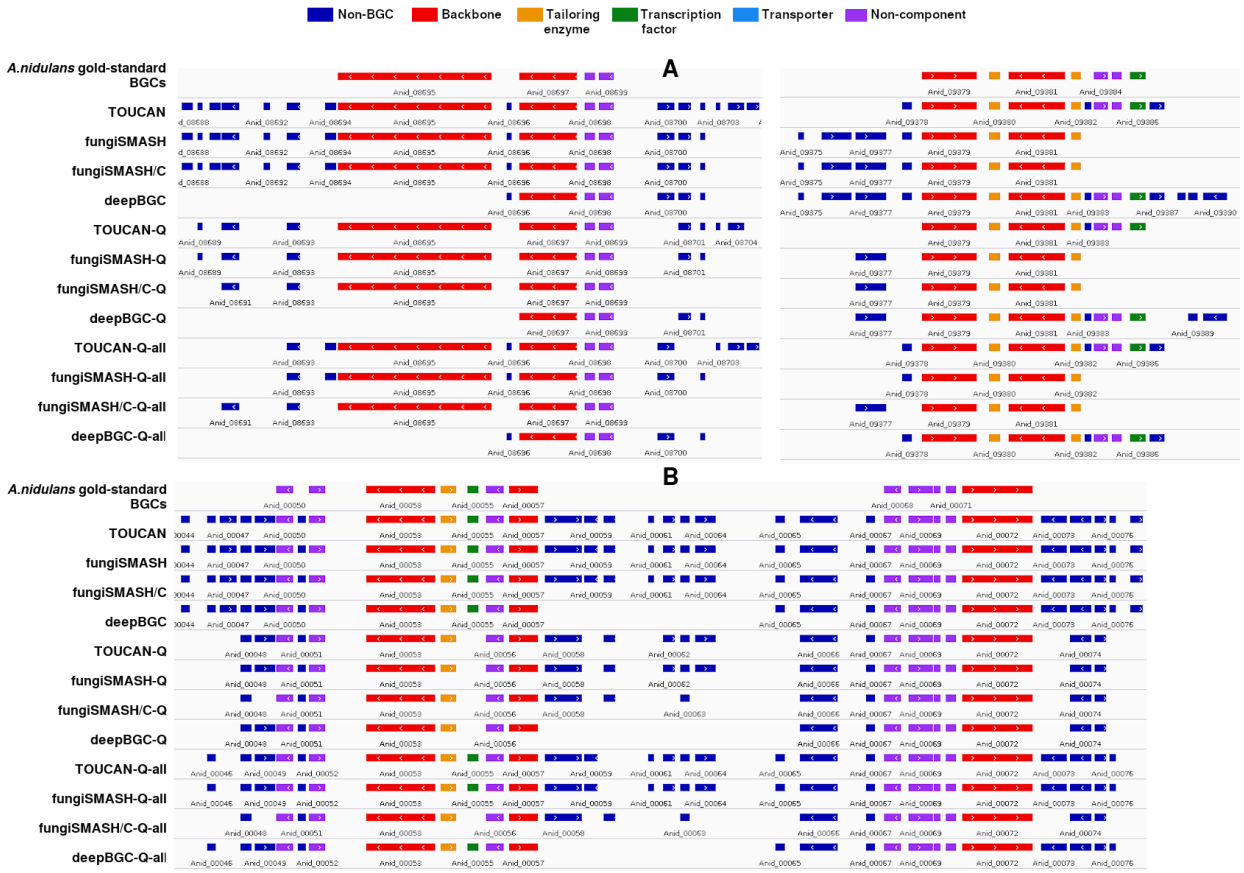
Pfam ID	Pfam Description	Component type	Pfam ID	Pfam Description	Component type
PF00106	adh_short	backbone	PF00076	RRM_1	tailoring enzyme
PF00107	ADH_zinc_N	backbone	PF00107	ADH_zinc_N	tailoring enzyme
PF00109	ketoacyl-synt	backbone	PF00109	ketoacyl-synt	tailoring enzyme
PF00195	Chal_sti_synt_N	backbone	PF00172	Zn_clus	tailoring enzyme
PF00501	AMP-binding	backbone	PF00176	SNF2_N	tailoring enzyme
PF00550	PP-binding	backbone	PF00271	Helicase_C	tailoring enzyme
PF00668	Condensation	backbone	PF00400	WD40	tailoring enzyme
PF00698	Acyl_transf_1	backbone	PF00501	AMP-binding	tailoring enzyme
PF00975	Thioesterase	backbone	PF00550	PP-binding	tailoring enzyme
PF01575	MaoC_dehydratas	backbone	PF00590	TP_methylase	tailoring enzyme
PF01648	ACPS	backbone	PF00668	Condensation	tailoring enzyme
PF02797	Chal_sti_synt_C	backbone	PF00698	Acyl_transf_1	tailoring enzyme
PF02801	Ketoacyl-synt_C	backbone	PF00743	FMO-like	tailoring enzyme
PF06330	TRI5	backbone	PF00891	Methyltransf_2	tailoring enzyme
PF07993	NAD_binding_4	backbone	PF00975	Thioesterase	tailoring enzyme
PF08240	ADH_N	backbone	PF01263	Aldose_epim	tailoring enzyme
PF08241	Methyltransf_11	backbone	PF01266	DAO	tailoring enzyme
PF08242	Methyltransf_12	backbone	PF01370	Epimerase	tailoring enzyme
PF08354	DUF1729	backbone	PF01408	GFO_IDH_MocA	tailoring enzyme
PF08659	KR	backbone	PF01494	FAD_binding_3	tailoring enzyme
PF11991	Trp_DMAT	backbone	PF01565	FAD_binding_4	tailoring enzyme
PF13193	AMP-binding_C	backbone	PF01717	Meth_synt_2	tailoring enzyme
PF13452	MaoC_dehydrat_N	backbone	PF02668	TauD	tailoring enzyme
PF13602	ADH_zinc_N_2	backbone	PF02801	Ketoacyl-synt_C	tailoring enzyme
PF13671	AAA_33	backbone	PF02894	GFO_IDH_MocA_C	tailoring enzyme
PF14765	PS-DH	backbone	PF03171	2OG-FeII_Oxy	tailoring enzyme
PF16073	SAT	backbone	PF04082	Fungal_trans	tailoring enzyme
PF16197	KAsynt_C_assoc	backbone	PF04191	PEMT	tailoring enzyme
PF17828	FAS_N	backbone	PF05721	PhyH	tailoring enzyme
PF17951	FAS_meander	backbone	PF07992	Pyr_redox_2	tailoring enzyme
PF18314	FAS_I_H	backbone	PF07993	NAD_binding_4	tailoring enzyme
PF18325	Fas_alpha_ACP	backbone	PF08031	BBE	tailoring enzyme
PF18558	HTH_51	backbone	PF08240	ADH_N	tailoring enzyme
PF00096	zf-C2H2	transcription factor	PF08241	Methyltransf_11	tailoring enzyme
PF00172	Zn_clus	transcription factor	PF08242	Methyltransf_12	tailoring enzyme
PF04082	Fungal_trans	transcription factor	PF08659	KR	tailoring enzyme
PF06331	Tfb5	transcription factor	PF13241	NAD_binding_7	tailoring enzyme
PF11951	Fungal_trans_2	transcription factor	PF13489	Methyltransf_23	tailoring enzyme
PF12157	DUF3591	transcription factor	PF13602	ADH_zinc_N_2	tailoring enzyme
PF00005	ABC_tran	transporter	PF13649	Methyltransf_25	tailoring enzyme
PF00083	Sugar_tr	transporter	PF13847	Methyltransf_31	tailoring enzyme
PF00664	ABC_membrane	transporter	PF14226	DIOX_N	tailoring enzyme
PF00854	PTR2	transporter	PF14765	PS-DH	tailoring enzyme
PF01061	ABC2_membrane	transporter	PF14823	Sirohm_synth_C	tailoring enzyme
PF01490	Aa_trans	transporter	PF14824	Sirohm_synth_M	tailoring enzyme
PF01544	CorA	transporter	PF16073	SAT	tailoring enzyme
PF03619	Solute_trans_a	transporter	PF16197	KAsynt_C_assoc	tailoring enzyme
PF06422	PDR_CDR	transporter	PF18558	HTH_51	tailoring enzyme
PF07690	MFS_1	transporter			

Supplementary Table 2: Parameter optimization for the reinforcement learning agent

Parameters			Penalty (500 episodes)		
α	ϵ	γ	**Max**	**Min**	**Average**
0.01	0.01	0.01	1336	34	50.81
0.01	0.01	0.1	1370	102	122.63
0.01	0.01	0.25	1370	129	148.50
0.01	0.01	0.5	1403	194	212.81
0.01	0.01	0.75	1394	207	223.89
0.01	0.01	1.0	1400	221	238.95
0.1	0.1	0.01	1477	283	336.46
0.1	0.1	0.1	1597	295	379.70
0.1	0.1	0.25	1601	328	437.76
0.1	0.1	0.5	1612	432	548.94
0.1	0.1	0.75	1732	646	734.96
0.1	0.1	1.0	1746	716	1349.31
0.25	0.25	0.01	1783	760	831.16
0.25	0.25	0.1	1864	759	843.07
0.25	0.25	0.25	1865	791	884.95
0.25	0.25	0.5	2026	904	1019.68
0.25	0.25	0.75	1977	1198	1335.59
0.25	0.25	1.0	2756	1697	2458.74
0.5	0.5	0.01	2370	1515	1661.68
0.5	0.5	0.1	2370	1587	1681.00
0.5	0.5	0.25	2411	1596	1732.92
0.5	0.5	0.5	2495	1795	1908.18
0.5	0.5	0.75	2596	2131	2249.36
0.5	0.5	1.0	3150	2570	2964.74
0.75	0.75	0.01	2827	2367	2489.42
0.75	0.75	0.1	2802	2390	2508.21
0.75	0.75	0.25	2878	2441	2553.75
0.75	0.75	0.5	2882	2558	2684.29
0.75	0.75	0.75	3018	2754	2886.99
0.75	0.75	1.0	3327	3061	3209.54
1.0	1.0	0.01	3440	3197	3317.77
1.0	1.0	0.1	3457	3199	3318.97
1.0	1.0	0.25	3446	3199	3319.27
1.0	1.0	0.5	3434	3190	3315.63
1.0	1.0	0.75	3442	3171	3318.10
1.0	1.0	1.0	3443	3179	3320.03

Supplementary Table 3: *keepSkip* parameter optimization for the reinforcement learning agent

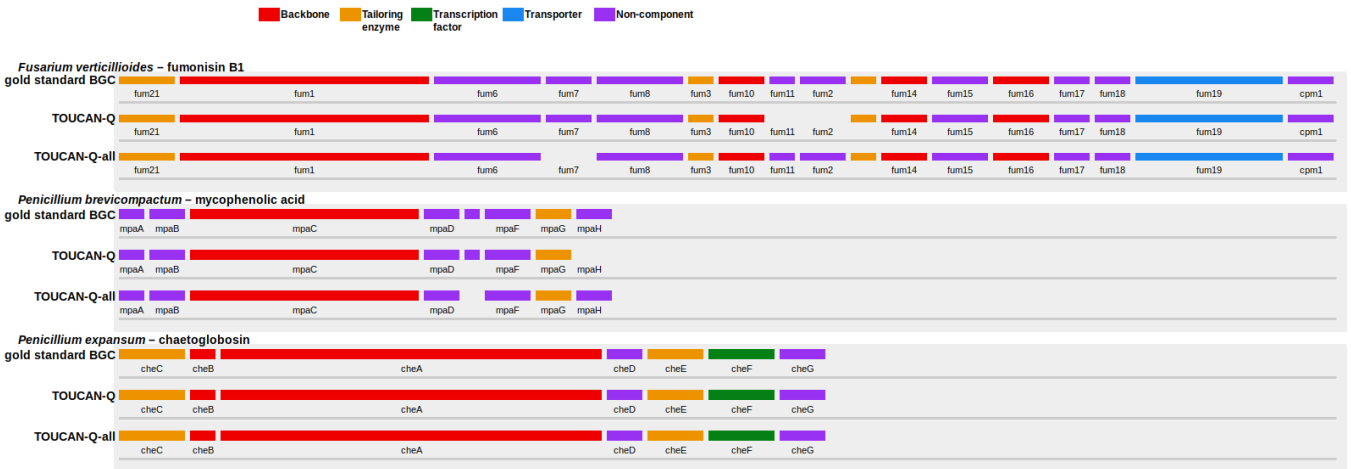
Parameter	Penalty (500 episodes)		
keepSkip	**Max**	**Min**	**Average**
0.1	1170	31	48.622
0.25	1194	31	48.718
0.5	1241	31	47.452
0.75	1239	30	48.762
1	1336	34	50.814
1.25	1343	33	51.428
1.5	1349	32	51.534



Supplementary Figure 1: Comparison between gold-standard and candidate BGC composition for four *A. nidulans* clusters. Non-BGC genes are shown in dark blue. (**A**) Candidate BGCs for which the reinforcement learning agent correctly skipped most non-BGC genes compared to their non-ribosomal peptide (left) and polyketide (right) gold standard BGCs. (**B**) Candidate BGCs for which the agent kept most non-BGC genes compared to their polyketide (left) and polyketide/non-ribosomal peptide (right) gold standard BGCs, possibly due to their ambiguous protein domains, which approximately half are associated to BGC component roles but do not belong to neighboring clusters.

Supplementary Table 4: Percentage of *A. niger* BGC components in gold-standard genes present in candidate BGCs

Component type	TOUCAN (classified positive)	TOUCAN-Q (keep)	TOUCAN-Q-all (keep)
Backbones	92.9%	92.9%	92.9%
Tailoring enzymes	91.9%	83.9%	87.1%
Transcription factors	89.2%	78.4%	86.5%
Transporters	86.9%	78.3%	84.8%
Non-component domains	87.6%	63.0%	68.5%
No domains	78.3%	0.0%	41.2%
Component type	fungiSMASH (classified positive)	fungiSMASH-Q (keep)	fungiSMASH-Q-all (keep)
Backbones	70.7%	70.7%	70.7%
Tailoring enzymes	61.3%	56.4%	59.7%
Transcription factors	62.2%	51.3%	56.7%
Transporters	71.7%	65.2%	69.6%
Non-component domains	65.6%	44.8%	47.7%
No domains	67.0%	0.0%	39.2%
Component type	DeepBGC (classified positive)	DeepBGC-Q (keep)	DeepBGC-Q-all (keep)
Backbones	64.6%	64.6%	64.6%
Tailoring enzymes	66.1%	61.3%	62.9%
Transcription factors	43.2%	40.5%	43.2%
Transporters	45.6%	43.5%	45.6%
Non-component domains	46.1%	34.4%	35.7%
No domains	36.1%	0.0%	25.8%



Supplementary Figure 2: Comparison between completely annotated MIBiG BGCs before and after processed by the proposed reinforcement learning method (with and without functional annotation strategies) for fumonisin B1 from *Fusarium verticillioides*, mycophenolic acid from *Penicillium brevicompactum*, and chaetoglobosin from *Penicillium expansum*. Post-processed BGCs were kept almost intact by the reinforcement learning method, potentially indicating its robustness on identifying relevant components for SM biosynthesis.

Supplementary Table 5: Protein domains associated with pseudo-annotations of BGC components in *A. nidulans*

Pfam ID	Pfam Description	Component type	Pfam ID	Pfam Description	Component type
PF00106	adh_short	backbone	PF02133	Transp_cyt_pur	transporter
PF00107	ADH_zinc_N	backbone	PF07690	MFS_1	transporter
PF00109	ketoacyl-synt	backbone	PF13577	SnoaL_4	transporter
PF00326	Peptidase_S9	backbone	PF00067	p450	tailoring enzyme
PF00501	AMP-binding	backbone	PF00106	adh_short	tailoring enzyme
PF00550	PP-binding	backbone	PF00248	Aldo_ket_red	tailoring enzyme
PF00668	Condensation	backbone	PF00296	Bac_luciferase	tailoring enzyme
PF00698	Acyl_transf_1	backbone	PF00550	PP-binding	tailoring enzyme
PF00755	Carn_acyltransf	backbone	PF00668	Condensation	tailoring enzyme
PF00975	Thioesterase	backbone	PF00743	FMO-like	tailoring enzyme
PF01575	MaoC_dehydratas	backbone	PF00881	Nitroreductase	tailoring enzyme
PF01583	APS_kinase	backbone	PF00891	Methyltransf_2	tailoring enzyme
PF01648	ACPS	backbone	PF01370	Epimerase	tailoring enzyme
PF02801	Ketoacyl-synt_C	backbone	PF01494	FAD_binding_3	tailoring enzyme
PF07859	Abhydrolase_3	backbone	PF01565	FAD_binding_4	tailoring enzyme
PF07993	NAD_binding_4	backbone	PF03171	2OG-FeII_Oxy	tailoring enzyme
PF08240	ADH_N	backbone	PF04140	ICMT	tailoring enzyme
PF08242	Methyltransf_12	backbone	PF05063	MT-A70	tailoring enzyme
PF08354	DUF1729	backbone	PF05368	NmrA	tailoring enzyme
PF08659	KR	backbone	PF05721	PhyH	tailoring enzyme
PF11991	Trp_DMAT	backbone	PF08031	BBE	tailoring enzyme
PF13193	AMP-binding_C	backbone	PF13450	NAD_binding_8	tailoring enzyme
PF13452	MaoC_dehydrat_N	backbone	PF13460	NAD_binding_10	tailoring enzyme
PF13489	Methyltransf_23	backbone	PF13489	Methyltransf_23	tailoring enzyme
PF13602	ADH_zinc_N_2	backbone	PF13649	Methyltransf_25	tailoring enzyme
PF14765	PS-DH	backbone	PF14226	DIOX_N	tailoring enzyme
PF16073	SAT	backbone	PF00009	GTP_EFTU	transcription factor
PF16197	KAsynt_C_assoc	backbone	PF00172	Zn_clus	transcription factor
PF17951	FAS_meander	backbone	PF00249	Myb_DNA-binding	transcription factor
PF18314	FAS_I_H	backbone	PF00320	GATA	transcription factor
PF18325	Fas_alpha_ACP	backbone	PF03143	GTP_EFTU_D3	transcription factor
PF18558	HTH_51	backbone	PF03144	GTP_EFTU_D2	transcription factor
PF00005	ABC_tran	transporter	PF04082	Fungal_trans	transcription factor
PF00083	Sugar_tr	transporter	PF08447	PAS_3	transcription factor
PF00149	Metallophos	transporter	PF08493	AflR	transcription factor
PF00153	Mito_carr	transporter	PF08938	HBS1_N	transcription factor
PF00324	AA_permease	transporter	PF10297	Hap4_Hap_bind	transcription factor
PF00664	ABC_membrane	transporter	PF11951	Fungal_trans_2	transcription factor
PF01544	CorA	transporter	PF13921	Myb_DNA-bind_6	transcription factor

CONCLUSION

The benefits brought by the discovery of secondary metabolites are remarkable, and have had significant impact on human health throughout the last decades. The demand for identifying new therapeutics increases, especially in the face of the surge in antibiotics drug resistance seen in cases such as cancer treatments (Vasan et al., 2019), and in treatment of pathogens such as *Candida* and *Aspergillus* species (Berman & Krysan, 2020), as well as in *Salmonella*, *Staphylococcus*, *Streptococcus* species (Aslam et al., 2018; Liu et al., 2020). Filamentous fungi show phenomenal potential to unveil a wide range of secondary metabolites (Bills & Gloer, 2016), with pharmaceutical properties or other applications that are yet to be uncovered.

This thesis addresses the challenges related to the discovery of biosynthetic gene clusters encoding the metabolic pathways that synthesize secondary metabolites in fungi, and proposes a robust machine learning based approach to accurately identify candidate BGCs. The first main contribution of this thesis is the development of benchmark datasets to support fungal BGC discovery and facilitate modeling the problem as a classification task, presented in Chapter 4. This addresses the data scarcity challenge presented in Section 2.1 and the first hypothesis of this thesis presented in Section 2.2, since the proposed benchmark datasets were designed to represent a variety of fungal genomic profiles relevant to BGC discovery, relying on curated fungal BGCs and orthologous regions from a variety of fungal families and species. Moreover the approach presented in Chapter 4 supports development of supervised learning approaches, since fungal orthologous genes are evaluated as relevant, discriminant non-BGC data for the discovery task.

The second main contribution of this thesis is the development of TOUCAN, a supervised learning framework to identify candidate BGC regions in fungi, capable of outputting predictions even in newly sequenced and non-annotated genomes. The TOUCAN framework, presented in Chapter 5, relies on classification models built based on k-mers, Pfam protein domains, and GO terms extracted from the previously proposed benchmark datasets. To obtain BGC predictions on new fungal data, test genomes were represented as sliding windows of fixed amino acid length. The framework allows candidate BGC predictions to be post-processed with different strategies to improve BGC regions. This addresses the challenge of discovering BGC regions and the second hypothesis of this thesis, presented respectively in Sections 2.1 and 2.2, since the TOUCAN framework demonstrates that the BGC discovery task can be modeled as a supervised learning problem, relying on robust benchmark datasets and able to handle the genomic diversity of fungal genomes.

The third contribution of this thesis is the development of a reinforcement learning approach to enhance composition of BGC predictions, presented in Chapter 6. The reinforcement learning approach is based on signature protein domains found in the fungal BGC benchmark datasets. It employs specific strategies based on functional annotations of BGC components, such as backbone, tailoring enzymes, transcriptional factors and transporters, to help increase the quality and improve the boundaries of candidate BGC predictions, potentially overcoming the common issue of overprediction of BGC regions encountered by many state-of-the-art tools. This addresses the challenge of defining BGC composition and boundaries, and the third hypothesis of this thesis, presented respectively in Sections 2.1 and 2.2, since the reinforcement learning approach manages to optimize the components of BGC predictions obtained with previous tools, which will potentially reduce the manual curation effort required to validate these candidates.

Achieving accurate BGC predictions in fungi imposes great challenges due to the limited number of known clusters curated to date, and due to the high genomic diversity of BGC regions, a consequence of the complex metabolic pathways that synthesize fungal secondary metabolites. Beyond these challenges to obtain *in silico* BGC predictions, experimental characterization and reproduction of these compounds is also a complex and expensive task (Pickens et al., 2011; Rahmat & Kang, 2020), since very often BGC genes are found to be silent or poorly expressed under laboratory conditions (Montiel et al., 2015; Zhang et al., 2019). Automatic approaches for BGC discovery that are able to overcome these challenges and generate accurate BGC predictions play an important role in facilitating the next steps in the production pipeline and reducing the cost of identifying and characterising valuable compounds.

Limitations While substantial efforts were dedicated to developing the machine learning approach described in this thesis, certain questions remain open and represent limitations associated with the methodology applied, which would benefit from further work to address them. First, the evaluations presented in Sections 4.5, 5.4, and 6.4 were performed with a set of gold standard BGCs manually curated from the *Aspergillus niger* genome, and reproducibility was evaluated based on previously published *Aspergillus nidulans* BGCs. Both *Aspergillus niger* and *Aspergillus nidulans* are model organisms for many research fields such as cell biology, genetics, physiology, enzyme biochemistry, protein secretion, and fermentation processes (Cerqueira et al., 2014; Cairns et al., 2018; Kumar, 2020). But the evaluation on these two species represents only a small fraction of the possible applications of the proposed methods in fungi. The lack of more fungal genomes manually curated for BGCs can limit a broad evaluation of machine learning approaches. It is not surprising however that this data are scarce, given the cost and

effort associated with the process of discovery and validation of these compounds.

The methodology applied to generate candidate BGC instances from test fungal genomes was based on extracting a sliding window of fixed amino acid length. While this methodology allows for newly sequenced and non-annotated genomes to be tested for candidate BGCs, it can be limiting for certain organisms or even for certain BGC types, cases in which sliding windows of variable length might be more suitable. The fixed length window may also contribute to under or overprediction of BGC regions, since previously curated BGCs are known to differ in size. It could be interesting to study applying variable lengths when generating test candidate BGCs, and evaluating its effect on coverage of true positive BGC regions, as well as its suitability to different fungal genomes.

The work described in Chapter 5 was based on standard supervised learning algorithms, and one might wonder whether the application of deep learning methods would be more suitable for the task. Preliminary evaluations demonstrated however a quick overfitting behavior generally around approximately 10 epochs at the same time showed weak generalization capability on test data. This behavior was observed when using different models − such as LSTM, GRU, or fully connected networks −, and irrespective of suitable hyperparameters. An additional analysis of the performance of deep learning approaches in this task was provided with the state-of-the-art performance comparisons, when models built with a deep learning based tool previously developed for bacteria, but with hyperparameters specifically tuned to fungal data (thanks to the authors) was generally outperformed by the methods presented in this thesis as well as other state-of-the-art approaches. The data availability is likely the most important limitation to apply deep learning approaches to tackle fungal BGC discovery. However, when increased curated data become publicly available, it will be worthwhile to revisit and re-evaluate the application of deep learning methods to identify fungal BGCs.

