


Data and text mining

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,*}

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

*To whom correspondence should be addressed.

Associate Editor: Zhiyong Lu

Received on September 2, 2021; revised on May 17, 2022; editorial decision on June 20, 2022; accepted on June 23, 2022

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](#) are 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, 2018; Kjærboelling *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) (Kautsar *et al.*, 2020; Keller, 2019). 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 (Kjærboelling *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 (Evdokias *et al.*, 2021; Keller, 2019; Kjærboelling *et al.*, 2020).

Various approaches to obtain candidate BGCs (potential sequence regions encoding biosynthesis of SMs) were previously presented (Chavali and Rhee, 2018), 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 toward 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 and Elliot, 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 (Hannigan *et al.*, 2019; Khaldi *et al.*, 2010), 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, 2019), 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 Materials and methods

The reinforcement learning approach presented here relies on Q-learning (Watkins and Dayan, 1992), an 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; Kjærboelling *et al.*, 2020) and discovery (Almeida *et al.*, 2020; Hannigan *et al.*, 2019). 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 BGC) (Kautsar *et al.*, 2020) repository, and synthetic non-BGC instances created from OrthoDB (Kriventseva *et al.*, 2019) 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 (Almeida *et al.*, 2020; Takeda *et al.*, 2014), while BGC regions tend to present high genomic diversity even among closely related species (Kjærboelling *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 (Aguilar-Pontes *et al.*, 2018; de Vries *et al.*, 2017; 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.

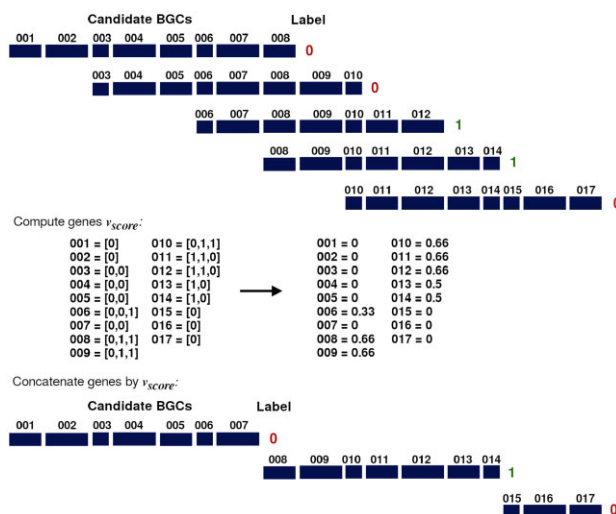


Fig. 1. Computation of majority vote pre-processing for candidate BGCs: regions are merged according to the average score of predicted labels

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 $L = l_0, l_1, \dots, l_m$ 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 \bar{L} . Sequential genes presenting a $v_{\text{score}} \geq 0.5$ are therefore concatenated as positive candidate BGCs, while the other genes with a $v_{\text{score}} < 0.5$ 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 (Sutton and Barto, 2018; Watkins and Dayan, 1992). In Q-learning, the action-value function Q converges toward 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. In addition, 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 $A = \text{keep}, \text{skip}$ 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 $C = c_0, c_1, \dots, c_n$, 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:

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

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:

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

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 $D = d_0, d_1, \dots, d_n$, where n is the number of domains found in g is computed as follows:

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

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

2.3 Integrating functional annotations

BGCs 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 S1.

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 $g_a = \text{skip}$. 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 $g_a = \text{keep}$ for a minimum gene threshold in a candidate BGC G , then genes in G that do not contain protein domains ($D = \emptyset$) will also be assigned an action $g_a = \text{keep}$. 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:

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

where n is the number of domains found in g , and b 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 $g_a = \text{keep}$. 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 $g_a = \text{skip}$. 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

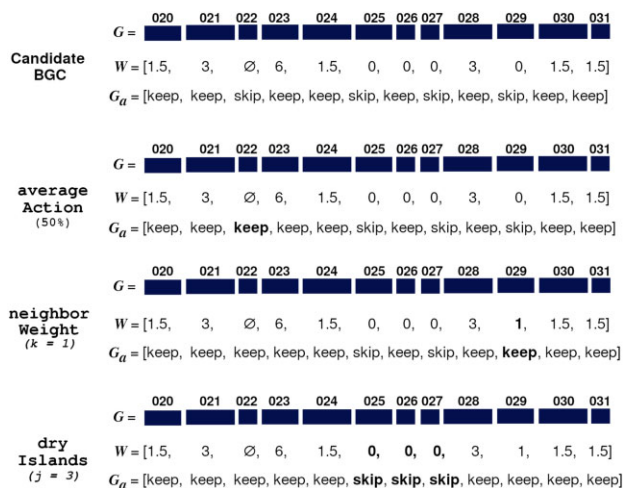


Fig. 2. Example of functional annotation strategies applied to a candidate BGC

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 $\text{pident} \geq 20$ and query coverage $\text{qcov} \geq 10$. 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) both using default parameters, 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

Table 1. Distribution of *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

if it was found to be present in genes annotated with different components.