The cluster metrics applied to evaluate performance on Sections 4.5, 5.4 and 6.4 consider the presence of a minimum of one gold-standard gene in a candidate BGC for it to be accounted as a true positive prediction. This criterion focuses on the retrieval capability of the presented models, favouring the recall of clusters and minimizing the number of missing true positive regions. However, this metric might favour the overestimation of cluster boundaries. The gene metrics counterbalance the cluster metrics perspective, because it compares the entirety of genes within a candidate BGC versus its corresponding gold standard cluster. An interesting analysis could be drawn from creating a new cluster metric criterion, identifying an evaluation threshold with the minimum accepted proportion of gold standard BGC genes within a candidate BGC.

Future directions There are many opportunities for advancement in the search for novel secondary metabolites through the application of automatic approaches. In general, the usage of machine learning methods to identify BGCs is in its infancy, especially when it comes to fungal data, the TOUCAN framework proposed in this thesis being the first supervised learning approach to be developed for discovering BGCs in these organisms. As well, the reinforcement learning approach proposed in this thesis is also the first implementation of such methods to tackle this task. The designs of both the TOUCAN framework and the reinforcement learning approach allow for continuous experimentation during their application in new test data, which can help to improve the discriminative power and prediction output of the proposed methods, and consequently benefit the manual curation and experimental characterization steps following BGC prediction.

Filamentous fungi have a strong potential to produce novel compounds especially the polyketides and nonribosomal peptides within the ascomycetes, terpenoids

within basidiomycetes, and terpenoids and polyketides within agaricomycetes (Bills & Gloer, 2016). Therefore, a precious opportunity awaits the application of BGC discovery approaches in more fungal genomes to identify relevant candidate regions, following their validation, characterization and ultimately the compound reproduction.

The BGC discovery approach presented in this thesis evaluated proposed datasets, a tool to identify candidate BGC regions, and a tool to improve predicted candidate BGCs based on signature protein domains and BGC components. This pipeline could also benefit from a tool to predict BGC product type and activity, focusing on fungi. Most previous approaches to analyse BGC activity and product, as presented in Section 3.2, were developed for bacteria. Predicting BGC product and activity will provide further context to support the following steps of curation and experimental characterization of candidate BGCs.

The methods presented in this thesis provide a robust pipeline to identify candidate BGC regions in fungal genomes that potentially synthesize secondary metabolite compounds. While many challenges still persist in generating accurate predictions, overcoming re-discovery of existing compounds and facilitating their reproduction, the approach proposed here showed the benefits of exploring machine learning methods to tackle fungal BGC discovery.

APPENDIX A

Supporting supervised learning in fungal Biosynthetic Gene Cluster discovery: new benchmark datasets

Hayda Almeida[1,2] Adrian Tsang[2] Abdoulaye Baniré Diallo[1]

[1]*University of Quebec in Montreal, Montreal, Canada*
[2]*Concordia University, Montreal, Canada*
Corresponding author: diallo.abdoulaye@uqam.ca

***Abstract*—Fungal Biosynthetic Gene Clusters (BGCs) of secondary metabolites are clusters of genes capable of producing natural products, compounds that play an important role in the production of a wide variety of bioactive compounds, including antibiotics and pharmaceuticals. Identifying BGCs can lead to the discovery of novel natural products to benefit human health. Previous work has been focused on developing automatic tools to support BGC discovery in plants, fungi, and bacteria. Data-driven methods, as well as probabilistic and supervised learning methods have been explored in identifying BGCs. Most methods applied to identify fungal BGCs were data-driven and presented limited scope. Supervised learning methods have been shown to perform well at identifying BGCs in bacteria, and could be well suited to perform the same task in fungi. But labeled data instances are needed to perform supervised learning.Openly accessible BGC databases contain only a very small portion of previously curated fungal BGCs. Making new fungal BGC datasets available could motivate the development of supervised learning methods for fungal BGCs and potentially improve prediction performance compared to data-driven methods. In this work we propose new publicly available fungal BGC datasets to support the BGC discovery task using supervised learning. These datasets are prepared to perform binary classification and predict candidate BGC regions in fungal genomes. In addition we analyse the performance of a well supported supervised learning tool developed to predict BGCs.**

***Index Terms*—biosynthetic gene clusters, secondary metabolites, supervised learning, BGC, fungi, dataset**

I. Introduction

Natural products (NPs) are specialized bioactive compounds primarily produced by plants, fungi and bacteria. NPs are a vital source for drugs: from anti-cancer, anti-virus, and cholesterol-lowering medications to antibiotics, and immunosuppressants [1]–[3]. Unlike those in plants, genes involved in the biosynthesis of many NPs in bacteria and fungi are co-localized in the genome of organisms and usually organized as clusters of genes [4]. Gene clusters capable of producing NPs are known as Biosynthetic Gene Clusters (BGC).

The task of identifying new BGCs could potentially lead to the discovery of novel NPs to benefit human health. However this task involves complex and costly processes, as well as the analysis of large amounts of biological data. Development of automatic tools that can support the identification of BGCs is therefore highly relevant. Various approaches have been used to develop such tools, such as data-driven methods, probabilistic methods, and supervised learning methods. In supervised learning the BGC discovery task can be represented as binary classification task. The goal in a binary classification task is to classify data instances as belonging to one out of two different categories. A binary classification BGC dataset would therefore be composed of positive and negative BGC instances.

Supervised learning has been previously used to predicting bacterial BGCs [5], [6] and shown to perform well. Supervised learning methods however are developed primarily based on annotated datasets, for which all instances are labeled as belonging to a specific class. Unlike for bacteria, the number of known fungal BGC data previously validated by curators is rather limited. The Minimum Information about a Biosynthetic Gene cluster (MIBiG) [7][1] repository is one of the largest freely available BGC databases. As an example of the disparity between known available BGC from bacteria versus fungi that has been annotated by curators,MIBiG holds over 1,196 bacteria BGCs, while only 206 are fungal BGCs[2].

Generating fungal BGC datasets for supervised learning approaches imposes a few challenges. For instance, negative samples are needed for binary classification, and they are not directly provided by BGC databases just as annotaded BGC data. To be able to support a robust classification approach, fungal BGC datasets used as input should include a variety of organisms and BGC types to properly represent fungal genomic profiles.

The availability of fungal BGC datasets could leverage the development of new supervised learning approaches to tackle BGC discovery in fungi. This work presents new datasets prepared to tackle fungal BGC discovery as a binary classification task. These datasets are constructed in such way that they include most variety of BGC types from different organisms, attempting to represent fungal genomic profiles to better suit the fungal BGC classification task. Finally we also analyse

[1]http://mibig.secondarymetabolites.org/
[2]As of July 2019.

978-1-7281-1867-3/19/$31.00 ©2019 IEEE

the usage of fungal BGC datasets with one of the state-of-the-art supervised learning methods developed for BGC discovery, DeepBGC [6].

II. Previous Work

In this section we present previous work on the availability of BGC data previously predicted or annotated by curatorsthat can support BGC discovery, and previous work conducted towards developing automatic approaches to identify fungal BGCs. BGC databases and some of their characteristics are discussed in Section II-A. Previous work on predicting BGCs in fungi is presented in Section II-B.

A. BGC Databases

Only a small number of open access BGC databases is currently available to support research on automatic tools to identify BGCs. The majority of entries in these databases corresponds to bacteria data, while only a small portion are fungal BGCs.[3] MIBiG is a BGC repository in which curated entries are submitted by curators, and added to the database in a format compliant with the Minimum Information about any Sequence (MIxS) framework data standard. It holds 206 fungi BGCs and 1,196 for bacteria. Clustermine360 [8] contains microbial polyketide synthases (PKS) and non-ribosomal peptide synthetases (NRPS) biosynthesis. It holds a total of 29 fungal BGCs, while over 900 are from bacteria. Clustermine360 entries are curated and submitted by curators, enriched with information from the National Center for Biotechnology Information (NCBI)[4], and analysed with the antiSMASH [9] tool. The antiSMASH database [10] has 24,773 microbial BGCs predicted based on its homonymous tool. Unlike its bacteria version, the fungal version of antiSMASH does not provide a database of fungal BGCs to the best of our knowledge.

The Integrated Microbial Genomes: Atlas of Biosynthetic Gene Clusters [11] (IMG/ABC) database contains BGCs predicted using the ClusterFinder algorithm [12]. IMG/ABC holds 127 fungal BGCs and 1,025 from bacteria.

These databases are not connected. Since it is likely that there are overlaps among the different databases, the number of unique fungal BGCs could be even smaller. The small proportion of fungal BGCs across databases is an example of the challenges in developing automatic tools to tackle BGC discovery in fungi. This work proposes new publicly available datasets to be an input of supervised learning tools to predict fungal BGCs, based on MIBiG and orthologous genes. The details on our datasets and their analysis are discussed in Section III.

B. BGC discovery in Fungi

Significant effort has been put towards developing approaches to discover BGCs [2], [3]. The majority of approaches focused on processing bacterial data, while some of them are specially focused on fungi. Identifying BGCs remains a challenging task specially in fungal genomes, due to the diversity of clusters [13].

Previous work on fungal BGC discovery made use mostly of data-driven methods, which are heavily based on the analysis of the input or output data and require fine parameter-tuning. These methods required as input the genome sequence combined with transcription data [14], [15], or gene functional annotations [16], as well as both nucleotide and amino acid sequences [17]. [14] and [15] focused on analysing similar gene expression levels, while [15] used virtual clusters. [14] looked at motif co-occurrence in promoters around anchor genes, and [17] analysed homologous genes through a comparative genomics approach.

Such data-driven methods are less dependent on curated-BGC data, which are time consuming to obtain, but they all present limitations. [16] requires gene functional annotations, which may not be available, and [14] relies heavily on manual curation of output to achieve the expected results. A very limited BGC prediction scope is considered in [18] and [17]. Both approaches are developed based on biological sequences from a single species, and they also require fine parameter-tuning. Such limitations of data-driven methods can restrict their ability to generalize to new data, and as a consequence compromise the discovery of novel BGCs.

Likely due to the larger availability of curatedBGC data, probabilistic [9], [12], [19] and machine learning approaches [5], [6] have been more explored in bacteria compared to fungi, and shown to perform well. Probabilistic and machine learning approaches could be beneficial for BGC discovery, since by nature they are more capable of generalizing given new data, and will likely perform better at identifying data patterns and discovering novel BGCs, when compared to data-driven methods. In this study we also analyse the performance of a supervised learning approach developed to tackle BGC discovery using the fungal BGC datasets proposed by our work. The details on our experimental setup are further discussed in Section III.

III. Methodology

Some of the challenges in generating fungal BGC datasets for binary classification are the need of negative instances, which are not directly provided in BGC databases; and accounting for a variety of organisms, BGC types, and also fungal genomic profiles. The availability of new fungal BGC datasets however could potentially motivate the development of supervised learning approaches to tackle fungal BGC discovery.

In this work we propose new publicly available fungal BGC datasets to support supervised learning approaches tackling BGC discovery as a binary classification task. We present here the methodology adopted to prepare fungal BGC datasets and their analysis using a supervised learning method, with the goal of analysing the method performance in fungal BGC data.

Details on our proposed fungal BGC datasets are presented in Section III-A. Section III-B presents the test datasets with which we analysed the performance of classification models

[3]Number of entries for databases are reported as of July 2019.
[4]https://www.ncbi.nlm.nih.gov/

built on fungal BGC datasets. In Section III-C we provide details on the parameters considered in our analysis based on a supervised learning method, as well as the classification models considered.

A. Proposed Datasets

Supervised learning was shown to perform well at BGC discovery in previous work that focused on handling bacteria data [5], [6]. Given that annotated data are needed to perform a supervised learning approach, we propose here fungal BGC datasets to support the development of this approach for fungi.

As mentioned in Section I, positive and negative instances are needed to perform fungal BGC discovery as a binary classification task using supervised learning. To create our fungal BGC datasets, we extracted and filtered positive instances from the MIBiG [7] repository, previously presented in Section II-A. MIBiG has the highest number of unique fungal BGCs among the BGC databases previously presented. Additionally, MIBiG BGCs were annotated and submitted by the research community, unlike BGCs in other databases that were automatically predicted.

From all MIBiG instances, we have selected only the fungal BGC subset, excluding BGCs belonging to *Aspergillus niger* (*A. niger*) to avoid overlaps during the test phase, resulting in a total of 200 positive instances.

We generated synthetic negative instances collecting and integrating orthologous genes from OrthoDB[5] [20]. Orthologs are homologous genes descendants from a single gene of a last common ancestor. The OrthoDB database contains protein-coding genes that represent the last common ancestors given a specific phylogeny radiation of a species, and are therefore known to retain ancestral function [20]. Orthologs represent regions conserved across species. They can correspond to a relevant negative instances for BGC discovery. this is due to the fact that fungal BGCs are known to have opposite characteristics and show large genomic diversity even in otherwise closely-related or same genus species [13]. Genes belonging to fungal BGCs have been previously referred to as "species-specific" [21], unlike orthologs.

Orthologous genes have been previously used to discover BGCs in fungi. In [17], the authors presented an alignment-based approach focused on identifying syntenic block regions, which are more likely to contain orthologs and less likely to contain BGCs. Non-syntenic blocks were then used to search for candidate BGCs and to better define candidate cluster boundaries. The approach in [17] was explored in small set of 10 filamentous fungi. The results showed good performance, predicting correctly 21 out of 24 fungal BGCs.

In this study we selected the fungal OrthoDB subset to construct the synthetic negative BGC instances. The OrthoDB fungal subset contains a total of 5,083,652 non-redundant orthologs. To avoid potential overlaps, we performed a BLAST analysis between the fungal subsets of both OrthoDB and MIBiG. We discarded 11,000 ortholog matches found using the BLAST parameter $evalue$ (expected value) set to $1e-60$.

[5]http://orthodb.org/

To generate synthetic negative instances, we then concatenated the amino acid sequence of fungal orthologs using a fixed length of 7,000 amino acids to create synthetic gene clusters. The 7,000 amino acid length is chosen since itcorresponds to the average length of fungal BGC amino acid sequences in MIBiG. Figure 1 shows an example of positive instances in our datasets and negative instances being generated from OrthoDB orthologs. After processing OrthoDB fungal orthologs a total of 693,195 synthetic negative clusters were generated.

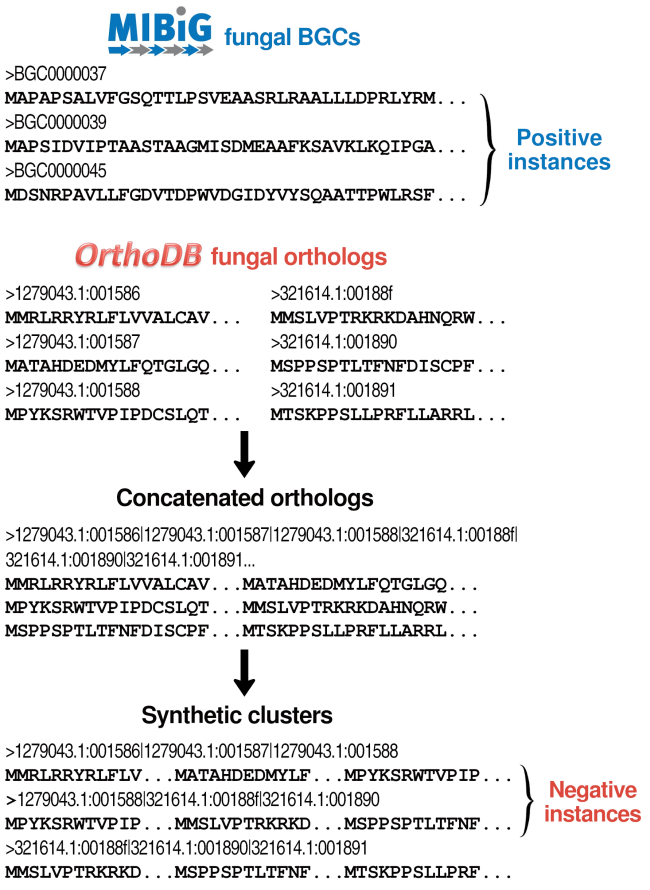


Fig. 1. Example of positive instances and the process to generate synthetic negative instances from orthologs

The MIBiG fungal subset and the pool of OrthoDB synthetic negative clusters were then considered to generate fungal BGC datasets with different distributions of positive and negative instances. Among the MIBiG fungal subset the annotated BGC regions corresponded in average to $\approx$1% of the total genome length of an organism, which provides a hint on the imbalance in class distribution that can be seen in a real test case scenario. Due to the natural imbalance of BGC regions versus non-BGC regions in a genome, we are interested in analysing the performance of a supervised learning approach based on datasets with various distributions of positive and negative instances. To analyse this aspect, we generated fungal BGC datasets with varying distributions by increasing the number of synthetic negative instances randomly selected from the

OrthoDB synthetic negative clusters pool. Table I shows the positive vs. negative distributions in each dataset.

TABLE I
DISTRIBUTION OF INSTANCES ACROSS FUNGAL BGC DATASETS

	Train		Validation	
Dataset	Pos	Neg	Pos	Neg
50%-50%	160	160	40	40
40%-60%	160	240	40	60
30%-70%	160	373	40	93
20%-80%	160	640	40	160
10%-90%	160	1,440	40	360
05%-95%	160	3,040	40	760
01%-99%	160	15,840	40	3,960

To generate classification models based on a supervised learning method, we extracted Pfam [22][6] IDs from the positive and negative instances. All datasets were converted into `pfamtsv` format [6], which is required as input in the supervised learning approach applied in this work. For each dataset, 80% were randomly selected for the training phase, while 20% were held out for the validation phase, as shown in Table I.

B. Test Datasets

To analyse the performance of classification models built based on fungal BGC datasets, we selected a fungal genome from the *Aspergillus* genus to represent a real test case scenario. *Aspergillus* is the most frequent genus among fungal species in MIBiG, together with *Penicillium*. For this evaluation we focused specifically on the *A. niger* species. *A. niger* is a genome of interest due to its biological diversity and major relevance to industrial processes [23]. In [24] the authors present manual annotation of BGCs in *Aspergilli*, among which a total of 79 BGCs are found in *A. niger*.

To generate candidate clusters for the test phase, we collected a manually curated *A. niger* genome sequence made publicly available through the Genozymes project[7]. We generated test candidate clusters by considering a sliding window of 30,000 nucleotides in the *A. niger* genome. The 30,000 sliding window length is defined based on the average length of the nucleotide sequence of MIBiG fungal BGCs. A similar approach was previously applied in fungal BGC discovery to generate virtual clusters [15].

The 30,000 sliding window was shifted along the genome using either a 50% or a 30% overlap. The overlaps in a sliding window mean that each test candidate cluster will contain the last 15,000 nucleotides (if a 50% overlap) or the last 9,000 nucleotides (if a 30% overlap) of the immediate previous candidate cluster. With the strategy of generating candidate clusters using overlaps, we are more likely to cover regions in between two or more genes. Figure 2 shows an example of candidate clusters being generated from *A. niger* genes using overlaps. The test datasets based on a 50% overlap contains a total of 1,184 candidate clusters, while the test datasets based on a 30% overlap contains a total of 846 candidate clusters.

Fig. 2. Example of *A. niger* candidate clusters generated for test phase

C. Classification Models

In this section we describe the methods applied to analyse the performance of a supervised learning approach using the fungal BGC datasets presented in Section III-A and the test data presented in Section III-B. To generate classification models with our fungal BGC datases, we utilized the DeepBGC system [6]. DeepBGC executable, source code and other resources are openly available[8]. Among these resources, there are pre-built BGC classification models and word2vec-based embeddings built using Pfam IDs, referred to as pfam2vec embeddings. In [6] the authors explained that pfam2vec embeddings were trained based in a skipgram architecture with 100 dimensions and over 15,686 unique Pfam IDs. DeepBGC classification is based on a Bidirectional Long Short Term Memory (BiLSTM) neural network, for which the input are pfam2vec embeddings. In [6] DeepBGC hyperparameters are described as a BiLSTM layer size of 128, dropout of 0.2, sigmoid activation, batch size of 64, 256 timestamps over 328 epochs, using Adam optimizer at a learning rate of 1e-4, with weighted binary cross-entropy loss. To generate classification models using fungal BGC datasets on the DeepBGC system we adopted the same hyperparameters described in [6], as well as the pfam2vec embeddings as input for training. For each fungal BGC dataset, we have generated a different

[6]http://pfam.xfam.org
[7]https://gb.fungalgenomics.ca/portal/
[8]https://github.com/Merck/deepbgc

classification model using DeepBGC. Fungal BGC models are named by their positive instance percentage:

- `pos50` (50%-50%)
- `pos40` (40%-60%)
- `pos30` (30%-70%)
- `pos20` (20%-80%)
- `pos10` (10%-90%)
- `pos05` (05%-95%)
- `pos01` (01%-99%)

To complement our analysis, we also analysed the performance of our test datasets using the four bacteria-based models made available at the DeepBGC repository:

- `deepbgc`
- `cf_o` (clusterfinder_original)
- `cf_r` (clusterfinder_retrained)
- `cf_g` (clusterfinder_geneborder)

According to the models description at the DeepBGC releases page[9] and [6], the `deepbgc` model is based on the BiLSTM DeepBGC architecture and trained on a MIBiG dataset. The other models are built based on ClusterFinder [12], which is a Hidden Markov Model (HMM). `cf_o` is a ClusterFinder HMM using original parameters; `cf_r` is also a ClusterFinder HMM but trained on a MIBiG dataset; and `cf_g` is a ClusterFinder HMM that switches stages only on gene borders, and trained on a MIBiG dataset.

IV. Results and discussion

We present here statistics and further details on the publicly available fungal BGC datasets proposed in this study. We also present results of validation and test phase obtained with classification models based on fungal BGC datasets and built using DeepBGC. Section IV-A has further information and statistics on the fungal BGC datasets proposed in our work. In Section IV-B we present results obtained at validation of training DeepBGC using the models `pos50`, `pos40`, `pos30`, `pos20`, `pos10`, `pos05`, and `pos01`. In Section IV-C we present results obtained at test phase. For the sake of comparison, we also report the results on test data using BGC classification models provided by DeepBGC and built based on bacteria data, as listed in Section III-C. All performance metrics are reported on the positive class only.

A. Fungal BGC datasets

The fungal BGC datasets proposed in this work are composed of positive and negative instances, as mentioned in Section III-A. These datasets are suitable for performing binary classification to predict fungal BGCs, and are made publicly available at https://github.com/bioinfoUQAM/fungalbgcdata. The availability of such resource can potentially motivate the development of supervised learning approaches to tackle BGC discovery in fungi.

Positive instances in our datasets represent fungal BGCs from 52 different fungal genera. The variety of fungal genus is relevant to provide a large representation of BGC occurrence through different organisms. Additionally, the positive instances contain samples of over 10 different BGC types. Table II shows the different BGC types and a summary of fungal genera in our datasets. As the table shows, the most common BGC type is Polyketide synthase (PKS), followed by Non-ribosomal peptide synthase (NRP) and Terpene synthase (TC). The presence of different fungal genus and BGC types in the datasets are important for representing a wide variety of BGC occurrences, and therefore contribute to building more robust supervised learning approaches.

BGC types	**#**
Alkaloid	3
Alkaloid/NRP	3
Alkaloid/TC	1
Alkaloid/NRP/TC	1
NRP	41
NRP/PKS	19
PKS	90
PKS/TC	5
RiPP	3
Saccharide	1
TC	23
Other	10
Total	**200**

BGC fungi genus	**#**	**BGC fungi genus**	**#**
Acremonium	1	Metacordyceps	1
Alternaria	5	Metarhizium	1
Armillaria	1	Monascus	3
Aspergillus	9	Mycosphaerella	1
Aureobasidium	1	Myrothecium	1
Beauveria	1	Neosartorya	1
Bipolaris	3	Neotyphodium	2
Botrytis	1	Nodulisporium	1
Byssochlamys	1	Paecilomyces	1
Cercospora	1	Parastagonospora	1
Chaetomium	2	Penicillium	13
Cladonia	2	Pestalotiopsis	1
Claviceps	2	Phoma	2
Diaporthe	1	Phomopsis	1
Elsinoe	1	Purpureocillium	1
Epichloe	2	Sarocladium	1
Fusarium	8	Shiraia	1
Glarea	1	Sordaria	1
Glycomyces	1	Sphaceloma	1
Hypholoma	1	Stachybotrys	1
Hypomyces	1	Starmerella	1
Isaria	1	Talaromyces	3
Lasiodiplodia	1	Tapinella	1
Lecanicillium	1	Tolypocladium	2
Leptosphaeria	1	Trichophyton	1
Malbranchea	1	Ustilago	1

TABLE II
FUNGAL GENERA AND BGC TYPES IN POSITIVE INSTANCES OF DATASETS

Negative instances in our datasets represent synthetic gene clusters composed of fungal orthologs. By using fungal orthologs as source for the negative instances, we can generate synthetic gene clusters that depict the genomic profile of fungi. A total of 549 fungal species are present in orthologs composing our negative instances. The main fungal groups to which the orthologs belong to are shown in Table III, according to their taxonomy level. In this table we show the number of species clustered under different taxonomy levels (genus, family, order, or class), and the corresponding total of non-redundant orthologous genes for each group.

The 52 fungal genera in positive instances together with the 549 fungal species in negative instance orthologs contribute to represent the genomic diversity in fungi, and therefore support the development of more robust classification models.

B. Validation performance

Table IV shows validation metrics obtained with fungal BGC datasets. During training phase, all models using fungal BGC datasets had early stopping before completing the total 328 epochs. This could be a sign that the models were overfitting, a possible consequence due to the size of the

[9]https://github.com/Merck/deepbgc/releases

Group	Taxonomy level	# Species	# Genes
Aspergillus	Genus	30	309,629
Cryptococcus	Genus	7	44,028
Exophiala	Genus	7	67,291
Metarhizium	Genus	5	45,563
Penicilium	Genus	21	208,580
Phytophthora	Genus	6	89,378
Hypocreaceae	Family	7	66,815
Pleosporaceae	Family	9	94, 817
Polyporaceae	Family	6	61,584
Saprolegniaceae	Family	6	81,114
Trichocomaceae	Family	6	52,941
Agaricales	Order	25	293,149
Eurotiales	Order	60	608,401
Helotiales	Order	14	162,251
Hypocreales	Order	50	512,282
Mucorales	Order	15	164,081
Polyporales	Order	17	169,368
Sordariales	Order	8	66,549
Agaricomycetes	Class	77	912,187
Eurotiomycetes	Class	103	1,002,099
Microbotryomycetes	Class	9	59,326
Pucciniomycetes	Class	6	64,018
Saccharomycetes	Class	76	390,808
Tremellomycetes	Class	18	121,702
Ustilaginomycetes	Class	9	55,465

TABLE III
MAIN FUNGAL GROUPS PRESENT IN NEGATIVE INSTANCES OF DATASETS

datasets and the imbalanced distribution between the two classes.

The best performing model `pos50` is the one with the most balanced distribution of positive and negative instances. It yield Precision (P) of 0.598, Recall (R) of 0.995, and F-measure (F) of 0.747. Models `pos10`, `pos05`, and `pos01`, the ones with the most imbalanced distributions, had the lowest validation loss but also the lowest P, R and F.

TABLE IV
VALIDATION PERFORMANCE
USING MODELS BUILT ON PROPOSED DATASETS

Model	Epochs	Loss	P	R	F
pos50	91	0.683	0.598	0.995	0.747
pos40	52	0.719	0.407	1	0.578
pos30	108	0.667	0.536	0.743	0.623
pos20	97	0.758	0.230	0.991	0.373
pos10	70	0.389	0	0	0
pos05	73	0.240	0	0	0
pos01	57	0.062	0	0	0

C. Test performance

The test phase show how the models would perform in a real case scenario, when a complete genome is being processed to predict candidate BGC regions. The dataset inputted in the test phase is composed of candidate clusters from the *A. niger* genome sequence, as described in Section III-B. The performance on the test data is presented in two ways: gene metrics and cluster metrics. Gene metrics show P, R, and F for genes that belong to knownBGCs. Cluster metrics show P, R, and F for knownBGCs where a minimum of one cluster gene must be correctly classified for the cluster to be predicted as positive. Tables V and VI show the results on *A. niger* test datasets, with overlaps of respectively 50% and 30%. These results were obtained using classification models built with the fungal BGC datasets described in Section III-A.

TABLE V
PERFORMANCE FOR *A. niger* TEST DATA USING MODELS
BUILT ON FUNGAL BGC DATASETS USING 50% OVERLAP

Model	Gene metrics			Cluster metrics		
	P	R	F	P	R	F
pos50	0.049	1.0	0.094	0.072	0.988	0.134
pos40	0.048	0.962	0.091	0.073	0.988	0.136
pos30	0.044	0.867	0.083	0.073	0.977	0.136
pos20	0.039	0.694	0.074	0.079	0.93	0.146
pos10	0	0	0	0	0	0
pos05	0	0	0	0	0	0
pos01	0	0	0	0	0	0

TABLE VI
PERFORMANCE FOR *A. niger* TEST DATA USING MODELS
BUILT ON FUNGAL BGC DATASETS USING 30% OVERLAP

Model	Gene metrics			Cluster metrics		
	P	R	F	P	R	F
pos50	0.05	1.0	0.096	0.1	0.988	0.182
pos40	0.048	0.951	0.092	0.099	0.953	0.179
pos30	0.045	0.865	0.085	0.1	0.942	0.18
pos20	0.039	0.669	0.073	0.105	0.884	0.188
pos10	0	0	0	0	0	0
pos05	0	0	0	0	0	0
pos01	0	0	0	0	0	0

Results in the test phase show an important decrease in performance compared to the validation phase metrics. However the behaviors observed at the validation step also appear in test. Similarly to the validation phase, the more imbalanced models `pos10, pos05, pos01` did not predict any candidate cluster as positive. This behavior happened with both test datasets of 50% or 30% overlap, and it could indicate that the model is sensitive to an imbalanced distribution of classes.

Also similarly to the validation phase the more balanced models `pos50, pos40, pos30, pos20` tended to predict most of candidate clusters as positives, leading to high recall but very low precision. Table VI shows slightly better performance for P, R, and F compared to table V. This behavior could indicate that using a 30% overlap in the test data is better suited for the task.

Following the results obtained with models based on fungal BGC datasets, we would like to also analyse the performance of DeepBGC models built using bacteria data on *A. niger* test datasets. Tables VII and VIII show the results obtained on *A. niger* data with respectively 50% and 30% overlap using DeepBGC bacteria models.

Among all DeepBGC bacteria models, `deepbgc` performed best at both gene and cluster metrics, either using 30% or 50% overlap, with 0.273 F. The model `cf_o` showed the lowest performance, with 0.138 F. Models `cf_r` and `cf_g` showed in both cases better performance than `cf_o`. The results using DeepBGC trained models yield a

TABLE VII
PERFORMANCE FOR *A. niger* TEST DATA WITH 50% OVERLAP USING MODELS PROVIDED BY DEEPBGC

Model	Gene metrics			Cluster metrics		
	P	R	F	P	R	F
deepbgc	0.074	0.972	0.138	0.114	0.988	0.205
cf_o	0.05	1.0	0.096	0.074	0.988	0.138
cf_r	0.056	0.997	0.106	0.083	0.988	0.153
cf_g	0.06	0.989	0.113	0.09	0.988	0.166

TABLE VIII
PERFORMANCE FOR *A. niger* TEST DATA WITH 30% OVERLAP USING MODELS PROVIDED BY DEEPBGC

Model	Gene metrics			Cluster metrics		
	P	R	F	P	R	F
deepbgc	0.074	0.954	0.138	0.159	0.988	0.273
cf_o	0.051	0.984	0.096	0.103	0.988	0.187
cf_r	0.058	0.994	0.109	0.118	0.988	0.211
cf_g	0.061	0.992	0.116	0.126	0.988	0.223

similar tendency than that of the fungal BGC models: high recall but very low precision.

A loss in performance between validation and test results is evident, either when using fungal BGC based models or DeepBGC bacteria models.

As mentioned in Section III-A, fungal BGCs seem to show larger genomic diversity, which possibly makes it more complex to perform BGC discovery in fungi if compared to bacteria. Therefore, performance is expected to be somehow affected by performing fungal BGC classification using bacteria-based models.

The dataset size at training time could also have had an impact on training `pos50`, `pos40`, `pos30`, `pos20`, `pos10`, `pos05` models, given DeepBGC classification approach. As the authors in [25] explained, the suitability of deep learning approaches varies according to the problem at hand; and in cases when available data is limited conventional approaches could be relevant and more advantageous. As discussed in Section III-A the number of known fungal BGC data previously validated by curators israther limited, which as a consequence will limit the size of fungal BGC datasets. It is possible and worth investigating that different classification methods, apart from a BiLSTM neural network as adopted in DeepBGC, will be better suited for handling fungal BGC discovery.

V. CONCLUSION

NPs are bioactive compounds that play a vital role in the production of a large variety of drugs, and the discovery of novel NPs can potentially benefit human health. Great effort has been put on identifying BGCs that are capable of producing NPs in plants, bacteria and fungi. Identifying BGCs is a challenging task, specially in fungi given the clusters genomic diversity.

Previous work on identifying BGCs in bacteria have resulted in a large variety of approaches and annotated data available. In fungi most previous approaches are based on data-driven methods and present a limited scope, such as covering only certain types of BGCs, or have been developed based on a single species data. The availability of new fungal BGC datasets could potentially motivate the development of new methods to identify BGCs in fungi. One example is supervised learning, a method that have shown to perform well in bacteria data.

In this work, we present new fungal BGC datasets to leverage supervised learning in the fungal BGC discovery task. These datasets are made publicly available at https://github.com/bioinfoUQAM/fungalbgcdata. The availability of such fungal BGC datasets can potentially motivate the development of binary classification approaches to tackle the BGC discovery task. We have shown results obtained on these fungal BGC datasets using a supervised learning approach developed for bacteria BGCs. We also analysed the performance of bacteria-based classification models applied on a fungal genome. The test performance on both fungal-based generated models or bacteria-based models was similar given precision (low) and recall (high) metrics using the same supervised learning method. This points to an opportunity to explore different supervised learning approaches than the one adopted by the DeepBGC system, that might be more suitable to handle fungal BGC datasets.

ACKNOWLEDGMENT

This work was supported by a fellowship of the Natural Sciences and Engineering Research Council of Canada (NSERC) to H.A., and NSERC Discovery Grant to A.B.D. and A.T.

REFERENCES

[1] A. K. Chaudhary, D. Dhakal, and J. K. Sohng, "An insight into the -omics based engineering of streptomycetes for secondary metabolite overproduction," *BioMed Research International*, vol. 2013, 2013.

[2] M. H. Medema and M. A. Fischbach, "Computational approaches to natural product discovery," *Nature Chemical Biology*, vol. 11, no. 9, pp. 639–648, 2015.

[3] A. K. Chavali and S. Y. Rhee, "Bioinformatics tools for the identification of gene clusters that biosynthesize specialized metabolites," *Briefings in Bioinformatics*, vol. 19, pp. 1022–1034, 04 2017.

[4] A. Osbourn, "Secondary metabolic gene clusters: evolutionary toolkits for chemical innovation," *Trends in Genetics*, vol. 26, no. 10, pp. 449–457, 2010.

[5] P. Agrawal, S. Khater, M. Gupta, N. Sain, and D. Mohanty, "RiPP-Miner: a bioinformatics resource for deciphering chemical structures of RiPPs based on prediction of cleavage and cross-links," *Nucleic Acids Research*, vol. 45, no. W1, pp. W80–W88, 2017.

[6] G. D. Hannigan, D. Prihoda, A. Palicka, J. Soukup, O. Klempir, L. Rampula, J. Durcak, M. Wurst, J. Kotowski, D. Chang, *et al.*, "A deep learning genome-mining strategy for biosynthetic gene cluster prediction," *Nucleic Acids Research*, 08 2019.

[7] M. H. Medema, R. Kottmann, P. Yilmaz, M. Cummings, J. B. Biggins, K. Blin, I. De Bruijn, Y. H. Chooi, J. Claesen, R. C. Coates, *et al.*, "Minimum information about a biosynthetic gene cluster," *Nature Chemical Biology*, vol. 11, no. 9, p. 625, 2015.

[8] K. R. Conway and C. N. Boddy, "ClusterMine360: a database of microbial PKS/NRPS biosynthesis," *Nucleic Acids Research*, vol. 41, no. D1, pp. D402–D407, 2012.

[9] K. Blin, T. Wolf, M. G. Chevrette, X. Lu, C. J. Schwalen, S. A. Kautsar, H. G. Suarez Duran, E. L. De Los Santos, H. U. Kim, M. Nave, *et al.*, "antiSMASH 4.0 improvements in chemistry prediction and gene cluster boundary identification," *Nucleic Acids Research*, vol. 45, no. W1, pp. W36–W41, 2017.

[10] K. Blin, M. H. Medema, R. Kottmann, S. Y. Lee, and T. Weber, "The antiSMASH database, a comprehensive database of microbial secondary metabolite biosynthetic gene clusters," *Nucleic Acids Research*, p. gkw960, 2016.
[11] M. Hadjithomas, I.-M. A. Chen, K. Chu, J. Huang, A. Ratner, K. Palaniappan, E. Andersen, V. Markowitz, N. C. Kyrpides, and N. N. Ivanova, "IMG-ABC: new features for bacterial secondary metabolism analysis and targeted biosynthetic gene cluster discovery in thousands of microbial genomes," *Nucleic Acids Research*, vol. 45, no. D1, pp. D560–D565, 2016.
[12] P. Cimermancic, M. H. Medema, J. Claesen, K. Kurita, L. C. W. Brown, K. Mavrommatis, A. Pati, P. A. Godfrey, M. Koehrsen, J. Clardy, *et al.*, "Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters," *Cell*, vol. 158, no. 2, pp. 412–421, 2014.
[13] I. Kjærbølling, T. C. Vesth, J. C. Frisvad, J. L. Nybo, S. Theobald, A. Kuo, P. Bowyer, Y. Matsuda, S. Mondo, E. K. Lyhne, *et al.*, "Linking secondary metabolites to gene clusters through genome sequencing of six diverse Aspergillus species," *Proceedings of the National Academy of Sciences*, vol. 115, no. 4, pp. E753–E761, 2018.
[14] T. C. Vesth, J. Brandl, and M. R. Andersen, "FunGeneClusterS: predicting fungal gene clusters from genome and transcriptome data," *Synthetic and Systems Biotechnology*, vol. 1, no. 2, pp. 122–129, 2016.
[15] M. Umemura, H. Koike, N. Nagano, T. Ishii, J. Kawano, N. Yamane, I. Kozone, K. Horimoto, K. Shin-ya, K. Asai, J. Yu, J. W. Bennett, and M. Machida, "MIDDAS-M: motif-independent de novo detection of secondary metabolite gene clusters through the integration of genome sequencing and transcriptome data," *PLOS ONE*, vol. 8, no. 12, p. e84028, 2013.
[16] T. Wolf, V. Shelest, N. Nath, and E. Shelest, "CASSIS and SMIPS: promoter-based prediction of secondary metabolite gene clusters in eukaryotic genomes," *Bioinformatics*, vol. 32, no. 8, pp. 1138–1143, 2016.
[17] I. Takeda, M. Umemura, H. Koike, K. Asai, and M. Machida, "Motif-independent prediction of a secondary metabolism gene cluster using comparative genomics: application to sequenced genomes of Aspergillus and ten other filamentous fungal species," *DNA Research*, vol. 21, no. 4, pp. 447–457, 2014.
[18] N. Khaldi, F. T. Seifuddin, G. Turner, D. Haft, W. C. Nierman, K. H. Wolfe, and N. D. Fedorova, "SMURF: genomic mapping of fungal secondary metabolite clusters," *Fungal Genetics and Biology*, vol. 47, no. 9, pp. 736–741, 2010.
[19] M. A. Skinnider, C. A. Dejong, P. N. Rees, C. W. Johnston, H. Li, A. L. Webster, M. A. Wyatt, and N. A. Magarvey, "Genomes to natural products prediction informatics for secondary metabolomes (PRISM)," *Nucleic Acids Research*, vol. 43, no. 20, pp. 9645–9662, 2015.
[20] E. V. Kriventseva, D. Kuznetsov, F. Tegenfeldt, M. Manni, R. Dias, F. A. Simão, and E. M. Zdobnov, "OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs," *Nucleic Acids Research*, vol. 47, no. D1, pp. D807–D811, 2018.
[21] T. C. Vesth, J. L. Nybo, S. Theobald, J. C. Frisvad, T. O. Larsen, K. F. Nielsen, J. B. Hoof, J. Brandl, A. Salamov, R. Riley, *et al.*, "Investigation of inter-and intraspecies variation through genome sequencing of Aspergillus section Nigri," *Nature Genetics*, vol. 50, no. 12, p. 1688, 2018.
[22] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, *et al.*, "The Pfam protein families database in 2019," *Nucleic Acids Research*, vol. 47, no. D1, pp. D427–D432, 2018.
[23] R. P. de Vries, R. Riley, A. Wiebenga, G. Aguilar-Osorio, S. Amillis, C. A. Uchima, G. Anderluh, M. Asadollahi, M. Askin, K. Barry, *et al.*, "Comparative genomics reveals high biological diversity and specific adaptations in the industrially and medically important fungal genus Aspergillus," *Genome biology*, vol. 18, no. 1, p. 28, 2017.
[24] D. O. Inglis, J. Binkley, M. S. Skrzypek, M. B. Arnaud, G. C. Cerqueira, P. Shah, F. Wymore, J. R. Wortman, and G. Sherlock, "Comprehensive annotation of secondary metabolite biosynthetic genes and gene clusters of Aspergillus nidulans, A. fumigatus, A. niger and A. oryzae," *BMC Microbiology*, vol. 13, no. 1, p. 91, 2013.
[25] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, "Deep learning for computational biology," *Molecular Systems Biology*, vol. 12, no. 7, p. 878, 2016.

APPENDIX B

Published online 27 November 2020

NAR Genomics and Bioinformatics, 2020, Vol. 2, No. 4 **1**
doi: 10.1093/nargab/lqaa098

TOUCAN: a framework for fungal biosynthetic gene cluster discovery

Hayda Almeida [1,2,3], Sylvester Palys[3], Adrian Tsang[1,3] and Abdoulaye Baniré Diallo [1,2,4,*]

[1]Departement d'Informatique, UQAM, Montréal, QC, H2X 3Y7, Canada, [2]Laboratoire d'Algèbre, de Combinatoire, et d'Informatique Mathématique, Montréal, QC, H2X 3Y7, Canada, [3]Centre for Structural and Functional Genomics, Concordia University, Montréal, QC, H4B 1R6, Canada and [4]Centre of Excellence in Research on Orphan Diseases—Courtois Foundation (CERMO-FC), Montréal, QC, H2X 3Y7, Canada

Received July 24, 2020; Revised September 28, 2020; Editorial Decision October 19, 2020; Accepted November 05, 2020

ABSTRACT

Fungal secondary metabolites (SMs) are an important source of numerous bioactive compounds largely applied in the pharmaceutical industry, as in the production of antibiotics and anticancer medications. The discovery of novel fungal SMs can potentially benefit human health. Identifying biosynthetic gene clusters (BGCs) involved in the biosynthesis of SMs can be a costly and complex task, especially due to the genomic diversity of fungal BGCs. Previous studies on fungal BGC discovery present limited scope and can restrict the discovery of new BGCs. In this work, we introduce TOUCAN, a supervised learning framework for fungal BGC discovery. Unlike previous methods, TOUCAN is capable of predicting BGCs on amino acid sequences, facilitating its use on newly sequenced and not yet curated data. It relies on three main pillars: rigorous selection of datasets by BGC experts; combination of functional, evolutionary and compositional features coupled with outperforming classifiers; and robust post-processing methods. TOUCAN best-performing model yields 0.982 *F*-measure on BGC regions in the *Aspergillus niger* genome. Overall results show that TOUCAN outperforms previous approaches. TOUCAN focuses on fungal BGCs but can be easily adapted to expand its scope to process other species or include new features.

INTRODUCTION

Secondary metabolites (SMs) are specialized bioactive compounds primarily produced by plants, fungi and bacteria. They represent a vital source for drug discovery: from anticancer, antiviral and cholesterol-lowering medications to antibiotics and immunosuppressants (1). Genes involved in the biosynthesis of many SMs in fungi are co-localized in the genome, organized as clusters of genes (2), and known as biosynthetic gene clusters (BGCs). Typically, BGCs are minimally composed of one or more synthase or synthetase genes encoding backbone enzymes, which produce the core structure of the compound, and genes that encode tailoring enzymes, which modify the core compound to generate variants (3). Backbone enzymes determine the class of SM produced by a BGC. BGCs may also contain other genes such as those encoding cluster-specific transcription factors, mitigating toxic properties, transporters, tailoring enzymes and genes with hypothetical functions (4). Identifying new fungal BGCs can potentially lead to the discovery of new compounds that can serve as vital source for drug discovery (5,6). Despite the availability of a large volume of fungal genome sequence data, BGC discovery remains a challenging task (1) due to the diversity of fungal BGCs. Fungal BGCs have been shown to present noticeable differences in synteny and non-conservation of sequences even in related species or different strains of the same species (3), where clustered genes of the same SM can appear in different scaffolds among evolutionarily close species.

Several studies have presented approaches to discover BGCs (1). Most approaches to identify fungal BGCs rely on probabilistic or data-driven methods, requiring as input genomic data (7) combined with gene functional annotations (8) and/or transcription data (9,10). Previous works also analysed fungal gene expression levels (9), motif co-occurrence in promoters around anchor genes (containing backbone enzymes) (8), compared expression levels of virtual gene clusters in conditions favourable to SM production (10) and analysed homologous genes through sequence alignment and filtering syntenic blocks (7). fungiSMASH (11) combines a probabilistic method (profile hidden Markov models from proteins) and curated BGC detection rules, and can use tools such as CASSIS (cluster assignment by islands of sites) (8) and ClusterFinder (12) to predict fungal BGC boundaries. These previous ap-

*To whom correspondence should be addressed. Tel: +1 514 987 3000 (post 3914); Email: diallo.abdoulaye@uqam.ca

proaches present several limitations: overprediction of BGC length (11,13); dependence on manual curation (9), which is expensive; or a very limited scope, potentially affecting the ability to process different BGC types or organisms (7,13).

Approaches derived from supervised learning have shown to perform well when predicting bacterial BGCs (14,15). To our knowledge, such methods have not been applied to identifying fungal BGCs. For instance, RiPPMiner (14) based on support vector machine (SVM) and random forest achieves 0.91 *F*-measure (*F*-m) in binary classification of ribosomally synthesized and post-translationally modified peptides. A recent approach, called DeepBGC, was designed to exploit Pfam (16) domain embeddings to represent bacterial BGCs (12) to feed a bidirectional long short-term memory (BiLSTM) neural network (15). DeepBGC relies also on post-processing methods such as merging consecutive BGC genes or filtering regions without known BGC protein domains. DeepBGC achieved a 0.923 area under the curve when predicting BGC positions in a set of 65 experimentally validated BGCs from six bacterial genomes, outperforming previous studies (15). When handling fungal BGC data, DeepBGC in its original version yielded performance no higher than 0.2 *F*-m (17), and when trained on fungal data underperformed previous methods such as fungiSMASH (11), as we show in the 'Results' section. This could indicate that BGC discovery methods developed for bacteria may not be suitable for fungi due to the high diversity of fungal BGCs that are found to vary even among closely related species (3). Hence, it is important to develop BGC discovery approaches dedicated to fungi, taking into account the specific characteristics of fungal BGCs, such as high diversity, BGC components, and BGC and genome lengths that are usually longer than bacteria. Here, we propose TOUCAN, a supervised learning framework to tackle BGC discovery in fungi that is based on a combination of heterogeneous biological feature types: *k*-mers, protein domains and Gene Ontology (terms) to represent protein motifs and functions relevant to fungal BGCs.

MATERIALS AND METHODS

TOUCAN classification models were built based on a set of six open-access fungal BGC datasets of varying distributions, a total of six classifiers and two post-processing methods. In this section, we present the methodology adopted to develop TOUCAN models. TOUCAN predictions are validated based on a set of curated fungal BGCs.

Datasets

TOUCAN classification models were developed with comprehensive and exhaustive fungal BGC datasets presented in (17) that are publicly available to support benchmarking of BGC discovery methods. The six fungal BGC training datasets are composed of different distributions of positive instances obtained from the MIBiG (Minimum Information about a Biosynthetic Gene cluster) (2) repository and synthetic negative instances generated from OrthoDB (18) orthologues. Fungal orthologous genes were previously applied in BGC discovery (7). Orthologues can be a relevant source of negative instances since they represent conserved genes across species, while BGCs are known to show large genomic diversity even in closely related species (3).

To build negative instances, the amino acid sequences of OrthoDB fungal orthologous genes were concatenated using a fixed window size of 7000 amino acids, which corresponds to the average amino acid length of all positive instances from the fungal subset in MIBiG. This process generated a pool of training samples of 693 195 synthetic negative clusters [see (17) for details]. Studying datasets of various distributions could shed light on the impact of class imbalance in fungal BGC discovery, which by nature presents a highly imbalanced scenario where only a small fraction of fungal genomes actually corresponded to BGCs (17). To account for genomic diversity in fungi, positive instances in the six datasets represent >10 different BGC types and >100 fungal species, while negative instances were generated from a pool of orthologous genes representing ≈300 fungal species. To build and validate our models, we performed a random fixed split in each training dataset for which 80% of instances are dedicated to training and 20% for validation. Supplementary Table S1 shows the positive versus negative distribution, and the training and validation splits in the six training fungal BGC datasets. A random fixed split allows us to evaluate the performances of the same training and validation sets under different parameters.

In the test phase, we evaluated our classification models with six test datasets, generated similarly to (17), from a manually curated genome sequence of *Aspergillus niger* NRRL3, available at https://gb.fungalgenomics.ca/portal. *Aspergillus niger* is an organism of interest for BGC discovery due to its relevance to industrial processes, and its ubiquitous distribution (6). In this work, 85 manually curated BGCs (19) in *A. niger* will be considered as gold standard. Test candidate BGCs are generated by sequentially extracted genomic regions of *A. niger* with a sliding window of 5000, 7000 or 10 000 amino acids, with a 30% or 50% overlap. The overlap of genomic regions allows us to cover BGC fragmented by the sliding windows. Multiple test datasets allow to analyse the impact of window lengths and overlaps when handling input data of test organisms, helping to determine recommended parameters to obtain BGC predictions in new genome sequences. By generating test candidates based on a fixed sliding window length, new sequence data can be processed without requiring curation, genome annotation or gene models as input, unlike that of other BGC discovery tools. In the 'Results' section, we report the performance obtained by the models using different window lengths and overlaps.

Features

To represent the fungal BGC dataset instances as feature vectors, we relied on heterogeneous biological features ex-

tracted from the protein sequences of dataset instances: *k*-mers, Pfam protein domains and GO terms. Several feature types are combined to better represent the diverse genomic profiles in fungal BGCs and help build relevant discriminative models. Feature vectors are composed of number of occurrences of features per training instance. *K*-mers (a contiguous number of *K* amino acids appearing sequentially) are common features in genomic classification tasks (20). We have extracted *k*-mers with varying lengths of $3 \leq K \leq 9$. *K*-mers appearing less than three times were discarded to reduce feature dimensionality, because presence of rare features could introduce bias (21). *K*-mer lengths were evaluated separately using validation sets to identify the *K* value yielding the best performance. Further details on validation of *K* values are provided in the 'Results' section.

Pfam protein domains were previously applied in BGC discovery in both fungi (13) and bacteria (12,15). Protein domains are relevant features for BGC classification and can indicate the presence of backbone enzymes, a key component of BGCs (3,19). We performed an analysis of protein domain distribution among positive instances in our datasets to understand their relevance as features. In our analysis, Pfam protein domains extracted from positive instances were manually labelled by us as *high* (corresponding to a domain usually only present in BGCs) and *medium* (a domain usually present in, but not limited to, BGCs). The complete lists of *medium* and *high* annotated Pfam domains are presented in Supplementary Tables S2 and S3. Then, we analysed all positive instance datasets for the presence or absence of such domains, shown in Supplementary Figure S1. This analysis highlights two important aspects: first, the protein domain diversity in fungal BGCs; and second, the presence of *high* domains shared by most BGCs suggesting that they share a structural pattern, most likely related to the presence of a backbone enzyme. The structural pattern yielded by the distribution of manually annotated protein domains in positive instances suggests that this feature type might carry an important discriminating power. Pfam domain features were extracted from our training datasets using the Pfam database.

GO term annotations were also modelled as features and obtained from our training instances using Swiss-Prot (22). To identify corresponding GO terms, we performed a BLAST analysis of amino acid sequences from our dataset instances against the Swiss-Prot database composed of 560 292 reviewed entries (as of June 2019). BLAST parameters considered were *evalue* (expected value) $\leq$ 1e−4 and *qcovs* (query coverage per subject) $\geq$ 50. A *qcovs* $\geq$ 50 could indicate relevant sequence similarity (23), since the alignment length would correspond to at least 50% of 7000 amino acids for each match. We considered GO terms from all classes. GO term matches found were filtered for duplicates, and only unique GO terms were kept to represent dataset instances.

The number of unique features per type extracted from each training dataset and used to build our classification models is shown in Supplementary Table S4. At this point, extracted features were all kept (except for *k*-mers that occur less than three times in a dataset), without relying on feature selection methods. The feature order is not necessarily conserved during classification, and it is by all purposes processed in a bag-of-words manner. Consider that all extracted features can be relevant at this point since the experiments performed in our work are still a learning space of suitable parameters to tackle BGC discovery. Feature selection could therefore limit the exploration of potentially relevant attributes or combinations of features, but it might be valuable as the next step.

Classification methods

TOUCAN classification models were built with a total of six classifiers. We performed experiments with different classification algorithms to assess the performance of heterogeneous features and post-processing methods, and then identified the best configuration to tackle the BGC discovery task. Three classifiers were SVM classifiers: C support vector (`svc`), linear support vector (`lsvc`) and nu support vector (`nusvc`) classifiers. SVM classifiers were previously applied in BGC discovery (14). Default parameters were used for `svc` and `lsvc` during experiments, while for the `nusvc` classifier the `nu` parameter was adjusted in connection with the percentage of positive instances *pos* in a given dataset:

$$\texttt{nu} = \begin{cases} 0.5, & \text{if } pos \geq 30\%, \\ \dfrac{pos}{100}, & \text{otherwise.} \end{cases} \quad (1)$$

The other three classifiers were a multilayer perceptron (`mlp`), logistic regression (`logit`) and random forest (`randomf`). While `logit` classifier can provide a baseline model for the task, neural networks (15) and `randomf` (14) were also previously applied in BGC discovery. Also for `mlp`, `logit` and `randomf`, default parameters were kept but could, however, be optimized to suit specific experiments if needed. These six classifiers were evaluated independently during our experiments.

Post-processing methods

Predictions of candidate BGCs outputted by TOUCAN are post-processed to improve output precision. Post-processing methods adopted in our work were greedy approaches, such as in PRISM (24) that identifies bond-forming domains and expands cluster boundaries on either ends of such domains. Unlike PRISM, TOUCAN does not require curation as input, and relies on classification models to identify potential BGC regions in which post-processing methods can be applied, facilitating its use on newly sequenced or not yet annotated genomes. The post-processing methods `succ` and `merge` are shown in the SUCCESSIVE-MERGE algorithm, and aim to address potential cluster boundary limitations (over- or underestimation) common in previous approaches (11,13).

Algorithm 1 - SUCCESSIVEMERGE: Compute successive or merged positives

```
begin
    nbSucc ;              // number of successive predictions
    doMerge ;               // boolean, True to perform merge
    threshold ;          // confidence threshold, default 0.5
    merges ;                    // list of merged predictions
    for i ∈ P do
        count = 1
        if i.confidence ≥ threshold then
            merged.ID = i.ID
            while count ≤ nbSucc do
                successor = P[i + count]
                if doMerge then
                    merged.ID += successor.ID
                else if successor.confidence < threshold then
                    successor.confidence = i.confidence
                    P.update(successor)
                count++
            if doMerge then
                merged.confidence = i.confidence
                merges.add(merged)
        else
            merges.add(i)
    if doMerge then
        return merges
    return P
```

BGC region length can vary greatly among fungal MIBiG BGCs: for an x number of amino acids, x can vary such as $195 \leq x \leq 62\,079$, with a standard deviation $\sigma(x) \approx 6013.73$ and mean $\overline{x} \approx 7033$. In this work, a fixed amino acid length to generate test candidate instances from an organism genome is applied. Both `succ` and `merge` post-processing help to overcome the shortcoming in cases where cluster regions have limited boundaries. The `succ` post-processing gives to an *nbSucc* of successive predictions the same *confidence* prediction score of a positive prediction (*confidence* $\geq$ *threshold*). The `merge` post-processing merges an *nbSucc* of successive predictions of a positive prediction (*confidence* $\geq$ *threshold*) into a single positive prediction. For both `succ` and `merge`, we considered $0 \leq$ *nbSucc* ≤ 3, set as an arbitrary parameter for the first evaluation of post-processing methods. Both post-processing methods were applied only if *nbSucc* successive predictions were also not positive.

Evaluation metrics

TOUCAN classification models were assessed in terms of precision (*P*), recall (*R*), *F*-m and a *clusterScore* metric. To compute *P*, *R* and *F*-m, we considered as true positives (TPs) BGC candidates predicted as positive that have at least one gene that matches a gold standard BGC. The *clusterScore* represents the coverage of expected gold standard BGC genes within a candidate BGC, where $0 \leq$ *clusterScore* ≤ 1, and was computed for each BGC candidate predicted positive. To compute the *clusterScore* for a BGC candidate C and its gold standard BGC match M, we first counted the number of *geneMatches* in C, meaning the number of M genes in C. We then computed a similarity value `sim` between all pairs of genes in the disjunctive union $C \triangle M$, and add to the *clusterScore* the best `sim` obtained for the unmatched $M - C$ genes. Computing the `sim` value allows us to account for the possible presence of gold standard orthologues among unmatched genes in a BGC candidate predicted positive. The `sim` value represents a percent identity *pident* obtained through a local alignment with BLAST between two genes, using cut-offs of minimum *pident* ≥ 20 and minimum query coverage *qcovs* ≥ 10. The *clusterScore* for a BGC candidate C was normalized by the number of genes in its gold standard BGC match M. The SIMILARITY algorithm shows the computation of `sim` scores, while the CLUSTERSCORE algorithm shows the computation of *clusterScore*. We analysed the *clusterScore* of TOUCAN predicted positives compared to state-of-the-art methods in the 'Results' section.

State-of-the-art performance comparison

The performance of TOUCAN models was compared to results obtained by two state-of-the-art tools: fungiSMASH (11) and DeepBGC (15) (version 0.1.18 and models as of February 2020 available at https://github.com/Merck/deepbgc). The experiments with fungiSMASM were performed with its three strictness levels: relaxed, strict and loose, and with default parameters for its extra feature options (as of January 2020): 'KnownClusterBlast', 'ActiveSiteFinder' and 'SubClusterBlast'. DeepBGC focuses on bacterial data and is based on a BiLSTM neural network and Pfam domain embeddings. A total of three DeepBGC classification models are applied in this work: one with original DeepBGC training dataset and hyperparameters, as in (15); one built with DeepBGC original hyperparameters and our best-performing training dataset; and one built with our best-performing training dataset and fungal optimized hyperparameters (thanks to the authors) (see Supplementary Table S7 for original and fungal optimized hyperparameters).

Algorithm 2 - SIMILARITY: Compute similarity score of genes

```
Data: pred.genes, genes in a positive predicted cluster
      gold.genes, genes in a gold cluster
      genesFound, predicted genes matching gold genes
      similarities, list of similarities between genes
Result: similarityScore, similarity score
begin
    pairs = Ø ;              // best scores for pairs of genes
    pairedGenes = Ø ;     // predicted and gold genes paired
    pred.genes = pred.genes − genesFound
    gold.genes = gold.genes − genesFound
    for i ∈ gold.genes do
        totalScore = 0
        for j ∈ pred.genes do
            score = similarities.get(i, j)
            if score ≥ totalScore then
                pair.gene1 = i
                pair.gene2 = j
                pair.score = score
                pairs.add (pair)
                totalScore = score
    // Sort pairs by score in descending order
    pairs = sortDescScore(pairs)
    for pair ∈ pairs do
        if pair.gene2 not ∈ pairedGenes then
            similarityScore += pair.score
            pairedGenes.add (pair.gene2)
    return similarityScore
```

Algorithm 3 - CLUSTERSCORE: Compute *clusterScore* evaluation metric

Data: P, list of prediction tuples $(geneID, clusterID, confidenceVal)$
G, list of gold tuples $(goldGeneID, goldClusterID)$
`useSimilarity`, boolean value for considering similarity scores
Result: $clusterScores$ for a list of predictions.
begin

```
threshold ;          // confidence threshold, default 0.5
geneMatches ;                // count of gene matches
geneMatches* ;          // count gene matches with
                                  // similarity scores
clustersFound ;        // list of gold clusters found
// Compute successive or merged positives
P = computeSuccOrMerge(P)
// Retrieve list of only positive predictions
posGenes,posClusters = unfoldPredictions(P)
for gold ∈ G do
    for gene ∈ gold.genes do
        // Find occurrences of gene in predictions
        pred = getAllOccurrences(gene,P)
        if pred ∈ posGenes then
            geneMatches += 1
            clustersFound.add(pred)
    if gold.confidence ≥ threshold & useSimilarity then
        geneMatches* = geneMatches +
          similarity(gold.genes,pred.genes,genesFound)
    clusterScore = geneMatches* / length(gold.genes)
    clusterScores.add(gold.ID,clusterScore)
return clusterScores
```

RESULTS

We present here results obtained with TOUCAN, a supervised learning framework to discover fungal BGCs. To identify the best configuration to tackle BGC discovery in fungi, we designed, trained and assessed several classification models combining heterogeneous biological features, datasets of various distributions, classifiers and post-processing methods, as described in the 'Materials and Methods' section. Validation results are drawn on held-out training instances corresponding to 20% of each training dataset. The performance of TOUCAN was assessed on test datasets of a gold standard of 85 manually annotated *A. niger* BGCs (19). The focus here is BGC discovery; hence, the model is optimized to correctly identify positive instances, rather than the negative ones. Thus, results were reported for the positive class.

Feature importance and performance on validation datasets

To identify the most suitable K for k-mer features within $3 \leq K \leq 9$, we performed a set of experiments on all six datasets and six classifiers, as presented in the 'Datasets' and 'Classification methods' sections. Performance of k-mer models on our validation sets is shown in Supplementary Figure S2. In general, better performance was achieved with $K = 6$, which was thus the K value considered for our following experiments. We also performed an analysis of feature importance across training datasets, obtained with a `randomf` classifier, with default parameters. Table 1 shows the top 15 ranked features across training datasets.

Features appearing on the top 15 of multiple datasets are highlighted. Protein domain feature names start with *PF*, GO term feature names start with *GO* and the other features are 6-mers. We can observe that every protein domain feature appearing among the top ranked of all datasets belonged to either the *high* or *medium* manually annotated domains, even though non-*high* and non-*medium* domain features are also included in our feature set. Moreover, while GO terms represent ≈30% of all top 15 ranked features, they make up for at most 0.7% of total features. This possibly indicates their strong discriminating power in the task. After evaluating feature importance, we trained several classification models combining the feature types for each classifier and training dataset distribution. For each training dataset distribution, a random fixed split, designating 80% of its instances, was selected for training and 20% for validation, as mentioned in the 'Datasets' section. The top *F*-m performances on validation sets per training dataset are shown in Supplementary Table S5. During validation, we noted that models built with three feature types outperformed models using one feature type, such as the ones built when evaluating the most suitable *k*-mer length.

Validation performance seems to be overall affected by the instance distribution: more imbalanced datasets show lower *F*-m compared to more balanced ones. When analysing MIBiG fungal BGCs, only ≈1% of a genome sequence would correspond to cluster regions (17), so utilizing more balanced training data could provide better performance than using real case scenario distributions. We selected the dataset with the best *F*-m average performance, which was the most balanced (50%–50%), to perform further evaluation with hyperparameter optimization through a grid search, followed by cross-validation (CV) classification for all six classifiers. Best-performing hyperparameters to maximize *F*-m for each classifier are listed in Supplementary Table S8. A 5-fold CV was performed with optimized hyperparameters on the 50%–50% dataset instances, randomly split between training and validation at each fold. Supplementary Table S6 shows the average performances on the 5-fold CV for each classifier.

TOUCAN performance on test datasets

We assessed TOUCAN models on six test datasets with amino acid sliding window lengths of 5000, 7000 and 10 000, with overlaps of 50% and 30%, as described in the 'Datasets' section. Candidate BGC predictions on the test data were obtained with TOUCAN classification models built using the six training dataset distributions with fixed training and validation splits, three feature types and six classifiers. We then processed TOUCAN predicted candidate BGCs with post-processing methods `succ` and `merge`, considering $0 \leq nbSucc \leq 3$.

Table 2 shows for the positive class the best *F*-m obtained for each test dataset among all training dataset distributions. The highest 0.931 *F*-m was obtained by a model built with a 50%–50% distributed training set, an `mlp` classifier and a `merge3` post-processing. The best *F*-m was achieved with 10 000-amino acid sliding window test datasets. Regarding classifiers, `mlp` and `logit` yielded best performance followed less often by `lsvc`. As mentioned in the 'Classification methods' section, default parameters were used when performing our experiments. Tuning the clas-

Table 1. Top 15 features ranked by importance for each training dataset, from completely balanced (50% positive, 50% negative) to most imbalanced (5% positive, 95% negative)

Training dataset distribution					
50%–50%	40%–60%	30%–70%	20%–80%	10%–90%	5%–95%
PF00698.21	**PF00698.21**	GO:0008168	**HGTGTQ**	PF00109.26	**TACSSS**
PF00668.20	**HGTGTQ**	**HGTGTQ**	GO:0008152	GO:0044550	GTGTQA
ADGYCR	**GO:0031177**	**GQGAQW**	PF00550.25	LYRTGD	GYARGE
GO:0016491	**GAGTGG**	GYCRAD	IDTACS	**VFTGQG**	**GO:0046148**
FDGYRF	VEMHGT	**GAGTGG**	PF02458.15	NFSAAG	TGDLAR
GO:0016740	**VFTGQG**	QQRLLL	DTACSS	VEAHGT	SINSFG
MHGTGT	**PF00668.20**	**TACSSS**	VTLSGD	GO:0043041	DPQQRL
DTACSS	GO:0016874	PF02801.22	FTGQGA	GHSLGE	LFTSGS
GO:1900557	YKTGDL	**GO:0009058**	PF08242.12	AYEALE	NSFGFG
GO:0009058	GO:0019184	**GO:0046148**	**AYGPTE**	**GO:0016491**	CDTAVA
GRFFAA	GO:0043042	GO:0047462	GO:0004315	TQVKIR	FDASFF
PF14765.6	PGRFFA	GEYAAL	**GO:0031177**	GO:0046500	**AYGPTE**
MDPQQR	**MHGTGT**	GO:0005829	KLRGFR	**DTACSS**	YILFTS
FTSGST	GO:1900790	PFAFHS	GO:0016021	GO:0032259	AIVLAG
GQGAQW	VEIGPH	LHSLEA	PF00067.22	DTFVRC	AVVGHS

Highlighted features appeared in multiple datasets.

sifier parameters may affect the performance, but this is not the focus of this study. Overall results showed that a 30% overlap seems to be more advantageous for all sliding window lengths, even though the best *F*-m was achieved with test candidates generated based on a 50% overlap. The training set distribution seemed to have little influence on test candidates with a sliding window length of 5000 amino acids, showing only a small variation on *F*-m for both 30% and 50% overlap. Less balanced training distribution seemed to affect performance more for candidates with a sliding window length of 10 000 amino acids, with an *F*-m varying from 0.618 to 0.931 when using 50% overlap, and from 0.629 to 0.917 when using 30% overlap.

We selected the best-performing test datasets (10 000-amino acid sliding window) to carry an evaluation using 5-fold CV classification models based on the best-performing training set (50%–50%). The predicted BGC candidates obtained with CV classification models were also processed with TOUCAN post-processing methods `succ` and `merge`, in the same manner as the models presented in Table 2. The best performance results obtained with the 10 000-amino acid sliding window test data among all 5-fold CV classification models are shown in Table 3.

As shown in Table 3, the 5-fold CV classification models improved to a 0.982 *F*-m from the previously best 0.931 *F*-m achieved with models based on fixed training and validation splits. Performance results in Tables 2 and 3 show TOUCAN models' discriminative power to identify candidate BGC regions from non-BGC regions. Our results also demonstrate TOUCAN models' capacity of obtaining relevant BGC predictions on new or non-annotated genomes in test dataset instances generated solely based on sliding windows of fixed amino acid length. This aspect distinguishes TOUCAN from previous approaches that rely on gene models and other genomic annotations as input (14,15).

Performance comparison with DeepBGC

We compared the performance of three DeepBGC classification models using the 10 000-amino acid sliding window test datasets, which yield the best *F*-m with TOUCAN. As mentioned in the 'Materials and Methods' section, two out of the three DeepBGC models were trained using the best-performing constructed training dataset (50%–50% dataset in this case). The DeepBGC hyperparameters applied in this comparison are also listed in the 'Materials and Methods' section. As shown in (17), during validation phase the DeepBGC model trained using the original hyperparameters and the 50%–50% training dataset had early stopping at epoch 109, from the original total of 328 epochs, as applied in (15).

Table 4 shows *P*, *R* and *F*-m performances of the three DeepBGC models for the positive class on the test dataset with a 50% or 30% overlap. DeepBGC models built with original hyperparameters yielded high recall but very low precision, consequently leading to *F*-m metrics <0.3 for either models based on the 50%–50% training set or models based on DeepBGC original data. Models built with fungal optimized hyperparameters yielded a noticeable performance improvement, with a 0.627 *F*-m.

For each of the three DeepBGC models, the test sets using a 30% overlap resulted in better performance than the ones using a 50% overlap. DeepBGC performance on predicting fungal BGCs shows high recall but very low precision, which consequently lead to *F*-m metrics <0.2. The most imbalanced models classified all test candidates as negative, which could be a sign of the model trying to optimize accuracy towards the majority class. Originally, DeepBGC was developed to predict bacterial BGCs, for which much more data are available compared to fungal BGCs. The larger amount of bacterial BGC data available benefits the development of supervised learning approaches. Fungal BGC data are more scarce, which makes it challenging to build robust classification models. Supervised learning approaches that fit bacteria may not be suitable to discover BGCs in fungi (17).

Performance comparison with fungiSMASH

We compared the performance of fungiSMASH on the same 10 000-amino acid sliding window test datasets used

Table 2. TOUCAN best-performing models per test set sliding windows and overlaps in *A. niger*

Sliding window	Overlap	Training set	Classifier	Post-process	*P*	*R*	*F*-m
10 000	50%	50%–50%	mlp	merge3	1	0.871	**0.931**
10 000	50%	40%–60%	mlp	merge3	1	0.753	0.859
10 000	50%	30%–70%	mlp	merge2	1	0.706	0.828
10 000	50%	20%–80%	mlp	merge2	1	0.706	0.828
10 000	50%	10%–90%	mlp	merge3	1	0.647	0.786
10 000	50%	5%–95%	mlp	merge3	1	0.447	0.618
7000	50%	50%–50%	logit	merge3	0.929	0.765	0.839
7000	50%	40%–60%	logit	merge3	1	0.741	0.851
7000	50%	30%–70%	mlp	merge3	0.969	0.729	0.832
7000	50%	20%–80%	mlp	merge3	1	0.741	0.851
7000	50%	10%–90%	mlp	merge3	1	0.694	0.819
7000	50%	5%–95%	mlp	merge3	1	0.647	0.786
5000	50%	50%–50%	logit	merge3	0.817	0.788	0.802
5000	50%	40%–60%	logit	merge3	0.914	0.753	0.826
5000	50%	30%–70%	logit	merge3	0.953	0.718	0.819
5000	50%	20%–80%	logit	merge3	1	0.718	0.836
5000	50%	10%–90%	mlp	merge3	0.913	0.741	0.818
5000	50%	5%–95%	mlp	merge3	0.923	0.706	0.800
10 000	30%	50%–50%	mlp	merge3	1	0.847	0.917
10 000	30%	40%–60%	mlp	merge3	1	0.741	0.851
10 000	30%	30%–70%	mlp	merge2	1	0.694	0.819
10 000	30%	20%–80%	mlp	merge2	1	0.671	0.803
10 000	30%	10%–90%	mlp	merge3	1	0.6	0.750
10 000	30%	5%–95%	mlp	merge3	1	0.459	0.629
7000	30%	50%–50%	mlp	merge3	0.95	0.906	0.928
7000	30%	40%–60%	mlp	merge3	1	0.824	0.903
7000	30%	30%–70%	mlp	merge2	1	0.741	0.851
7000	30%	20%–80%	mlp	merge3	1	0.741	0.851
7000	30%	10%–90%	lsvc	merge3	1	0.553	0.712
7000	30%	5%–95%	mlp	merge3	1	0.635	0.777
5000	30%	50%–50%	logit	merge3	0.908	0.812	0.857
5000	30%	40%–60%	logit	merge3	0.985	0.788	0.876
5000	30%	30%–70%	logit	merge3	1	0.753	0.859
5000	30%	20%–80%	mlp	merge3	0.985	0.776	0.868
5000	30%	10%–90%	mlp	merge3	0.984	0.729	0.838
5000	30%	5%–95%	mlp	merge3	1	0.706	0.828

Table 3. TOUCAN best performances for the completely balanced (50% positive, 50% negative) CV models on *A. niger* test sets generated with a 10 000-amino acid sliding window

Training set	Sliding window	Overlap	Classifier	Post-process	*P*	*R*	*F*-m
50%–50%	10 000	50%	svc	merge3	0.941	0.941	0.941
50%–50%	10 000	30%	randomf	merge3	1	0.965	**0.982**

Table 4. Performance metrics of DeepBGC models for *A. niger* test sets generated with 10 000-amino acid sliding window

Training dataset	DeepBGC model	Sliding window	Overlap	*P*	*R*	*F*-m
DeepBGC	Original	10 000	50%	0.114	1	0.205
DeepBGC	Original	10 000	30%	0.159	1	0.274
50%–50%	Original	10 000	50%	0.075	1	0.140
50%–50%	Original	10 000	30%	0.105	1	0.191
50%–50%	Fungal	10 000	50%	0.464	0.765	0.578
50%–50%	Fungal	10 000	30%	0.580	0.682	**0.627**

to compare with DeepBGC. The fungiSMASH parameters considered in this comparison are described in the 'Materials and Methods' section. fungiSMASH predictions are also assessed in terms of *P*, *R* and *F*-m, which are shown for the positive class in Table 5. fungiSMASH best performance yielded a 0.571 *F*-m when using a 50% overlap and 0.692 *F*-m when using a 30% overlap, both under relaxed strictness. As expected, loose strictness results in higher recall and lower precision, while a strict parameter results in higher precision but lower recall.

Similar to TOUCAN models, fungiSMASH seems to yield generally better performance on 30% overlap test candidates. fungiSMASH showed in general a more stable performance predicting fungal BGCs compared to DeepBGC. Apart from being based on a different approach than DeepBGC, fungiSMASH was developed focusing on fun-

Table 5. Performance metrics of fungiSMASH models for *A. niger* test sets generated with 10 000-amino acid sliding window

fungiSMASH strictness	Sliding window	Overlap	*P*	*R*	*F*-m	Overlap	*P*	*R*	*F*-m
Relaxed (default)	10 000	50%	0.470	0.729	0.571	30%	0.649	0.741	**0.692**
Strict	10 000	50%	0.471	0.576	0.519	30%	0.671	0.600	0.634
Loose	10 000	50%	0.435	0.788	0.561	30%	0.591	0.800	0.68

gal organisms. The difference in performance between the bacteria-focused approach of DeepBGC and the fungal-focused approach of fungiSMASH may be another indication that BGC discovery is a complex task, and can benefit from approaches built to target related organisms.

TOUCAN yields reproducible performance on *Aspergillus nidulans*

To assess TOUCAN reproducibility, we assessed the performance of its models in the *A. nidulans* genome. As in *A. niger*, *A. nidulans* is a species known as an important source of BGCs (3,19). Previous work on manual annotation of BGCs in Aspergilli (19) identified a total of 70 BGCs in *A. nidulans*, which are considered as gold standard for this analysis. To obtain candidate BGCs for testing, *A. nidulans* genome sequence was processed in the same manner as *A. niger*. Test candidate BGCs for *A. nidulans* were obtained by extracting genomic regions sequentially from its genome, using amino acid sliding windows of 10 000 amino acids that overlap by 30% and 50%. The analysis on *A. nidulans* used the best-performing model parameters previously established in *A. niger*: 50%–50% dataset, hyperparameter optimization and 5-fold CV.

Table 6 shows TOUCAN best performance results among all six classifiers and post-processing methods for the *A. nidulans* 10 000-amino acid sliding window test sets. For comparison, we evaluated *A. nidulans* BGC predictions obtained with the best fungiSMASH and DeepBGC models on the same test sets, for which the results are also shown in Table 6. We observed that similar *F*-m performance metrics were achieved for *A. nidulans* and *A. niger*. TOUCAN and DeepBGC, both based on supervised learning, yielded the least *F*-m variation on the results obtained for the two *Aspergillus* species, suggesting that due to their generalization ability, supervised learning approaches may be a suitable approach to tackle BGC discovery.

TOUCAN TP predictions improve coverage of BGC genes

We compared TP predictions (BGC candidate predicted positives that have at least one gene matching a gold standard BGC) obtained from best-performing models in *A. niger* and *A. nidulans* for TOUCAN (0.982 *F*-m and 0.910 *F*-m, respectively) versus fungiSMASH (0.692 *F*-m and 0.780 *F*-m, respectively) and DeepBGC (0.620 *F*-m and 0.607 *F*-m, respectively). First, we analysed the distribution of *clusterScore* computed for each BGC candidate predicted positive. Figure 1 shows the *clusterScore* distribution in *A. niger* and *A. nidulans* TP predictions obtained with TOUCAN, DeepBGC and fungiSMASH best models.

We observed that compared to the other tools, TOUCAN TP predictions more often present a *clusterScore* = 1, meaning that TOUCAN predictions better encompass genes matching gold standard BGCs, possibly as a result of TOUCAN `merge` post-processing. Although `merge` post-processing leads to more comprehensive predictions, it could result in overprediction of cluster boundaries. To mitigate, filtering methods could be applied to refine candidate cluster regions, and also as an opportunity to fine-tune TOUCAN predictions to specific genus or species of interest. One possible way to apply targeted filtering is to rely on manually curated annotations of relevant features, such as the annotated *high* and *medium* Pfam protein domains shown in the 'Features' section.

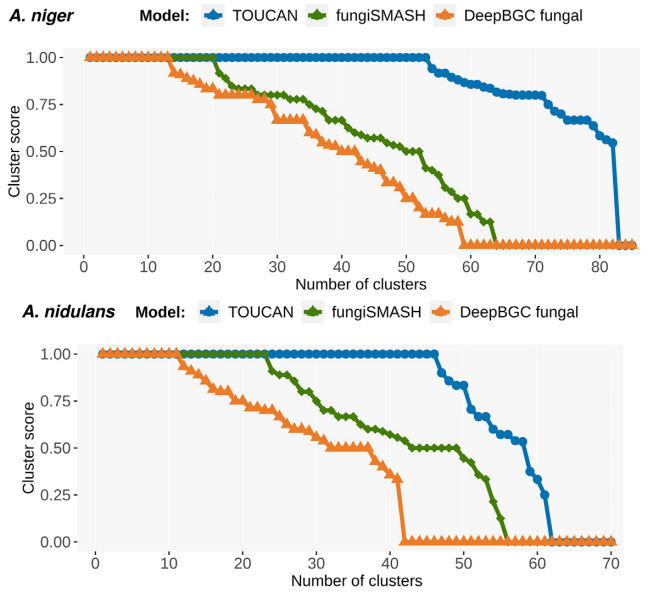


Figure 1. Distribution of *clusterScore* among TP predictions in *A. niger* and *A. nidulans* genomes. *clusterScore* distribution was computed for best-performing models of each system (*A. niger*: TOUCAN: 0.982 *F*-m, DeepBGC: 0.627 *F*-m, fungiSMASH: 0.692 *F*-m; *A. nidulans*: TOUCAN: 0.910 *F*-m, DeepBGC: 0.607 *F*-m, fungiSMASH: 0.780 *F*-m).

We also analysed the presence of backbone enzymes within genes of TP predictions. Backbone enzymes are considered as the BGC core (3), playing a key role in its biosynthesis and defining the BGC compound to be produced (19). We mapped the presence and absence of backbone genes among TOUCAN, DeepBGC and fungiSMASH best models' TP predictions. Figure 2 shows backbone genes and product types found in *A. niger* and *A. nidulans*, respectively. Scores in Figure 2 (or the colour intensity) correspond to the *clusterScore* computed for the predicted BGC. Backbone enzyme genes were present in 86.6% of all TOUCAN TP predictions for *A. niger*, versus 76.2% in fungiSMASH and 75.9% in DeepBGC. For *A. nidulans*, 93.5% of TOU-

Table 6. Best performances per overlap of TOUCAN compared to fungiSMASH and DeepBGC for *A. nidulans* test sets generated with 10 000-amino acid sliding window

System	Model	Sliding window	Overlap	*P*	*R*	*F*-m
TOUCAN	`lsvc` + merge3	10 000	50%	0.919	0.814	0.864
TOUCAN	`svc` + merge3	10 000	30%	0.953	0.871	**0.910**
fungiSMASH	Relaxed (default)	10 000	50%	0.550	0.786	0.647
fungiSMASH	Relaxed (default)	10 000	30%	0.775	0.786	**0.780**
DeepBGC	50%–50% fungal	10 000	50%	0.473	0.629	0.540
DeepBGC	50%–50% fungal	10 000	30%	0.631	0.586	**0.607**

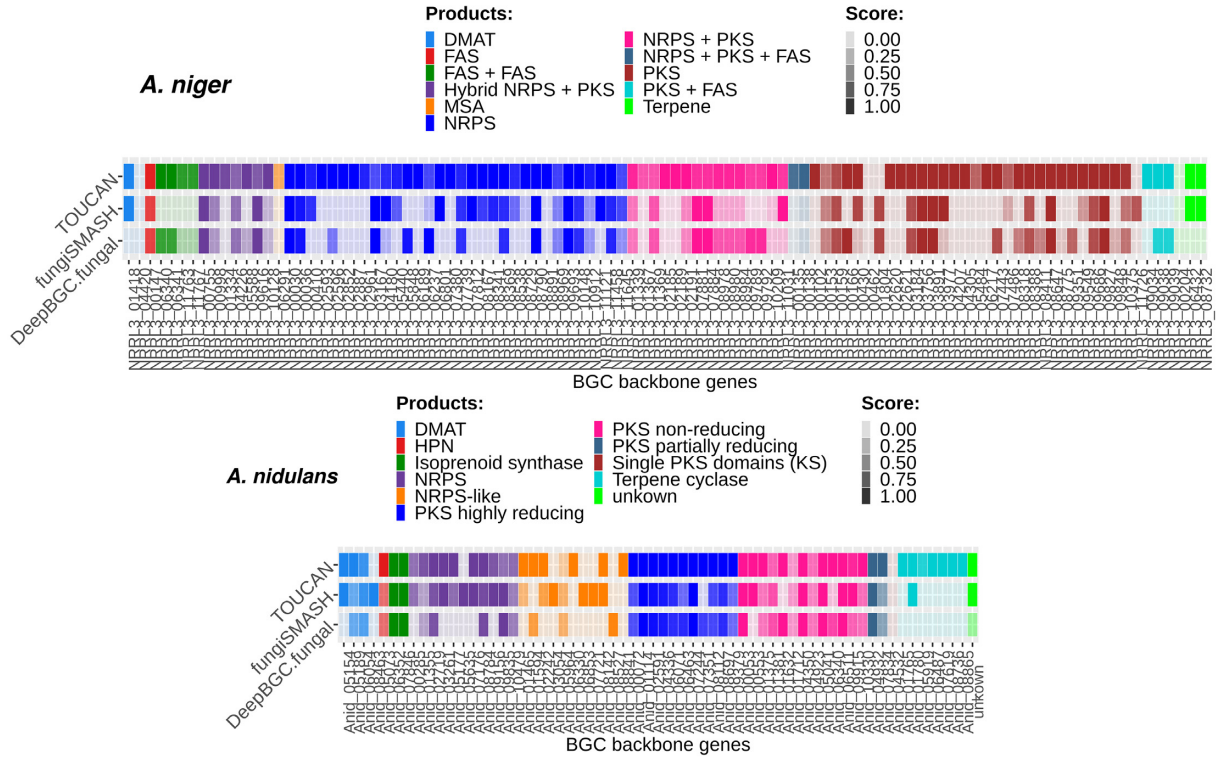


Figure 2. Presence of backbone enzymes among positive predictions in *A. niger* and *A. nidulans* genomes. Each backbone enzyme is shown per the gene ID it is associated with and the *clusterScore* assigned to the candidate predicted BGC.

CAN TP predictions found backbone enzymes, versus 89% for fungiSMASH and 82.9% for DeepBGC.

DISCUSSION

SMs are bioactive compounds that play a vital role in the production of various drugs. Discovery of novel fungal BGCs can potentially benefit human health. In this work, we presented TOUCAN, a supervised learning framework for fungal BGC discovery. We evaluated classification models based on fungal BGC datasets of various distributions, six classifiers, heterogeneous biological features and three post-processing methods. TOUCAN best-performing model achieved 0.982 *F*-m in *A. niger* and 0.910 *F*-m in *A. nidulans*, outperforming previous methods. The results obtained with TOUCAN models could indicate that standard supervised learning approaches are suitable to tackle BGC discovery. TOUCAN outperformance is possibly due to a combination of factors: combining feature types, evaluating the impact of different class distributions during training and post-processing candidate BGC predictions. `merge` post-processing can help identify regions that might have been missed, but in certain cases it may potentially lead to overestimation of predicted cluster boundaries.

The performance of TOUCAN models was compared to two BGC discovery state-of-the-art approaches: DeepBGC, based on deep learning, and fungiSMASH, based on probabilistic methods. TOUCAN models showed better *F*-m when predicting BGCs in *A. niger* and *A. nidulans* compared to DeepBGC and fungiSMASH. TOUCAN also yielded more comprehensive coverage of gold standard BGC genes within predicted clusters, and was able to identify backbone enzyme genes more often in its TP predictions compared to the other methods. The presence of backbone enzymes can be a crucial aspect in determining the presence

of a BGC in a given genomic region. The results obtained by TOUCAN, as well as the performance of DeepBGC models, demonstrate the potential of exploring supervised learning approaches for BGC discovery, and relevance of developing BGC prediction tools focused on fungal organisms. Fungi were shown to be an important source for bioactive compounds (5,6) used in the pharmaceutical industry, but fungal BGC data available in open-access databases are scarce compared to bacteria. The availability of more annotated fungal BGCs is hence an important aspect to promote development and improvement of existing and new fungal BGC discovery approaches. Previous BGC discovery tools require curated data to identify candidate BGC regions in an organism, which may not be available or is expensive to obtain. Unlike previous approaches, TOUCAN is capable of outputting BGC predictions from amino acid sequences without requiring further data curation as input. This aspect can facilitate TOUCAN usage and its application on newly sequenced genomes, promoting the discovery of novel candidate BGC regions and potentially novel drugs, such as antibiotics, immunosuppressants and anticancer medications.

DATA AVAILABILITY

TOUCAN source code as well as all datasets applied in our experiments are made publicly available at http://github.com/bioinfoUQAM/TOUCAN. TOUCAN source code is available under the MIT permissive software license. The datasets used in this work were obtained from open-access databases, which are available under the Creative Commons Attribution 4.0 international license.

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

ACKNOWLEDGEMENTS

We acknowledge the bioinformatics teams at CSFG and UQAM for many inspiring discussions and comments on this manuscript; Amine Remita for helpful discussions and for providing comments on an earlier version of this manuscript; Diego Maupomé for helping with deep learning frameworks; and Antoine Briand for helping with web frameworks.

FUNDING

Natural Sciences and Engineering Research Council of Canada (NSERC); Fonds de Recherche du Québec—Nature et Technologies (FRQNT).
Conflict of interest statement. None declared.

REFERENCES

1. Chavali,A.K. and Rhee,S.Y. (2017) Bioinformatics tools for the identification of gene clusters that biosynthesize specialized metabolites. *Brief. Bioinform.*, **19**, 1022–1034.
2. Kautsar,S.A., Blin,K., Shaw,S., Navarro-Muñoz,J.C., Terlouw,B.R., van der Hooft,J.J., Van Santen,J.A., Tracanna,V., Suarez Duran,H.G., Pascal Andreu,V. *et al.* (2020) MIBiG 2.0: a repository for biosynthetic gene clusters of known function. *Nucleic Acids Res.*, **48**, D454–D458.
3. Kjærbølling,I., Vesth,T., Frisvad,J.C., Nybo,J.L., Theobald,S., Kildgaard,S., Petersen,T.I., Kuo,A., Sato,A., Lyhne,E.K. *et al.* (2020) A comparative genomics study of 23 *Aspergillus* species from section *Flavi*. *Nat. Commun.*, **11**, 1106.
4. Keller,N.P. (2019) Fungal secondary metabolism: regulation, function and drug discovery. *Nat. Rev. Microbiol.*, **17**, 167–180.
5. Macheleidt,J., Mattern,D.J., Fischer,J., Netzker,T., Weber,J., Schroeckh,V., Valiante,V. and Brakhage,A.A. (2016) Regulation and role of fungal secondary metabolites. *Annu. Rev. Genet.*, **50**, 371–392.
6. de Vries,R.P., Riley,R., Wiebenga,A., Aguilar-Osorio,G., Amillis,S., Uchima,C.A., Anderluh,G., Asadollahi,M., Askin,M., Barry,K. *et al.* (2017) Comparative genomics reveals high biological diversity and specific adaptations in the industrially and medically important fungal genus *Aspergillus*. *Genome Biol.*, **18**, 28.
7. Takeda,I., Umemura,M., Koike,H., Asai,K. and Machida,M. (2014) Motif-independent prediction of a secondary metabolism gene cluster using comparative genomics: application to sequenced genomes of *Aspergillus* and ten other filamentous fungal species. *DNA Res.*, **21**, 447–457.
8. Wolf,T., Shelest,V., Nath,N. and Shelest,E. (2016) CASSIS and SMIPS: promoter-based prediction of secondary metabolite gene clusters in eukaryotic genomes. *Bioinformatics*, **32**, 1138–1143.
9. Vesth,T.C., Brandl,J. and Andersen,M.R. (2016) FunGeneClusterS: predicting fungal gene clusters from genome and transcriptome data. *Synth. Syst. Biotechnol.*, **1**, 122–129.
10. Umemura,M., Koike,H., Nagano,N., Ishii,T., Kawano,J., Yamane,N., Kozone,I., Horimoto,K., Shin-ya,K., Asai,K. *et al.* (2013) MIDDAS-M: motif-independent *de novo* detection of secondary metabolite gene clusters through the integration of genome sequencing and transcriptome data. *PLoS One*, **8**, e84028.
11. Blin,K., Wolf,T., Chevrette,M.G., Lu,X., Schwalen,C.J., Kautsar,S.A., Suarez Duran,H.G., De Los Santos,E.L., Kim,H.U., Nave,M. *et al.* (2017) antiSMASH 4.0—improvements in chemistry prediction and gene cluster boundary identification. *Nucleic Acids Res.*, **45**, W36–W41.
12. Cimermancic,P., Medema,M.H., Claesen,J., Kurita,K., Brown,L. C.W., Mavrommatis,K., Pati,A., Godfrey,P.A., Koehrsen,M., Clardy,J. *et al.* (2014) Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. *Cell*, **158**, 412–421.
13. Khaldi,N., Seifuddin,F.T., Turner,G., Haft,D., Nierman,W.C., Wolfe,K.H. and Fedorova,N.D. (2010) SMURF: genomic mapping of fungal secondary metabolite clusters. *Fungal Genet. Biol.*, **47**, 736–741.
14. Agrawal,P., Khater,S., Gupta,M., Sain,N. and Mohanty,D. (2017) RiPPMiner: a bioinformatics resource for deciphering chemical structures of RiPPs based on prediction of cleavage and cross-links. *Nucleic Acids Res.*, **45**, W80–W88.
15. Hannigan,G.D., Prihoda,D., Palicka,A., Soukup,J., Klempir,O., Rampula,L., Durcak,J., Wurst,M., Kotowski,J., Chang,D. *et al.* (2019) A deep learning genome-mining strategy for biosynthetic gene cluster prediction. *Nucleic Acids Res.*, **47**, e110.
16. El-Gebali,S., Mistry,J., Bateman,A., Eddy,S.R., Luciani,A., Potter,S.C., Qureshi,M., Richardson,L.J., Salazar,G.A., Smart,A. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.
17. Almeida,H., Tsang,A. and Diallo,A.B. (2019) Supporting supervised learning in fungal biosynthetic gene cluster discovery: new benchmark datasets. In: *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, Piscataway, pp. 1280–1287.
18. Kriventseva,E.V., Kuznetsov,D., Tegenfeldt,F., Manni,M., Dias,R., Simão,F.A. and Zdobnov,E.M. (2018) OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. *Nucleic Acids Res.*, **47**, D807–D811.
19. Inglis,D.O., Binkley,J., Skrzypek,M.S., Arnaud,M.B., Cerqueira,G.C., Shah,P., Wymore,F., Wortman,J.R. and Sherlock,G.

(2013) Comprehensive annotation of secondary metabolite biosynthetic genes and gene clusters of *Aspergillus nidulans*, *A. fumigatus*, *A. niger* and *A. oryzae*. *BMC Microbiol.*, **13**, 91.
20. Vinje,H., Liland,K.H., Almøy,T. and Snipen,L. (2015) Comparing *K*-mer based methods for improved classification of 16S sequences. *BMC Bioinformatics*, **16**, 205.
21. Yang,Y. and Pedersen,J.O. (1997) A comparative study on feature selection in text categorization. In: *Proceedings of the International Conference on Machine Learning (ICML)*. Nashville, Vol. **97**, p. 35.
22. UniProt Consortium (2019) UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.*, **47**, D506–D515.
23. Rost,B. (1999) Twilight zone of protein sequence alignments. *Protein Eng. Des. Sel.*, **12**, 85–94.
24. Skinnider,M.A., Dejong,C.A., Rees,P.N., Johnston,C.W., Li,H., Webster,A.L., Wyatt,M.A. and Magarvey,N.A. (2015) Genomes to natural products prediction informatics for secondary metabolomes (PRISM). *Nucleic Acids Res.*, **43**, 9645–9662.

APPENDIX C

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Improving candidate Biosynthetic Gene Clusters in fungi through reinforcement learning

Hayda Almeida [1,2,3], Adrian Tsang [1,2] and Abdoulaye Baniré Diallo [1,3,4]*

[1] Departement d'Informatique, UQAM, Montréal, QC, Canada
[2] Centre for Structural and Functional Genomics, Concordia University, Montréal, QC, Canada
[3] Laboratoire d'Algèbre, de Combinatoire, et d'Informatique Mathématique, UQAM, Montréal, QC, Canada
[4] Centre of Excellence in Research on Orphan Diseases—Courtois Foundation (CERMO-FC), Montréal, QC, Canada

*To whom correspondence should be addressed.

Abstract

Motivation: Precise identification of Biosynthetic Gene Clusters (BGCs) is a challenging task. Performance of BGC discovery tools is limited by their capacity to accurately predict components belonging to candidate BGCs, often overestimating cluster boundaries. To support optimizing the composition and boundaries of candidate BGCs, we propose reinforcement learning approach relying on protein domains and functional annotations from expert curated BGCs.

Results: The proposed reinforcement learning method aims to improve candidate BGCs obtained with state-of-the-art tools. It was evaluated on candidate BGCs obtained for two fungal genomes, *Aspergillus niger* and *Aspergillus nidulans*. The results highlight an improvement of the gene precision by above 15% for TOUCAN, fungiSMASH and DeepBGC; and cluster precision by above 25% for fungiSMASH and DeepBCG, allowing these tools to obtain almost perfect precision in cluster prediction. This can pave the way of optimizing current prediction of candidate BGCs in fungi, while minimizing the curation effort required by domain experts.

Availability and Implementation: `https://github.com/bioinfoUQAM/RL-bgc-components`

Contact: diallo.abdoulaye@uqam.ca

Supplementary information: Supplementary data is available at Bioinformatics online.

1 Introduction

Filamentous fungi produce a large array of Secondary Metabolites (SM) which play an important role in the survival and development of producing organisms (Keller, 2015). Identifying novel fungal SMs is a field of high interest, given the relevance of these compounds particularly in the pharmaceutical industry for production of various medications (Chavali and Rhee, 2017; Kjærbølling *et al.*, 2019). Biosynthetic pathways that produce SM compounds are encoded by clusters of genes often appearing contiguously in an organism genome, known as Biosynthetic Gene Clusters (BGCs) (Keller, 2019; Kautsar *et al.*, 2020). The genomic diversity of fungal genomes makes accurate identification of BGCs in fungi a highly challenging task for dedicated state-of-the-art tools, and even for manual curation or experimental characterization performed by experts (Kjærbølling *et al.*, 2019). BGCs generally contain minimal components: backbone enzymes, defining the core chemical compound to be produced; and tailoring enzymes, capable of generating variants by modifying the cluster core compound (Keller, 2019). They may also present other components, such as cluster-specific transcription factors, transporters, and hypothetical proteins (Keller, 2015). Fungal BGCs are known to vary considerably in composition (similar clusters with different components), and location (cluster regions overlapping or spanning multiple chromosomes) even among closely related species (Keller, 2019; Kjærbølling *et al.*, 2020; Evdokias *et al.*, 2021).

Various approaches to obtain candidate BGCs (potential sequence regions encoding biosynthesis of SMs) were previously presented (Chavali and Rhee, 2017), such as fungiSMASH (Blin *et al.*, 2021), DeepBGC (Hannigan *et al.*, 2019), and TOUCAN (Almeida *et al.*, 2020). However these approaches show limitations when it comes to the identification of components and boundaries of candidate BGCs, often overpredicting candidate regions. fungiSMASH offers the option to integrate CASSIS (Wolf *et al.*, 2016) to improve cluster boundary prediction. Apart from being a potentially time-consuming option, CASSIS requires curated input, such as gene start and end positions and a reference anchor (backbone) gene, which may not be readily available and therefore limit its stand-alone application to other state-of-the-art BGC discovery approaches.

Obtaining accurate candidate BGCs is a critical step towards chemical synthesis of SM compounds, which can be a complex and costly process as

many of these metabolic pathways are silent or poorly expressed (Montiel *et al.*, 2015; Zhang *et al.*, 2019). In this work, we propose a reinforcement learning approach based on protein family domains from Pfam (El-Gebali *et al.*, 2019) and functional annotations to support optimizing the boundaries and composition of candidate BGCs obtained with state-of-the-art tools, therefore potentially facilitating validation and experimental characterization of SM compounds. Protein domains were previously used in approaches to identify BGCs (Khaldi *et al.*, 2010; Hannigan *et al.*, 2019), and are used here to represent common or shared functional profiles among BGCs, such as presence of relevant components. Reinforcement learning methods are capable of adapting dynamically given feedback received (Neftci and Averbeck, 2019), and therefore might be suitable to handle the overestimation of candidate BGC boundaries, as well as the intrinsic diversity of fungal BGC components, potentially favoring the discovery of novel compounds.

In reinforcement learning, a learning agent interacts directly with an environment through actions in a goal-oriented manner, attempting to maximize its task reward and find an optimal solution (Sutton and Barto, 2018). The agent actions are assigned rewards or penalties, computed based on a given function and according to environment states reached (Sutton and Barto, 2018). When optimizing candidate BGCs, rewards could be assigned for when the agent identifies correct components and properly defines cluster boundaries, while penalties could be given when the agent disregards relevant components from a candidate BGC. While navigating through the environment, the learning agent tries to balance exploitation (acquired knowledge of best actions taken) and exploration (choose actions not tried previously) (Sutton and Barto, 2018). Reinforcement learning approaches had limited applications in biological contexts so far (Mahmud *et al.*, 2018), however results show they generated robust policies and outperformed previous methods in tasks performing multiple sequence alignment (Mircea *et al.*, 2018), controlling gene regulatory networks (Imani and Braga-Neto, 2018), optimizing DNA and protein sequences (Angermueller *et al.*, 2020), and performing *de novo* drug design (Gottipati *et al.*, 2020). Our reinforcement learning approach relies on protein domains and functional annotations of BGC components to optimize candidate BGCs obtained with state-of-the-art tools, which often overestimate cluster boundaries.

2 Methods

The reinforcement learning approach presented here relies on Q-learning (Watkins and Dayan, 1992), a off-policy temporal difference algorithm, which is capable of learning directly from interacting with the environment, without relying on an environment model nor on a long-term value. Rather, a Q-learner uses the next step reward and estimates its gain for the following update and learns from each state transition (Sutton and Barto, 2018). To model a reinforcement learner agent, Pfam protein domains were extracted from curated BGC instances and synthetic non-BGC instances, as described in Section 2.1. Specific rewards were computed for protein domains according to their occurrence in cluster regions of BGC and synthetic non-BGCs, as described in Section 2.2. Test candidate BGCs were then submitted to the reinforcement learning agent to decide on potential BGC components to keep or skip. As a final step, the agent decisions could then be further enhanced by strategies developed based on curated functional annotations of BGC components, as described in Section 2.3. Overall performance is evaluated based on cluster and gene metrics, as described in Section 2.4.

2.1 Datasets

Publicly available fungal BGC benchmark datasets (Almeida *et al.*, 2019) were applied to develop the reinforcement learning approach presented here. Both training and test data are represented through the occurrence of Pfam protein domain features in curated BGC regions, non-BGC regions, and test candidate BGC regions. Previous work has shown the relevance of Pfam domains as features for BGC analysis (Inglis *et al.*, 2013; Kjærbølling *et al.*, 2020) and discovery (Hannigan *et al.*, 2019; Almeida *et al.*, 2020). Pfam domains can indicate the presence of key BGC components as discussed in Section 1, such as polyketide synthase or non-ribosomal peptide synthetase genes encoding backbone enzymes, genes encoding tailoring enzymes, transcription factors or transporters. Genes (or genomic regions, if gene annotations are not available) composing BGCs may contain none to multiple relevant Pfam domains.

Training Publicly available training datasets are presented in Almeida *et al.* (2019). These training datasets are composed of curated fungal BGC instances obtained from MIBiG (Minimum Information about a Biosynthetic Gene cluster) (Kautsar *et al.*, 2020) repository, and synthetic non-BGC instances created from OrthoDB (Kriventseva *et al.*, 2018) fungal orthologous genes. Training datasets of various distributions were generated through sampling of orthologous synthetic non-BGC instances, combined with curated fungal BGC instances (see Almeida *et al.* (2019) for details). Previous work has shown the relevance of orthologous genes in BGC discovery as they indicate conserved genomic regions (Takeda *et al.*, 2014; Almeida *et al.*, 2020), while BGC regions tend to present high genomic diversity even among closely related species (Kjærbølling *et al.*, 2020). Publicly available training datasets of various distributions were previously evaluated in Almeida *et al.* (2020), identifying the most balanced one (50% BGC and 50% non-BGC instances) as the dataset yielding the best performance. For comparison purposes, this is therefore the training dataset applied in our approach.

Testing The decisions taken by the reinforcement learning agent are evaluated on candidate BGCs obtained for the *Aspergillus niger* NRRL3 genomic sequence (publicly available at `https://gb.fungalgenomics.ca/portal`) by three tools: TOUCAN (Almeida *et al.*, 2020), fungiSMASH (Blin *et al.*, 2021), and DeepBGC (Hannigan *et al.*, 2019). *Aspergillus niger* is an organism of interest given its ubiquitous presence, and its importance for industrial processes and biotechnology, which makes it a relevant species in the study of BGC discovery (de Vries *et al.*, 2017; Aguilar-Pontes *et al.*, 2018; Evdokias *et al.*, 2021). To obtain test candidate BGCs from *A. niger* amino acid sequence, we extracted sequentially sliding windows of fixed 10,000 amino acid length with a 30% window overlap (see Almeida *et al.* (2020) for details). *Aspergillus niger* candidate BGCs were then obtained from each BGC discovery tool, based on the same sequentially sliding windows to allow candidate predictions to be compared across the three tools. Before being processed by the proposed reinforcement learning agent, candidate BGCs obtained by all three tools were pre-processed using a majority vote strategy.

Candidate BGC pre-processing – Majority vote: Candidate BGCs contain a set of genomic region identifiers (such as gene names), as well as their corresponding Pfam protein domains. Examples of candidate BGCs are shown in Figure 1. For our experiments, candidate BGCs were obtained based on a test set of *A. niger* genomic regions of 10,000 amino acid sliding windows with a 30% overlap.

On one hand, overlapping regions allow for covering potential BGC fragmentation due to fixed length sliding windows. On the other hand it will also generate repeated regions in candidate BGCs. The majority vote strategy, shown in Figure 1, therefore handles duplicated regions based on a local consensus. It works as follows: each gene g in a candidate BGC is represented by a label vector $L = l_0, l_1, ..., l_m$ where m is the number of candidate BGCs in which g appears and l_i the candidate BGC label (0 for predicted as non-BGC and 1 for predicted as BGC). The majority vote score v_{score} for a gene g is therefore the average value of its predicted labels $\overline{L}$. Sequential genes presenting a $v_{score} \geq 0.5$ are therefore concatenated as

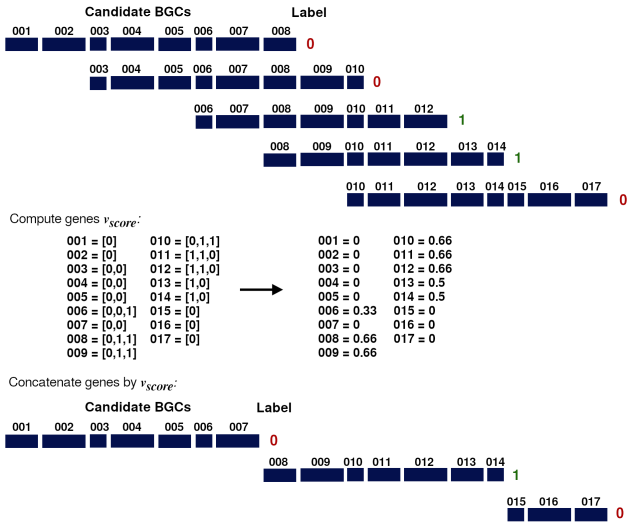


Fig. 1. Computation of majority vote pre-processing for candidate BGCs: regions are merged according to the average score of predicted labels

positive candidate BGCs, while the other genes with a $v_{score} < 0.5$ are concatenated as negative candidate BGCs, up to a limit of 10,000 amino acids per cluster. In our experiments, *A. niger* gene models were used as reference points, however in the lack of gene models, regions of fixed smaller size than the sliding window length could be considered instead.

2.2 Reinforcement learning method

The proposed reinforcement learning approach is based on the temporal-difference and off-policy algorithm Q-learning (Watkins and Dayan, 1992; Sutton and Barto, 2018). In Q-learning, the action-value function Q converges towards an optimal policy, and allows the reinforcement learning agent to decide on the next step. The Q function provides the expected value of an action a, given a state s, and it is dynamically updated during the agent experience of interacting with the environment. Given a set of actions A, a set of states S and respective rewards R at a timestep t, the Q function is computed as:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha\,[\,R_{t+1} + \gamma \max_a Q(S_{t+1,a}) - Q(S_t, A_t)]$$

where α is the learning rate, and γ the discount-rate factor. Additionally, a probability ϵ defines the algorithm exploration versus exploitation rate (Sutton and Barto, 2018). In the context of optimizing BGC components, the reinforcement learning agent chooses the most suitable action within the set of actions $A = keep, skip$ for a candidate BGC, which is a set of states represented by Pfam domains within each gene. At the training phase state rewards were computed by extracting Pfam protein domains from the selected training dataset, as described in Section 2.1. Each protein domain d is represented by an occurrence vector $C = c_0, c_1, ...c_n$, where n is the number training dataset instances, and c_i the domain occurrence per training instance ($c_i > 0$ if a curated BGC instance, and $c_i < 0$ otherwise). To determine the rewards per action R_{keep} and R_{skip} of a domain d, we first compute a score s as follows:

$$s_{keep} = \sum_{x \in C} \frac{x}{|C|} \qquad s_{skip} = |1 - s_{keep}|$$

After computing both s_{keep} and s_{skip}, a $keepSkip$ threshold is applied to finally determine the rewards R_{keep} and R_{skip} for domain d, as in:

$$R_{keep}, R_{skip} = \begin{cases} s_{keep}, -s_{keep} & \text{if } s_{keep} > (s_{skip} * keepSkip) \\ -s_{skip}, s_{skip} & \text{otherwise.} \end{cases}$$

The agent is assigned a penalty for each step it receives a negative reward $R < 0$, with a total penalty computed per episode. An episode is completed when the agent has gone through the entire training dataset.

In the testing phase, the reinforcement learning agent is evaluated by the $keep$ or $skip$ actions it decides on for genes in candidate BGCs. Pfam domains are therefore extracted per gene (or per fixed size region, in case gene models are not available) in candidate BGCs. The optimal action for a gene g containing a set of domains $D = d_0, d_1, ..., d_n$, where n is the number of domains found in g is computed as follows:

$$g_a = argmax(\sum_{i=0}^{n} d_i(R_{keep}), \sum_{i=0}^{n} d_i(R_{skip}))$$

Genes for which $R_{skip} > R_{keep}$ are assigned the action $g_a = skip$, otherwise they are assigned a $g_a = keep$. Only genes assigned a $g_a = keep$ action will be maintained in a given candidate BGC.

2.3 Integrating functional annotations

Biosynthetic gene clusters are generally formed by components that play different roles in the cluster, such as backbone and tailoring enzymes, transcription factors, transporters, and hypothetical proteins, as discussed in Section 1. Backbone and tailoring enzymes for instance are considered essential BGC building blocks for the biosynthesis of SM compounds (Keller, 2019). A total of 85 *A. niger* BGCs (Inglis *et al.*, 2013) were used as our gold standard. To define these BGCs, Inglis *et al.* (2013) described obtaining *in silico* BGCs from state-of-the-art tools, and refining their boundaries based on published experimental data, synteny between BGC genes across multiple species, assignment of experimentally based GO terms, intergenic distance between boundary and adjacent genes. These 85 gold standard *A. niger* BGCs were then manually curated with their functional annotation within clusters. Pfam protein domains were then extracted from functionally annotated BGC gold-standard genes, and associated with a BGC component role. A list of all Pfam domains associated with each annotated BGC component is shown in Supplementary Table 1.

To integrate the functional annotation of BGC components, three strategies were developed based on Pfam domains associated to component roles. The three strategies are applied to enhance the reinforcement learning agent decisions. The `averageAction` strategy handle genes lacking Pfam domains; the `neighborWeight` strategy handles presence of annotations in neighboring genes; and the `dryIslands` strategy handles absence of annotations in contiguous neighboring genes.

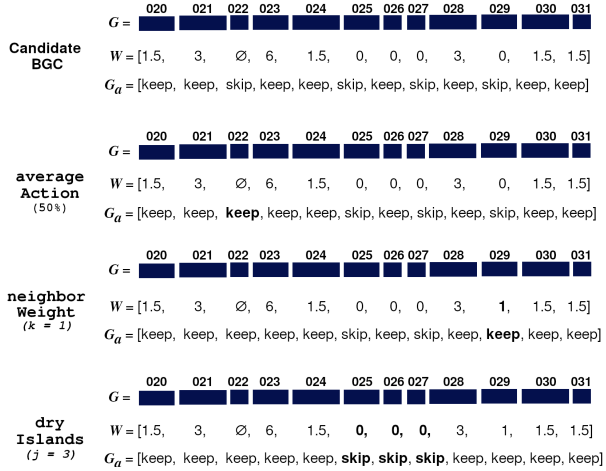


Fig. 2. Example of functional annotation strategies applied to a candidate BGC

Various gold-standard BGC genes, mostly annotated as hypothetical proteins, simply do not contain any Pfam domain annotations and therefore may be directly assigned an action $g_a = skip$. BGC components considered hypothetical proteins may play a relevant role in the cluster (Keller, 2015). However they become challenging components to identify due to their lack of features, which makes them harder to distinguish from the noise within non-relevant components. With the `averageAction` strategy, if the reinforcement learning agent assigns an action $g_a = keep$ for a minimum gene threshold in a candidate BGC G, then genes in G that do not contain protein domains ($D = \varnothing$) will also

be assigned an action $g_a = keep$. Optimization of the minimum threshold ([25%, 50%, 75%]) has yielded 50% as the most suitable value.

To implement the `neighborWeight` and `dryIslands` strategies, a candidate BGC G is assigned a weight vector W, where for each gene g in G a weight w is computed as follows:

$$w = \sum_{i=0}^{n} h_i \qquad h_i = \begin{cases} \beta & \text{if backbone,} \\ \lambda & \text{if other annotation,} \\ \sigma & \text{otherwise.} \end{cases}$$

where n is the number of domains found in g, and h the score associated with the BGC component functional annotation. For the sake of the experiments described in Section 3, we have set the following values: $\beta = 2$ if backbone, $\lambda = 1.5$ if other annotation, and $\sigma = 0$ otherwise. For the `neighborWeight` strategy, if a k number of surrounding neighbors of a gene g present a $\sum_{i=0}^{k} w_i > 1$, then the gene weight $g_w = 1$ and the gene action $g_a = keep$. Optimization of the number of neighbor genes $k = [1, 2, 3]$ has yielded the most suitable $k = 1$. For the `dryIslands` strategy, if $\sum_{i=0}^{j} g_w = 0$ for j sequential genes in G, then the gene action $g_a = skip$. Optimization of the dry island size $j = [3, 4, 5]$ has yielded the most suitable $j = 3$. Figure 2 shows an example of how the reinforcement learning agent decisions are adjusted by the `neighborWeight` and `dryIslands` strategies. Functional annotations of BGC components provide expert domain knowledge and could potentially improve the actions chosen by the reinforcement learning agent, therefore improving precision of candidate BGC components.

2.4 Evaluation metrics

The performance of the reinforcement learning approach proposed here is evaluated in terms of *gene metrics* and *cluster metrics*, for which precision (P), recall (R), F-measure (F-m) are computed. *Cluster metrics* show the performance on identifying cluster regions, and considers as true positives (TPs) candidate BGCs G that have at least one gene g that belongs to the set of gold-standard BGC genes. *Gene metrics* shows the performance on matching genes in candidate BGCs with the complete set of gold-standard BGC genes, and considers as true positives (TPs) the candidate BGC genes that are identical or similar gene matches to gold-standard BGC genes. The similarity between candidate and gold-standard BGC genes is obtained through local BLAST alignment, with minimum thresholds of percent identity $pident \geq 20$ and query coverage $qcov \geq 10$. We also compute the average F-m between *cluster* and *gene metrics* F-m.

3 Results

The reinforcement learning approach proposed here is evaluated on candidate BGCs obtained with three BGC discovery tools: TOUCAN (Almeida *et al.*, 2020), fungiSMASH (Blin *et al.*, 2021) independently and also combined with CASSIS (Wolf *et al.*, 2016), and DeepBGC (Hannigan *et al.*, 2019) for the *A. niger* genome. A total of 85 *A. niger* BGCs (Inglis *et al.*, 2013) were manually curated and are considered as gold standard to evaluate the performance of our reinforcement learning approach on selecting BGC components from candidate BGCs. In Section 3.1 we present an overview of the distribution of genes presenting protein domains associated to functional annotations in the training and test data. Section 3.2 presents the results obtained by the reinforcement learning approach on candidate BGCs from the three tools, and Section 3.3 shows an analysis of reproducibility of the reinforcement learning approach in a second fungal genome, *Aspergillus nidulans*.

3.1 Distribution of domains linked to BGC components

We performed an analysis of the presence of protein domains associated with BGC component roles in genes belonging to the training and test datasets. The distribution of genes that present protein domains associated with BGC component types is shown in Table 1. A protein domain may be associated with multiple component roles if it was found to be present in genes annotated with different components.

Table 1. Domains linked to *A. niger* BGC components in dataset genes

Component type	Training		Test	
	BGCs	non-BGCs	gold BGCs	non-gold BGCs
Backbones	17.0%	2.0%	15.9%	2.2%
Tailoring enzymes	30.5%	7.8%	9.9%	11.9%
Transcription factors	4.8%	2.1%	5.9%	4.3%
Transporters	5.6%	2.8%	7.4%	4.6%
Non-component domains	44.7%	46.93%	49.3%	58.9%
No domains	14.6%	41.15%	15.5%	23.2%
Total # genes	2833	1781	624	11239

It is noticeable from Table 1 that protein domains appearing in BGC components are mostly found among genes in BGCs and gold BGCs instances. Genes that do not contain any protein domains are mostly found among non-BGCs and non-gold BGCs instances. The percentage of genes without any encoded protein domains is higher than that of genes with encoded domains associated to transcription factors and transporters among BGCs and gold BGC genes.

The distribution of genes encoding protein domains associated with backbones in the training data is similar to the that of the test data. Genes without any encoded protein domains also yield a similar distribution among BGCs (14.6%) and gold BGCs (15.5%) genes. Among non-gold-standard BGC genes, more than half encode protein domains that are not associated to any component role. Overall the percentages in Table 1 demonstrate how the presence of protein domains associated to BGC components is ubiquitous both in BGCs and non-BGC regions, which makes correctly identifying BGC components a challenging task.

3.2 Reinforcement learning improves candidate BGCs

We present here the results obtained by the proposed reinforcement learning approach on candidate BGCs obtained with three BGC discovery tools: TOUCAN, fungiSMASH (fungiSMASH/C combined with CASSIS), and DeepBGC. Previously to processing candidate BGCs, we optimized the following reinforcement learning agent parameters: learning rate α, discount-rate factor γ, exploration-exploitation probability ϵ, and the $keepSkip$ threshold, as described in Section 2.2, over a set of 500 episodes on the training data evaluating both fixed and incremental parameter values. The parameters $\alpha = 0.01, \gamma = 0.01, \epsilon = 0.01, keepSkip = 0.5$ yielded the smallest average penalty over 500 episodes. Supplementary Tables 2 and 3 show a summary of the parameter optimization. In this Section, we refer here to `TOUCAN`, `fungiSMASH`, `fungiSMASH/C` and `DeepBGC` as the candidate BGCs directly outputted by each tool; `TOUCAN-Q`, `fungiSMASH-Q`, `fungiSMASH/C-Q` and `DeepBGC-Q` as the candidate BGCs processed by the proposed reinforcement learning approach; and `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/C-Q-all` and `DeepBGC-Q-all` as the candidate BGCs processed by the reinforcement learning approach combined with functional annotation strategies.

Table 2. Performance on *A. niger* candidate BGCs from TOUCAN, fungiSMASH and DeepBGC

model	gene metrics			cluster metrics			average	% gold-std. genes	
	P	R	F-m	P	R	F-m	F-m	negative	skipped
`TOUCAN`	0.269	0.906	0.414	0.963	0.929	0.946	0.68	12.6%	-
`TOUCAN-Q`	0.402	0.68	0.506	0.963	0.929	0.946	0.726	12.6%	26.4%
`TOUCAN-Q-all`	0.409	0.74	0.527	0.963	0.929	0.946	0.737	12.6%	16.2%
`fungiSMASH`	0.341	0.665	0.451	0.649	0.741	0.692	0.571	33.2%	-
`fungiSMASH-Q`	0.521	0.516	0.519	1	0.741	0.851	0.685	33.2%	22.3%
`fungiSMASH-Q-all`	0.495	0.575	0.532	1	0.741	0.851	0.691	33.2%	13.8%
`fungiSMASH/C`	0.371	0.713	0.488	1	0.729	0.844	0.666	34.13%	-
`fungiSMASH/C-Q`	0.523	0.508	0.515	1	0.729	0.844	0.680	34.13%	22.11%
`fungiSMASH/C-Q-all`	0.523	0.508	0.515	1	0.729	0.844	0.680	34.13%	22.11%
`DeepBGC`	0.351	0.481	0.406	0.732	0.612	0.667	0.536	52.4%	-
`DeepBGC-Q`	0.574	0.42	0.485	1	0.612	0.759	0.622	52.4%	12.2%
`DeepBGC-Q-all`	0.538	0.46	0.496	1	0.612	0.759	0.627	52.4%	7.1%

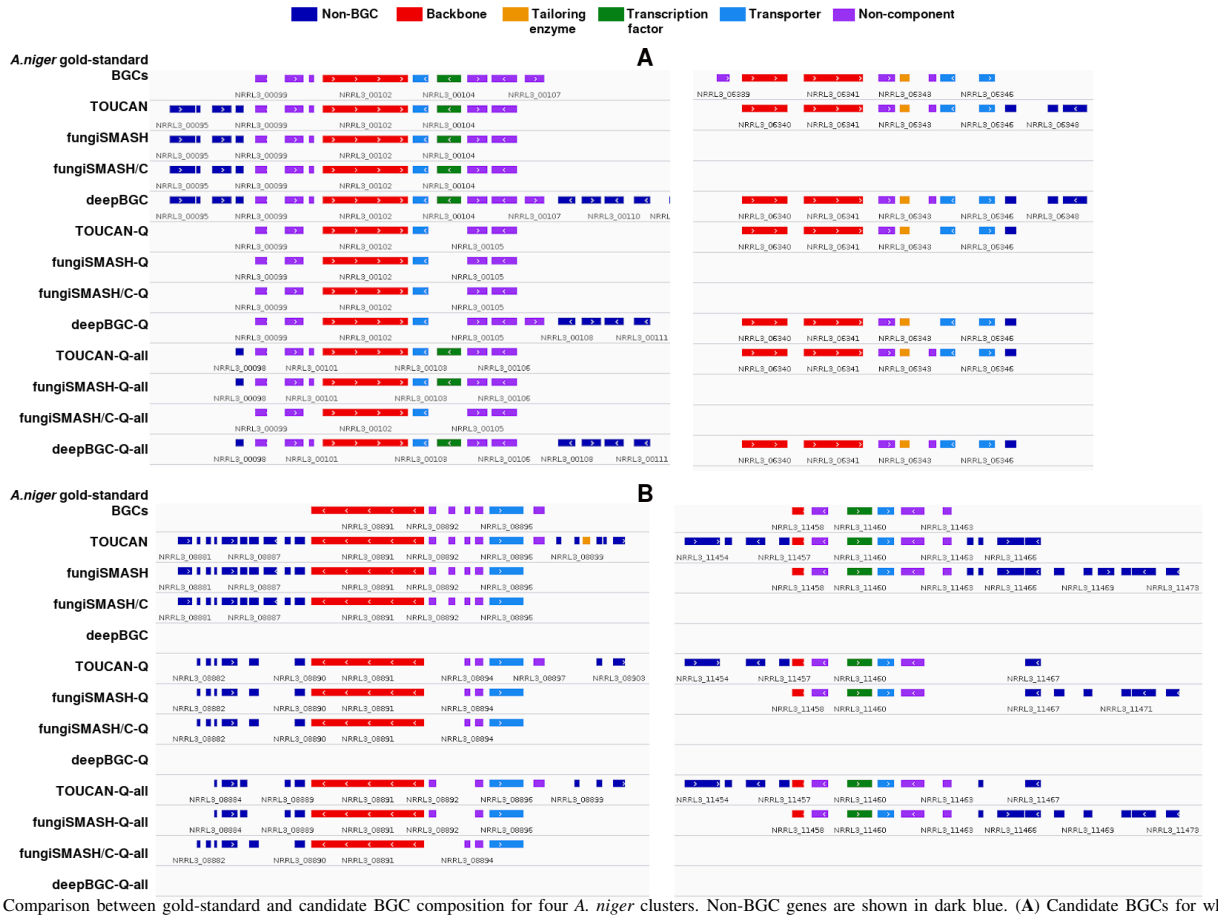


Fig. 3. Comparison between gold-standard and candidate BGC composition for four *A. niger* clusters. Non-BGC genes are shown in dark blue. (**A**) Candidate BGCs for which the reinforcement learning agent correctly skipped most non-BGC genes compared to their polyketide (left) and fatty acid (right) gold standard BGCs. (**B**) Candidate BGCs for which the agent kept most non-BGC genes compared to their two non-ribosomal peptide gold standard BGCs, possibly due to their ambiguous protein domains, which more than half are associated to BGC component roles but do not belong to neighboring clusters.

Table 2 shows the results obtained by the reinforcement learning agent on candidate BGCs for all three tools. As discussed in Section 2.4, cluster metrics show the approach performance on identifying cluster regions, while gene metrics show the performance on matching candidate and gold-standard genes within a BGC. The average F-m shows the overall performance, considering both cluster F-m and gene F-m. The proposed reinforcement learning approach improved gene metrics, more noticeably gene precision in candidate BGCs outputted by all three tools: an increase of 14%, 15.4%, 15.2%, and 18.7% achieved by `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/C-Q-all` and `DeepBGC-Q-all` respectively. For `TOUCAN-Q-all` and `fungiSMASH/C-Q-all`, gene metrics were improved without harming cluster metrics, while for `fungiSMASH-Q-all` and `DeepBGC-Q-all` cluster metrics were also improved considerably, with an F-m increase of 15.9% and 9.2% for `fungiSMASH-Q-all` and `DeepBGC-Q-all` respectively. This indicates that the reinforcement learning agent was capable of improving the precision of candidate BGC components without discarding correctly predicted candidate BGCs, and improving coverage of true positive BGC regions and properly targeting false positive ones predicted by both `fungiSMASH` and `DeepBGC`. The average F-m of all three tools also improved when applying the reinforcement learning agent combined with the functional annotation strategies. An increase in average F-m of 5.7%, 12%, 1.4%, and 9.1% was shown for `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/C-Q-all` and `DeepBGC-Q-all` respectively. Apart from improving gene precision, all candidate BGCs processed by the reinforcement learning agent combined with functional annotation strategies (`Q-all`) yielded a smaller percentage of gold-standard genes skipped, except for `fungiSMASH/C-Q-all`, which yield the same performance for `Q` and `Q-all` models. This suggests that BGC functional annotations can be relevant features to support improving precision of predicted BGCs, and better determine their structure.

Candidate BGCs shown in Figure 3 demonstrate the changes in cluster composition before and after applying the presented reinforcement learning method. A comparison between gold-standard and candidate BGCs in Figure 3-A shows how the reinforcement learning agent improved candidate BGCs from all three tools by correctly skipping non-BGC genes (in blue). Certain cases however are more complex for the agent, given the ambiguity of protein domains in candidate BGC genes. As the examples in Figure 3-B show, more non-BGC genes were kept by the agent, which can lead to processed candidate BGCs to be somehow overpredicted. This behavior could be caused by the fact that domains found in non-BGC genes in Figure 3-B also appear in true positive BGC genes, as opposed to Figure 3-A for which most domains in non-BGC genes were not present in any true positive BGC genes. Among protein domains of non-BGC genes (blue) in Figure 3-B, more than 50% are associated to BGC component roles, and found immediately after true positive BGC genes. Non-BGC genes shown in Figure 3-A presented only 20% of domains linked to BGC component roles. This demonstrates how ambiguous domains in candidate BGCs or their neighboring genes, along with the genomic diversity of these clusters, may increase the complexity of accurately identifying BGC components and boundaries.

Properly identifying BGC components is a challenging task not only for computational approaches that attempt to do so, but even for synthetic approaches that try to express genes composing candidate BGCs (Keller, 2019). Supplementary Table 4 shows an analysis of *A. niger* BGC

component types found in gold-standard BGC genes and components found in candidate BGCs, before and after applying the reinforcement learning approach proposed here. As discussed in Section 2.3, gold BGC genes may contain none to multiple domains, therefore they may present none to multiple functional annotations. Candidate BGCs outputted by `fungiSMASH` and `DeepBGC` presented a smaller number of true positives, and consequently a smaller number of components was found compared to `TOUCAN` candidates, as shown in Supplementary Table 4.

The reinforcement learning agent aims to improve precision of candidate BGC components by removing potentially non-relevant regions. At the same time, the agent has to handle ambiguous genes that map to protein domains, normally found in both BGC and non-BGC instances. The number of backbone genes properly identified by `TOUCAN` (92.9%), `fungiSMASH` (70.7%), `fungiSMASH/C` (69.7%) and `DeepBGC` (64.6%) remains the same even after processing by the reinforcement learning agent for all three tools. This could indicate that the reinforcement learning agent was capable of learning correctly the relevance of regions encoding such enzymes. Backbone enzymes are vital components of BGCs (Kjærbølling *et al.*, 2020), and their accurate identification could demonstrate the robustness of a BGC discovery method. Transcription factors and transporters in `DeepBGC` candidate BGCs were maintained by the reinforcement learning agent, however the overall percentage of these components remains lower than the percentage identified by `TOUCAN` and `fungiSMASH`.

Some BGC genes are not associated to any component role, and often do not even contain any Pfam protein domains, as discussed in Section 2.3. Usually considered as hypothetical proteins, these genes pose a challenge on correctly identifying BGC components, and could be overlooked by BGC discovery approaches since their computational representation will likely be more analogous to non-BGC regions. These hypothetical proteins can seem to diverge from other BGC components but they may play important self-protection roles for the organism producing a SM compound (Keller, 2019). As shown in Supplementary Table 4, genes without any domains were the most missed by the reinforcement learning approach (`Q`) among candidate BGCs from all three tools. The `averageAction` strategy aims to address this issue by keeping candidate BGC genes without domains when at least a minimum 50% threshold of genes within a candidate BGC are assigned the action $keep$. A more lenient threshold was experimented with for `averageAction` strategy, however it can lead to the agent identifying a higher number false positives − genes without protein domains and often associated with non-relevant BGC regions − resulting in a decrease in precision.

3.3 Reproducibility in *Aspergillus nidulans* candidate BGCs

Similarly to *A. niger*, *A. nidulans* is a source of highly useful SMs compounds which are also largely utilized in the pharmaceutical industry (Inglis *et al.*, 2013; Drott *et al.*, 2020). To further evaluate the reproducibility of the proposed reinforcement learning approach, we processed the *A. nidulans* genome considering as gold standard a total of 72 gold standard BGCs presented in Drott *et al.* (2020). Assignment of functional annotations to BGC components is a costly and time-consuming process. Since manually curated component annotations were not available for *A. nidulans* gold-standard BGCs, we generated pseudo-annotations by assigning potential component types to gold-standard BGC genes based on similar keywords found in their protein domain descriptions matching annotated BGC components in *A. niger*.

For instance, backbone pseudo-annotations were assigned to genes containing similar descriptions to the annotated backbone genes in *A. niger*, such as polyketide synthases, non-ribosomal peptide synthetases, dimethylallyltryptophan synthases and terpene synthases. Tailoring enzymes pseudo-annotations were considered as genes containing similar descriptions of *A. niger* tailoring enzymes, such as methyltransferases, monooxygenases, and oxidoreductases. Transcription factor and transporter pseudo-annotations were assigned to genes presenting domains described as presenting these functions. A list of all Pfam domains associated with a pseudo-functional annotation is shown in Supplementary Table 5. The distribution of component pseudo-annotations found in the training data and gold-standard genes for *A. nidulans* is shown in Table 3.

Table 3. Domains linked to *A. nidulans* pseudo BGC components dataset genes

Pseudo component type	**Training**		**Test**	
	BGCs	**non-BGCs**	**gold BGCs**	**non-gold BGCs**
Backbones	17.5%	2.13%	20%	2.45%
Tailoring enzymes	36%	3.70%	31.63%	4.5%
Transcription factors	4.83%	2.35%	5.92%	3.92%
Transporters	5.82%	3.65%	7.55%	5.2%
Non-component domains	33.15%	48.28%	35.3%	62.12%
No domains	14.6%	41.15%	12.65%	22.8%
Total # genes	2833	1781	490	10002

Candidate BGCs for *A. nidulans* were obtained from TOUCAN, fungiSMASH, fungiSMASH combined with CASSIS, and DeepBGC in the same manner as candidates were obtained for *A. niger*, performing the test set pre-processing using a majority vote of overlapping sliding windows of fixed 10,000 amino acids as described in Section 2.1 by the reinforcement learning agent on TOUCAN, fungiSMASH, and DeepBGC candidate BGCs for *A. nidulans* are shown in Table 4.

Table 4. Performance on *A. nidulans* candidate BGCs from the three tools

	gene metrics			**cluster metrics**			**average**	**% gold genes**	
model	P	R	F-m	P	R	F-m	F-m	negative	skipped
`TOUCAN`	0.272	0.681	0.389	1	0.685	0.813	0.601	32.24%	-
`TOUCAN-Q`	0.441	0.591	0.505	1	0.681	0.810	0.657	32.24%	13.47%
`TOUCAN-Q-all`	0.402	0.646	0.495	1	0.681	0.810	0.653	32.24%	7.55%
`fungiSMASH`	0.319	0.727	0.443	0.817	0.795	0.806	0.624	30.61%	-
`fungiSMASH-Q`	0.479	0.592	0.53	1	0.781	0.877	0.703	30.61%	15.92%
`fungiSMASH-Q-all`	0.469	0.605	0.529	1	0.736	0.848	0.688	30.61%	13.88%
`fungiSMASH/C`	0.318	0.762	0.449	1	0.792	0.884	0.666	28.16%	-
`fungiSMASH/C-Q`	0.484	0.581	0.528	1	0.778	0.875	0.702	28.16%	19.18%
`fungiSMASH/C-Q-all`	0.484	0.581	0.528	1	0.778	0.875	0.702	28.16%	19.18%
`DeepBGC`	0.328	0.493	0.394	0.723	0.466	0.567	0.480	50.61%	-
`DeepBGC-Q`	0.491	0.441	0.465	1	0.466	0.636	0.550	50.61%	8.57%
`DeepBGC-Q-all`	0.473	0.492	0.482	1	0.472	0.642	0.562	50.61%	2.86%

Like in *A. niger*, the reinforcement learning approach improved gene precision in candidate BGCs outputted by all three tools: an increase of 13%, 15%, 16.6%, and 14.5% is seen for `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/C-Q-all` and `DeepBGC-Q-all` respectively. Gene metrics also yield improvement in *A. nidulans* without harming the cluster metrics for `TOUCAN-Q-all`, while improving it for `fungiSMASH-Q-all` and `DeepBGC-Q-all`, and only showing a less than 1% difference for `fungiSMASH/C-Q-all`. As previously mentioned, this indicates that the reinforcement learning agent was able to improve the precision of candidate BGC components without discarding correctly predicted candidate BGC regions. Average F-m performance also showed improvement for all three tools when compared to their original candidate BGCs, with an increase of 5.2%, 6.4%, 3.6%, and 8.2% for `TOUCAN-Q-all`, `fungiSMASH-Q-all`, `fungiSMASH/c-Q-all` and `DeepBGC-Q-all`. When comparing the models relying on the reinforcement learning agent only (`Q`) versus the ones relying on both the agent and the functional annotation strategies (`Q-all`) we can observe improvements on gene recall and the percentage of gold-standard genes skipped, but a small drop on gene precision, with the exception of `fungiSMASH/C` models that yield similar performance for `Q` and `Q-all` models. Likely, the usage of *A. nidulans* pseudo-annotations resulted in a slight increase of false positive components. However it might be an useful alternative when manually curated functional annotations are not available, or also when wanting to favor recall over precision.

Candidate BGC composition before and after applying the reinforcement learning agent is shown in Supplementary Figure 1. Similarly to *A. niger*, Supplementary Figure 1-A demonstrates

improvements in candidate BGCs achieved by the agent by skipping non-BGC genes (in blue). When handling more complex cases, as shown in Supplementary Figure 1-B, the agent kept most non-BGC genes, potentially resulting in overpredicted boundaries. Approximately 50% of protein domains from non-BGC genes in Supplementary Figure 1-B were associated to pseudo-functional annotations in *A. nidulans*, while only 20% of domains from non-BGC genes in Supplementary Figure 1-A were associated to any annotation.

4 Discussion and Conclusion

Secondary metabolites are a crucial source of compounds that benefit human health. Identifying BGCs responsible for synthesizing these compounds in fungi may lead to the discovery of new natural products, and potentially novel drugs. State-of-the-art tools for BGC discovery often overpredict BGC boundaries and components. In fungi BGCs are typically encoded by a high diversity of components, known to vary even among evolutionary closely related species. Precise identification of BGC components is therefore a challenging task, and can facilitate the validation and experimental characterization of SM compounds. In this work we presented a reinforcement learning method and functional annotation strategies to support optimizing fungal candidate BGCs obtained with state-of-the-art tools. We evaluated our proposed approach on candidate BGCs obtained for *A. niger* and *A. nidulans* by three BGC discovery tools: TOUCAN, based on supervised learning; fungiSMASH, based on probabilistic and rule-based methods, as well as a version of fungiSMASH combined with CASSIS for cluster border prediction; and DeepBGC, based on deep learning. The results obtained by our reinforcement learning approach yield improvement of cluster and gene precision of BGC candidates obtained from all three tools, without affecting correctly predicted BGC regions.

Overall, best average F-m performances obtained for *A. niger* relied on the combination of the reinforcement learning method and functional annotation strategies based on expert curation. In *A. nidulans*, even pseudo-functional annotations were able to improve BGC gene recall, and reduce the number of gold-standard genes being skipped by the reinforcement learning agent. This indicates that, when available, integrating functional annotations further advances the approach capabilities. Functional annotations may however not always be publicly available, since they can be time-consuming to obtain. The results have shown however that the reinforcement learning approach alone, based solely on Pfam protein domains, improved average F-m of candidate BGCs in average by 7% in *A. niger* and 5.8% in *A. nidulans*. The performance of the reinforcement learning approach indicates its ability to identify the relevance of certain protein domain profiles associated with fungal BGCs, supporting previous findings of these as relevant features in the context of BGC discovery (Khaldi *et al.*, 2010; Cimermancic *et al.*, 2014; Hannigan *et al.*, 2019).

The results achieved through reinforcement learning in candidate BGCs from both fungal genomes evaluated are indicative of the method generalization power and robustness by handling candidate BGCs from different organisms. Additionally a preliminary analysis, shown in Supplementary Figure 2, was performed by processing completely annotated MIBiG BGCs from three fungal species using the proposed reinforcement learning method. The fact that the completely annotated BGCs were kept almost intact by the reinforcement learning method, with or without functional annotation strategies is another indication of its potential robustness on properly identifying essential BGC components for the SM biosynthesis.

As discussed in Section 1, properly identifying BGC components can be a great challenge, given the underlying high diversity of BGCs. Moreover, another important challenge related to the scarcity of validated fungal BGC data are potential biases, both of cluster boundary definition, as well as of BGC composition, since most MIBiG fungal BGCs composing the training dataset are polyketide synthases. While reported as manually curated (Kautsar *et al.*, 2020), most MIBiG fungal BGCs in the training dataset are partially annotated, and Inglis *et al.* (2013) presented limited experimental characterization evidence for the annotated *Aspergillus* BGCs considered as gold standard BGCs in this work. While the number of completely or partially annotated fungal BGCs is scarce, the number of experimentally characterized clusters is even smaller. This only highlights that improving the availability of validated and experimentally characterized fungal BGC data can be a fundamental step towards supporting the development of robust *in silico* approaches for fungal BGC discovery.

Data availability The source code as well as all datasets applied in our experiments are made publicly available at `https://github.com/bioinfoUQAM/RL-bgc-components`. All material is available under the MIT software license. The datasets used in this work were obtained from open access databases, which are available under the Creative Commons Attribution 4.0 international license.

Acknowledgements

We acknowledge the bioinformatics teams at CSFG and UQAM; Sylvester Palys, Marie Beigas, and Isabelle Benoit for helping with manual curation of functional annotations; and Maria Victoria Aguilar and Amine Remita for helpful discussions and brainstorming.

Funding

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) and the Fonds de recherche du Québec – Nature et technologies (FRQNT).

References

Aguilar-Pontes, M., Brandl, J., McDonnell, E., Strasser, K., Nguyen, T., Riley, R., Mondo, S., Salamov, A., Nybo, J. L., Vesth, T. C., *et al.* (2018). The gold-standard genome of Aspergillus niger NRRL 3 enables a detailed view of the diversity of sugar catabolism in fungi. *Studies in Mycology*, **91**, 61–78.

Almeida, H., Tsang, A., and Diallo, A. B. (2019). Supporting supervised learning in fungal Biosynthetic Gene Cluster discovery: new benchmark datasets. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1280–1287. IEEE.

Almeida, H., Palys, S., Tsang, A., and Diallo, A. B. (2020). TOUCAN: a framework for fungal biosynthetic gene cluster discovery. *NAR Genomics and Bioinformatics*, **2**(4). lqaa098.

Angermueller, C., Dohan, D., Belanger, D., Deshpande, R., Murphy, K., and Colwell, L. (2020). Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations (ICLR)*.

Blin, K., Shaw, S., Kloosterman, A. M., Charlop-Powers, Z., van Wezel, G. P., Medema, M. H., and Weber, T. (2021). antiSMASH 6.0: improving cluster detection and comparison capabilities. *Nucleic Acids Research*.

Chavali, A. K. and Rhee, S. Y. (2017). Bioinformatics tools for the identification of gene clusters that biosynthesize specialized metabolites. *Briefings in Bioinformatics*, **19**(5), 1022–1034.

Cimermancic, P., Medema, M. H., Claesen, J., Kurita, K., Brown, L. C. W., Mavrommatis, K., Pati, A., Godfrey, P. A., Koehrsen, M., Clardy, J., *et al.* (2014). Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. *Cell*, **158**(2), 412–421.

de Vries, R. P., Riley, R., Wiebenga, A., Aguilar-Osorio, G., Amillis, S., Uchima, C. A., Anderluh, G., Asadollahi, M., Askin, M., Barry, K., *et al.* (2017). Comparative genomics reveals high biological diversity and

specific adaptations in the industrially and medically important fungal genus Aspergillus. *Genome Biology*, **18**(28).

Drott, M., Bastos, R., Rokas, A., Ries, L., Gabaldón, T., Goldman, G., Keller, N., and Greco, C. (2020). Diversity of Secondary Metabolism in Aspergillus nidulans Clinical Isolates. *Msphere*, **5**(2).

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A., *et al.* (2019). The Pfam protein families database in 2019. *Nucleic Acids Research*, **47**(D1), D427–D432.

Evdokias, G., Semper, C., Mora-Ochomogo, M., Di Falco, M., Nguyen, T. T. M., Savchenko, A., Tsang, A., and Benoit-Gelber, I. (2021). Identification of a Novel Biosynthetic Gene Cluster in Aspergillus niger Using Comparative Genomics. *Journal of Fungi*, **7**(5), 374.

Gottipati, S. K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J., *et al.* (2020). Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 3668–3679.

Hannigan, G. D., Prihoda, D., Palicka, A., Soukup, J., Klempir, O., Rampula, L., Durcak, J., Wurst, M., Kotowski, J., Chang, D., *et al.* (2019). A deep learning genome-mining strategy for biosynthetic gene cluster prediction. *Nucleic Acids Research*, **47**, e110.

Imani, M. and Braga-Neto, U. M. (2018). Control of gene regulatory networks using Bayesian inverse reinforcement learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **16**(4), 1250–1261.

Inglis, D. O., Binkley, J., Skrzypek, M. S., Arnaud, M. B., Cerqueira, G. C., Shah, P., Wymore, F., Wortman, J. R., and Sherlock, G. (2013). Comprehensive annotation of secondary metabolite biosynthetic genes and gene clusters of Aspergillus nidulans, A. fumigatus, A. niger and A. oryzae. *BMC Microbiology*, **13**(91).

Kautsar, S. A., Blin, K., Shaw, S., Navarro-Muñoz, J. C., Terlouw, B. R., van der Hooft, J. J., Van Santen, J. A., Tracanna, V., Suarez Duran, H. G., Pascal Andreu, V., *et al.* (2020). MIBiG 2.0: a repository for biosynthetic gene clusters of known function. *Nucleic Acids Research*, **48**(D1), D454–D458.

Keller, N. P. (2015). Translating biosynthetic gene clusters into fungal armor and weaponry. *Nature Chemical Biology*, **11**(9), 671–677.

Keller, N. P. (2019). Fungal secondary metabolism: regulation, function and drug discovery. *Nature Reviews Microbiology*, **17**(3), 167–180.

Khaldi, N., Seifuddin, F. T., Turner, G., Haft, D., Nierman, W. C., Wolfe, K. H., and Fedorova, N. D. (2010). SMURF: genomic mapping of fungal secondary metabolite clusters. *Fungal Genetics and Biology*, **47**(9), 736–741.

Kjærbølling, I., Mortensen, U. H., Vesth, T., and Andersen, M. R. (2019). Strategies to establish the link between biosynthetic gene clusters and secondary metabolites. *Fungal Genetics and Biology*, **130**, 107–121.

Kjærbølling, I., Vesth, T., Frisvad, J. C., Nybo, J. L., Theobald, S., Kildgaard, S., Petersen, T. I., Kuo, A., Sato, A., Lyhne, E. K., *et al.* (2020). A comparative genomics study of 23 Aspergillus species from section Flavi. *Nature Communications*, **11**(1106).

Kriventseva, E. V., Kuznetsov, D., Tegenfeldt, F., Manni, M., Dias, R., Simão, F. A., and Zdobnov, E. M. (2018). OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. *Nucleic Acids Research*, **47**(D1), D807–D811.

Mahmud, M., Kaiser, M. S., Hussain, A., and Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE Transactions on Neural Networks and Learning Systems*, **29**(6), 2063–2079.

Mircea, I.-G., Bocicor, I., and Czibula, G. (2018). A Reinforcement Learning Based Approach to Multiple Sequence Alignment. In *Soft Computing Applications*, pages 54–70. Springer.

Montiel, D., Kang, H.-S., Chang, F.-Y., Charlop-Powers, Z., and Brady, S. F. (2015). Yeast homologous recombination-based promoter engineering for the activation of silent natural product biosynthetic gene clusters. *Proceedings of the National Academy of Sciences*, **112**(29), 8953–8958.

Neftci, E. O. and Averbeck, B. B. (2019). Reinforcement learning in artificial and biological systems. *Nature Machine Intelligence*, **1**(3), 133–143.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Takeda, I., Umemura, M., Koike, H., Asai, K., and Machida, M. (2014). Motif-independent prediction of a secondary metabolism gene cluster using comparative genomics: application to sequenced genomes of Aspergillus and ten other filamentous fungal species. *DNA Research*, **21**(4), 447–457.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, **8**(3-4), 279–292.

Wolf, T., Shelest, V., Nath, N., and Shelest, E. (2016). CASSIS and SMIPS: promoter-based prediction of secondary metabolite gene clusters in eukaryotic genomes. *Bioinformatics*, **32**(8), 1138–1143.

Zhang, X., Elliot, M. A., *et al.* (2019). Unlocking the trove of metabolic treasures: activating silent biosynthetic gene clusters in bacteria and fungi. *Current Opinion in Microbiology*, **51**, 9–15.

BIBLIOGRAPHY

Agrawal, P., Amir, S., Barua, D., Mohanty, D. et al. (2021). RiPPMiner-Genome: A Web Resource for Automated Prediction of Crosslinked Chemical Structures of RiPPs by Genome Mining. *Journal of Molecular Biology*, *433*(11), 166887.

Agrawal, P., Khater, S., Gupta, M., Sain, N. & Mohanty, D. (2017). RiPPMiner: a bioinformatics resource for deciphering chemical structures of RiPPs based on prediction of cleavage and cross-links. *Nucleic Acids Research*, *45*(W1), W80–W88.

Aguilar-Pontes, M., Brandl, J., McDonnell, E., Strasser, K., Nguyen, T., Riley, R., Mondo, S., Salamov, A., Nybo, J. L., Vesth, T. C. et al. (2018). The gold-standard genome of Aspergillus niger NRRL 3 enables a detailed view of the diversity of sugar catabolism in fungi. *Studies in Mycology*, *91*, 61–78.

Almeida, H., Meurs, M.-J., Kosseim, L., Butler, G. & Tsang, A. (2014). Machine learning for biomedical literature triage. *PloS one*, *9*(12), e115892.

Almeida, H., Palys, S., Tsang, A. & Diallo, A. B. (2020). TOUCAN: a framework for fungal biosynthetic gene cluster discovery. *NAR Genomics and Bioinformatics*, *2*(4). lqaa098, `http://dx.doi.org/10.1093/nargab/lqaa098`. Retrieved from `https://doi.org/10.1093/nargab/lqaa098`

Almeida, H., Tsang, A. & Diallo, A. B. (2019). Supporting supervised learning in fungal Biosynthetic Gene Cluster discovery: new benchmark datasets. In *In proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1280–1287. IEEE. `http://dx.doi.org/10.1109/BIBM47256.2019.8983041`

Angermueller, C., Dohan, D., Belanger, D., Deshpande, R., Murphy, K. & Colwell, L. (2020). Model-based reinforcement learning for biological sequence design. In

In proceedings of the 8th International Conference on Learning Representations (ICLR).

Angermueller, C., Pärnamaa, T., Parts, L. & Stegle, O. (2016). Deep learning for computational biology. *Molecular Systems Biology*, *12*(7), 878.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T. et al. (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics*, *25*(1), 25–29.

Aslam, B., Wang, W., Arshad, M. I., Khurshid, M., Muzammil, S., Rasool, M. H., Nisar, M. A., Alvi, R. F., Aslam, M. A., Qamar, M. U. et al. (2018). Antibiotic resistance: a rundown of a global crisis. *Infection and Drug Resistance*, *11*, 1645.

Bailey, T. L., Johnson, J., Grant, C. E. & Noble, W. S. (2015). The MEME suite. *Nucleic Acids Research*, *43*(W1), W39–W49.

Berman, J. & Krysan, D. J. (2020). Drug resistance and tolerance in fungi. *Nature Reviews Microbiology*, *18*(6), 319–331.

Bills, G. F. & Gloer, J. B. (2016). Biologically active secondary metabolites from the fungi. *Microbiology Spectrum*, *4*(6), 4–6.

Binns, D., Dimmer, E., Huntley, R., Barrell, D., O'donovan, C. & Apweiler, R. (2009). QuickGO: a web-based tool for Gene Ontology searching. *Bioinformatics*, *25*(22), 3045–3046.

Blin, K., Medema, M. H., Kottmann, R., Lee, S. Y. & Weber, T. (2016). The antiSMASH database, a comprehensive database of microbial secondary metabolite biosynthetic gene clusters. *Nucleic Acids Research*, *45*(D1), gkw960.

Blin, K., Pascal Andreu, V., de los Santos, E. L. C., Del Carratore, F., Lee, S. Y., Medema, M. H. & Weber, T. (2018). The antiSMASH database version 2: a comprehensive resource on secondary metabolite biosynthetic gene clusters. *Nucleic Acids Research*, *47*(D1), D625–D630.

Blin, K., Shaw, S., Kloosterman, A. M., Charlop-Powers, Z., van Wezel, G. P.,

Medema, M. H. & Weber, T. (2021). antiSMASH 6.0: improving cluster detection and comparison capabilities. *Nucleic Acids Research*.

Blin, K., Wolf, T., Chevrette, M. G., Lu, X., Schwalen, C. J., Kautsar, S. A., Suarez Duran, H. G., De Los Santos, E. L., Kim, H. U., Nave, M. et al. (2017). antiSMASH 4.0 — improvements in chemistry prediction and gene cluster boundary identification. *Nucleic Acids Research*, *45*(W1), W36–W41.

Borrajo, L., Romero, R., Iglesias, E. L. & Marey, C. R. (2011). Improving imbalanced scientific text classification using sampling strategies and dictionaries. *Journal of Integrative Bioinformatics*, *8*(3), 90–104.

Breitwieser, F. P., Baker, D. & Salzberg, S. L. (2018). KrakenUniq: confident and fast metagenomics classification using unique k-mer counts. *Genome Biology*, *19*(1), 1–10.

Cairns, T. C., Nai, C. & Meyer, V. (2018). How a fungus shapes biotechnology: 100 years of Aspergillus niger research. *Fungal Biology and Biotechnology*, *5*(1), 1–14.

Cerqueira, G. C., Arnaud, M. B., Inglis, D. O., Skrzypek, M. S., Binkley, G., Simison, M., Miyasato, S. R., Binkley, J., Orvis, J., Shah, P. et al. (2014). The Aspergillus Genome Database: multispecies curation and incorporation of RNA-Seq data to improve structural gene annotations. *Nucleic Acids Research*, *42*(D1), D705–D710.

Chaudhary, A. K., Dhakal, D. & Sohng, J. K. (2013). An insight into the "-omics" based engineering of streptomycetes for secondary metabolite overproduction. *BioMed Research International*, *2013*.

Chavali, A. K. & Rhee, S. Y. (2017). Bioinformatics tools for the identification of gene clusters that biosynthesize specialized metabolites. *Briefings in Bioinformatics*, *19*(5), 1022–1034.

Chor, B., Horn, D., Goldman, N., Levy, Y. & Massingham, T. (2009). Genomic DNA k-mer spectra: models and modalities. *Genome Biology*, *10*(10), 1–10.

Cimermancic, P., Medema, M. H., Claesen, J., Kurita, K., Brown, L. C. W., Mavrommatis, K., Pati, A., Godfrey, P. A., Koehrsen, M., Clardy, J. et al. (2014). Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. *Cell*, *158*(2), 412–421.

Conway, K. R. & Boddy, C. N. (2012). ClusterMine360: a database of microbial PKS/NRPS biosynthesis. *Nucleic Acids Research*, *41*(D1), D402–D407.

Croteau, R., Kutchan, T. M., Lewis, N. G. et al. (2000). Natural Products (Secondary Metabolites). *Biochemistry and Molecular Biology of Plants*, *24*, 1250–1319.

de Los Santos, E. L. (2019). NeuRiPP: Neural network identification of RiPP precursor peptides. *Scientific Reports*, *9*(1), 1–9.

de Vries, R. P., Riley, R., Wiebenga, A., Aguilar-Osorio, G., Amillis, S., Uchima, C. A., Anderluh, G., Asadollahi, M., Askin, M., Barry, K. et al. (2017). Comparative genomics reveals high biological diversity and specific adaptations in the industrially and medically important fungal genus Aspergillus. *Genome Biology*, *18*(1), 28.

Drott, M., Bastos, R., Rokas, A., Ries, L., Gabaldón, T., Goldman, G., Keller, N. & Greco, C. (2020). Diversity of Secondary Metabolism in Aspergillus nidulans Clinical Isolates. *mSphere*, *5*(2), e00156–20.

Eastman, P., Shi, J., Ramsundar, B. & Pande, V. S. (2018). Solving the RNA design problem with reinforcement learning. *PLOS Computational Biology*, *14*(6), 1–15.

El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A. et al. (2019). The Pfam protein families database in 2019. *Nucleic Acids Research*, *47*(D1), D427–D432.

Evdokias, G., Semper, C., Mora-Ochomogo, M., Di Falco, M., Nguyen, T. T. M., Savchenko, A., Tsang, A. & Benoit-Gelber, I. (2021). Identification of a Novel Biosynthetic Gene Cluster in Aspergillus niger Using Comparative Genomics.

Journal of Fungi, *7*(5), 374.

Frisvad, J. C., Møller, L. L., Larsen, T. O., Kumar, R. & Arnau, J. (2018). Safety of the fungal workhorses of industrial biotechnology: update on the mycotoxin and secondary metabolite potential of Aspergillus niger, Aspergillus oryzae, and Trichoderma reesei. *Applied Microbiology and Biotechnology*, *102*(22), 9481–9515.

Gene Ontology Consortium (2021). The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Research*, *49*(D1), D325–D334.

Gerke, J., Bayram, Ö., Feussner, K., Landesfeind, M., Shelest, E., Feussner, I. & Braus, G. H. (2012). Breaking the silence: protein stabilization uncovers silenced biosynthetic gene clusters in the fungus Aspergillus nidulans. *Applied and Environmental Microbiology*, *78*(23), 8234–8244.

Gerlt, J. A., Bouvier, J. T., Davidson, D. B., Imker, H. J., Sadkhin, B., Slater, D. R. & Whalen, K. L. (2015). Enzyme function initiative-enzyme similarity tool (efi-est): A web tool for generating protein sequence similarity networks. *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics*, *1854*(8), 1019–1037. `http://dx.doi.org/https://doi.org/10.1016/j.bbapap.2015.04.015`. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1570963915001120`

Gluck-Thaler, E., Haridas, S., Binder, M., Grigoriev, I. V., Crous, P. W., Spatafora, J. W., Bushley, K. & Slot, J. C. (2020). The Architecture of Metabolism Maximizes Biosynthetic Diversity in the Largest Class of Fungi. *Molecular Biology and Evolution*, *37*(10), 2838–2856.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016). *Deep Learning*, volume 1. MIT Press Cambridge.

Gottipati, S. K., Sattarov, B., Niu, S., Pathak, Y., Wei, H., Liu, S., Blackburn, S., Thomas, K., Coley, C., Tang, J. et al. (2020). Learning to navigate the synthetically accessible chemical space using reinforcement learning. In *In proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 3668–3679.

Hadjithomas, M., Chen, I.-M. A., Chu, K., Huang, J., Ratner, A., Palaniappan, K., Andersen, E., Markowitz, V., Kyrpides, N. C. & Ivanova, N. N. (2016). IMG-ABC: new features for bacterial secondary metabolism analysis and targeted biosynthetic gene cluster discovery in thousands of microbial genomes. *Nucleic Acids Research*, *45*(D1), D560–D565.

Haft, D. H., Selengut, J. D., Richter, R. A., Harkins, D., Basu, M. K. & Beck, E. (2012). TIGRFAMs and Genome Properties in 2013. *Nucleic Acids Research*, *41*(D1), D387–D395.

Hannigan, G. D., Prihoda, D., Palicka, A., Soukup, J., Klempir, O., Rampula, L., Durcak, J., Wurst, M., Kotowski, J., Chang, D. et al. (2019). A deep learning genome-mining strategy for biosynthetic gene cluster prediction. *Nucleic Acids Research*, *47*(18), e110–e110.

Hashim, F. A., Mabrouk, M. S. & Al-Atabany, W. (2019). Review of different sequence motif finding algorithms. *Avicenna Journal of Medical Biotechnology*, *11*(2), 130.

Hasin, Y., Seldin, M. & Lusis, A. (2017). Multi-omics approaches to disease. *Genome Biology*, *18*(1), 1–15.

He, H. & Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons.

Hyatt, D., Chen, G.-L., LoCascio, P. F., Land, M. L., Larimer, F. W. & Hauser, L. J. (2010). Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, *11*(1), 1–11.

Imani, M. & Braga-Neto, U. M. (2018). Control of gene regulatory networks using Bayesian inverse reinforcement learning. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *16*(4), 1250–1261.

Inglis, D. O., Binkley, J., Skrzypek, M. S., Arnaud, M. B., Cerqueira, G. C., Shah, P., Wymore, F., Wortman, J. R. & Sherlock, G. (2013). Comprehensive annotation of secondary metabolite biosynthetic genes and gene clusters of Aspergillus nidulans, A. fumigatus, A. niger and A. oryzae. *BMC Microbiology*, *13*(91).

Jeon, W. & Kim, D. (2020). Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors. *Scientific Reports*, *10*(1), 1–11.

Jones, P., Binns, D., Chang, H.-Y., Fraser, M., Li, W., McAnulla, C., McWilliam, H., Maslen, J., Mitchell, A., Nuka, G. et al. (2014). InterProScan 5: genome-scale protein function classification. *Bioinformatics*, *30*(9), 1236–1240.

Kautsar, S. A., Blin, K., Shaw, S., Navarro-Muñoz, J. C., Terlouw, B. R., van der Hooft, J. J., Van Santen, J. A., Tracanna, V., Suarez Duran, H. G., Pascal Andreu, V. et al. (2020). MIBiG 2.0: a repository for biosynthetic gene clusters of known function. *Nucleic Acids Research*, *48*(D1), D454–D458.

Kautsar, S. A., Blin, K., Shaw, S., Navarro-Muñoz, J. C., Terlouw, B. R., van der Hooft, J. J. J., van Santen, J. A., Tracanna, V., Suarez Duran, H. G., Pascal Andreu, V., Selem-Mojica, N., Alanjary, M., Robinson, S. L., Lund, G., Epstein, S. C., Sisto, A. C., Charkoudian, L. K., Collemare, J., Linington, R. G., Weber, T. & Medema, M. H. (2019). MIBiG 2.0: a repository for biosynthetic gene clusters of known function. *Nucleic Acids Research*, *48*(D1), D454–D458.

Keller, N. P. (2015). Translating biosynthetic gene clusters into fungal armor and weaponry. *Nature Chemical Biology*, *11*(9), 671–677.

Keller, N. P. (2019). Fungal secondary metabolism: regulation, function and drug discovery. *Nature Reviews Microbiology*, *17*(3), 167–180.

Khaldi, N., Seifuddin, F. T., Turner, G., Haft, D., Nierman, W. C., Wolfe, K. H. & Fedorova, N. D. (2010). SMURF: genomic mapping of fungal secondary metabolite clusters. *Fungal Genetics and Biology*, *47*(9), 736–741.

Kirk, J. M., Kim, S. O., Inoue, K., Smola, M. J., Lee, D. M., Schertzer, M. D., Wooten, J. S., Baker, A. R., Sprague, D., Collins, D. W. et al. (2018). Functional classification of long non-coding RNAs by k-mer content. *Nature Genetics*, *50*(10), 1474–1482.

Kjærbølling, I., Mortensen, U. H., Vesth, T. & Andersen, M. R. (2019). Strategies to establish the link between biosynthetic gene clusters and secondary metabo-

lites. *Fungal Genetics and Biology*, *130*, 107–121.

Kjærbølling, I., Vesth, T., Frisvad, J. C., Nybo, J. L., Theobald, S., Kildgaard, S., Petersen, T. I., Kuo, A., Sato, A., Lyhne, E. K. et al. (2020). A comparative genomics study of 23 Aspergillus species from section Flavi. *Nature Communications*, *11*(1106).

Kjærbølling, I., Vesth, T. C., Frisvad, J. C., Nybo, J. L., Theobald, S., Kuo, A., Bowyer, P., Matsuda, Y., Mondo, S., Lyhne, E. K. et al. (2018). Linking secondary metabolites to gene clusters through genome sequencing of six diverse Aspergillus species. *Proceedings of the National Academy of Sciences (PNAS)*, *115*(4), E753–E761.

Kloosterman, A. M., Shelton, K. E., van Wezel, G. P., Medema, M. H., Mitchell, D. A. & Gutierrez, M. (2020). RRE-Finder: a Genome-Mining Tool for Class-Independent RiPP Discovery. *mSystems*, *5*(5), e00267–20.

Koch, M., Duigou, T. & Faulon, J.-L. (2019). Reinforcement learning for bioretrosynthesis. *ACS Synthetic Biology*, *9*(1), 157–168.

Kriventseva, E. V., Kuznetsov, D., Tegenfeldt, F., Manni, M., Dias, R., Simão, F. A. & Zdobnov, E. M. (2018). OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. *Nucleic Acids Research*, *47*(D1), D807–D811.

Kumar, A. (2020). Aspergillus nidulans: A Potential Resource of the Production of the Native and Heterologous Enzymes for Industrial Applications. *International Journal of Microbiology*, *2020*.

Liu, J., Gefen, O., Ronin, I., Bar-Meir, M. & Balaban, N. Q. (2020). Effect of tolerance on the evolution of antibiotic resistance under drug combinations. *Science*, *367*(6474), 200–204.

Macheleidt, J., Mattern, D. J., Fischer, J., Netzker, T., Weber, J., Schroeckh, V., Valiante, V. & Brakhage, A. A. (2016). Regulation and role of fungal secondary metabolites. *Annual Review of Genetics*, *50*, 371–392.

Mahmud, M., Kaiser, M. S., Hussain, A. & Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(6), 2063–2079.

Medema, M. H. (2021). The year 2020 in natural product bioinformatics: an overview of the latest tools and databases. *Natural Product Reports*, *38*, 301–306.

Medema, M. H. & Fischbach, M. A. (2015). Computational approaches to natural product discovery. *Nature Chemical Biology*, *11*(9), 639–648.

Medema, M. H., Kottmann, R., Yilmaz, P., Cummings, M., Biggins, J. B., Blin, K., De Bruijn, I., Chooi, Y. H., Claesen, J., Coates, R. C. et al. (2015). Minimum information about a biosynthetic gene cluster. *Nature Chemical Biology*, *11*(9), 625.

Merwin, N. J., Mousa, W. K., Dejong, C. A., Skinnider, M. A., Cannon, M. J., Li, H., Dial, K., Gunabalasingam, M., Johnston, C. & Magarvey, N. A. (2020). DeepRiPP integrates multiomics data to automate discovery of novel ribosomally synthesized natural products. *Proceedings of the National Academy of Sciences (PNAS)*, *117*(1), 371–380.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mircea, I.-G., Bocicor, I. & Czibula, G. (2018). A Reinforcement Learning Based Approach to Multiple Sequence Alignment. In *Soft Computing Applications*, pp. 54–70. Springer International Publishing.

Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G. A., Sonnhammer, E. L. L., Tosatto, S. C. E., Paladin, L., Raj, S., Richardson, L. J., Finn, R. D. & Bateman, A. (2020). Pfam: The protein families database in 2021. *Nucleic Acids Research*, *49*(D1), D412–D419.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill New York.

Montiel, D., Kang, H.-S., Chang, F.-Y., Charlop-Powers, Z. & Brady, S. F. (2015).

Yeast homologous recombination-based promoter engineering for the activation of silent natural product biosynthetic gene clusters. *Proceedings of the National Academy of Sciences*, *112*(29), 8953–8958.

Murphy, K. P. (2021). *Probabilistic Machine Learning: An introduction*. MIT Press.

Neftci, E. O. & Averbeck, B. B. (2019). Reinforcement learning in artificial and biological systems. *Nature Machine Intelligence*, *1*(3), 133–143.

Osbourn, A. (2010). Secondary metabolic gene clusters: evolutionary toolkits for chemical innovation. *Trends in Genetics*, *26*(10), 449–457.

Pevsner, J. (2015). *Bioinformatics and functional genomics*. John Wiley & Sons.

Pickens, L. B., Tang, Y. & Chooi, Y.-H. (2011). Metabolic Engineering for the Production of Natural Products. *Annual Review of Chemical and Biomolecular Engineering*, *2*, 211–236.

Prihoda, D., Maritz, J. M., Klempir, O., Dzamba, D., Woelk, C. H., Hazuda, D. J., Bitton, D. A. & Hannigan, G. D. (2021). The application potential of machine learning and genomics for understanding natural product diversity, chemistry, and therapeutic translatability. *Natural Product Reports*, *38*, 1100–1108.

Rahmat, E. & Kang, Y. (2020). Yeast metabolic engineering for the production of pharmaceutically important secondary metabolites. *Applied Microbiology and Biotechnology*, *104*(11), 4659–4674.

Ramakrishnan, R. K., Singh, J. & Blanchette, M. (2018). RLALIGN: a reinforcement learning approach for multiple sequence alignment. In *In proceedings of the 18th IEEE International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 61–66. IEEE.

Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, *12*(2), 85–94.

Russell, S. & Norvig, P. (2002). *Artificial intelligence: a modern approach*. Pearson.

Santos-Aberturas, J., Chandra, G., Frattaruolo, L., Lacret, R., Pham, T. H., Vior, N. M., Eyles, T. H. & Truman, A. W. (2019). Uncovering the unexplored diversity of thioamidated ribosomal peptides in Actinobacteria using the RiPPER genome mining tool. *Nucleic Acids Research*, *47*(9), 4624–4637.

Sélem-Mojica, N., Aguilar, C., Gutiérrez-García, K., Martínez-Guerrero, C. E. & Barona-Gómez, F. (2019). EvoMining reveals the origin and fate of natural product biosynthetic enzymes. *Microbial Genomics*, *5*(12).

Skinnider, M. A., Dejong, C. A., Rees, P. N., Johnston, C. W., Li, H., Webster, A. L., Wyatt, M. A. & Magarvey, N. A. (2015). Genomes to natural products prediction informatics for secondary metabolomes (PRISM). *Nucleic Acids Research*, *43*(20), 9645–9662.

Skinnider, M. A., Johnston, C. W., Gunabalasingam, M., Merwin, N. J., Kieliszek, A. M., MacLellan, R. J., Li, H., Ranieri, M. R., Webster, A. L., Cao, M. P. et al. (2020). Comprehensive prediction of secondary metabolite structure and biological activity from microbial genome sequences. *Nature Communications*, *11*(1), 1–9.

Song, Y.-J., Ji, D. J., Seo, H., Han, G. B. & Cho, D.-H. (2021). Pairwise heuristic sequence alignment algorithm based on deep reinforcement learning. *IEEE Open Journal of Engineering in Medicine and Biology*, *2*, 36–43.

Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Takeda, I., Umemura, M., Koike, H., Asai, K. & Machida, M. (2014). Motif-independent prediction of a secondary metabolism gene cluster using comparative genomics: application to sequenced genomes of Aspergillus and ten other filamentous fungal species. *DNA Research*, *21*(4), 447–457.

Tietz, J. I., Schwalen, C. J., Patel, P. S., Maxson, T., Blair, P. M., Tai, H.-C., Zakai, U. I. & Mitchell, D. A. (2017). A new genome-mining tool redefines the lasso peptide biosynthetic landscape. *Nature Chemical Biology*, *13*(5), 470.

Umemura, M., Koike, H., Nagano, N., Ishii, T., Kawano, J., Yamane, N., Kozone,

I., Horimoto, K., Shin-ya, K., Asai, K., Yu, J., Bennett, J. W. & Machida, M. (2013). MIDDAS-M: motif-independent de novo detection of secondary metabolite gene clusters through the integration of genome sequencing and transcriptome data. *PLOS ONE*, *8*(12), e84028.

UniProt Consortium (2019). UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, *47*(D1), D506–D515.

UniProt Consortium (2020). UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, *49*(D1), D480–D489.

Vasan, N., Baselga, J. & Hyman, D. M. (2019). A view on drug resistance in cancer. *Nature*, *575*(7782), 299–309.

Vesth, T. C., Brandl, J. & Andersen, M. R. (2016). FunGeneClusterS: predicting fungal gene clusters from genome and transcriptome data. *Synthetic and Systems Biotechnology*, *1*(2), 122–129.

Vesth, T. C., Nybo, J. L., Theobald, S., Frisvad, J. C., Larsen, T. O., Nielsen, K. F., Hoof, J. B., Brandl, J., Salamov, A., Riley, R. et al. (2018). Investigation of inter-and intraspecies variation through genome sequencing of Aspergillus section Nigri. *Nature Genetics*, *50*(12), 1688.

Vining, L. C. (1990). Functions of Secondary Metabolites. *Annual Review of Microbiology*, *44*(1), 395–427.

Vinje, H., Liland, K. H., Almøy, T. & Snipen, L. (2015). Comparing K-mer based methods for improved classification of 16S sequences. *BMC Bioinformatics*, *16*(205).

Walker, A. S. & Clardy, J. (2021). A Machine Learning Bioinformatics Method to Predict Biological Activity from Biosynthetic Gene Clusters. *Journal of Chemical Information and Modeling*, *61*(6), 2560–2571.

Wang, R., Xu, Y. & Liu, B. (2016). Recombination spot identification based on gapped k-mers. *Scientific Reports*, *6*(1), 1–10.

Watkins, C. J. & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3-4), 279–

292.

Wolf, T., Shelest, V., Nath, N. & Shelest, E. (2016). CASSIS and SMIPS: promoter-based prediction of secondary metabolite gene clusters in eukaryotic genomes. *Bioinformatics*, *32*(8), 1138–1143.

Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *In proceedings of the 14th International Conference on Machine Learning (ICML)*, volume 97, pp. 35.

Zhang, X., Elliot, M. A. et al. (2019). Unlocking the trove of metabolic treasures: activating silent biosynthetic gene clusters in bacteria and fungi. *Current Opinion in Microbiology*, *51*, 9–15.