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 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 $\alpha = 0.01, \gamma = 0.01, \epsilon = 0.01, \text{keepSkip} = 0.5$ yielded the smallest average penalty over 500 episodes. Supplementary Tables S2 and S3 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 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

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

Model	Gene metrics			Cluster metrics			Average F-m	% gold-std. genes	
	P	R	F-m	P	R	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%

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 3A 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 3B 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 3B also appear in true positive BGC genes, as opposed to Figure 3A 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 3B, more than 50% are associated to BGC component roles, and found immediately after true positive BGC genes. Non-BGC genes shown in Figure 3A 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 S4 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 S4.

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 (Kjærboelling 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 S4, 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 (Drott et al., 2020; Inglis et al., 2013). To further evaluate the reproducibility of the proposed reinforcement learning approach, we processed the *A.nidulans* genome considering 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-

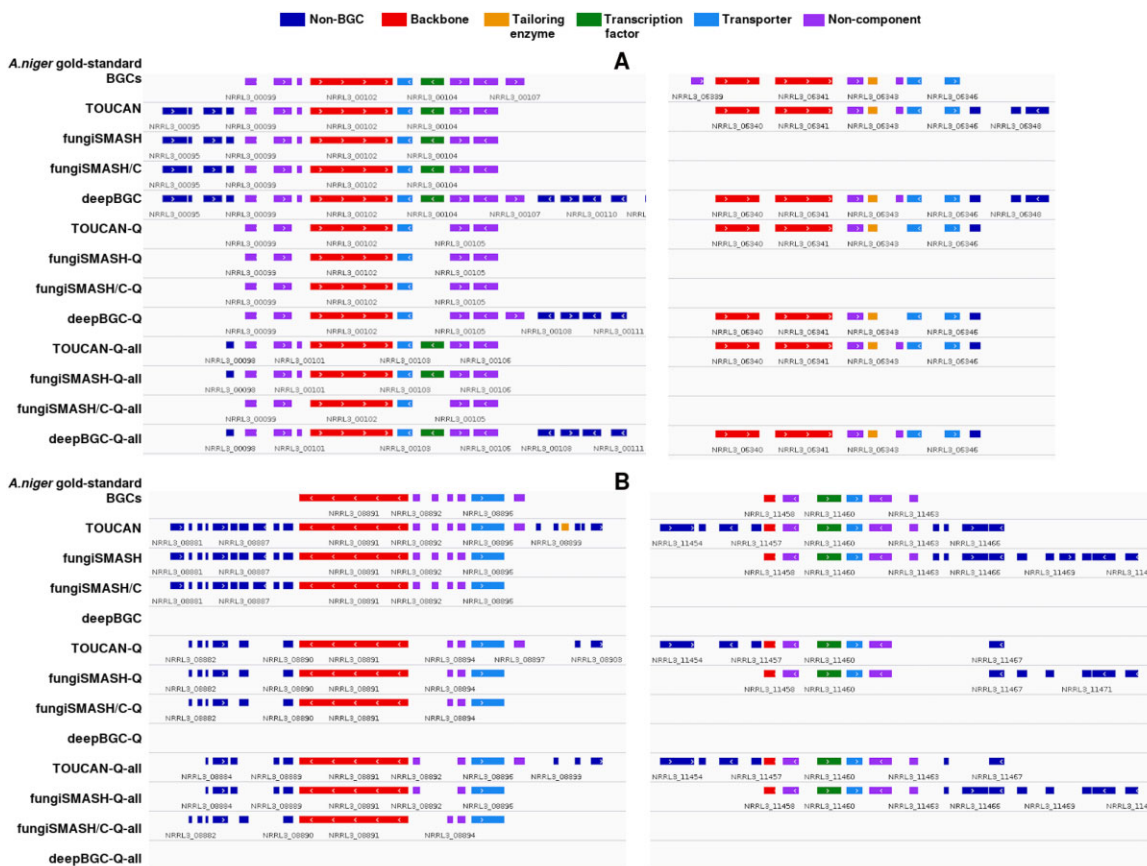


Fig. 3. Comparison between gold-standard and candidate BGC composition for four *A.niger* clusters. Non-BGC genes are shown in dark blue. (A) Candidate BGCs for which the reinforcement learning agent correctly skipped most non-BGC genes compared to their polyketide (left) and fatty acid (right) gold 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 (A color version of this figure appears in the online version of this article.)

Table 3. Distribution of *A.nidulans* pseudo BGC components in 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

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 S5](#). The distribution of component pseudo-annotations found in the training data and gold-standard genes for *A.nidulans* is shown in [Table 3](#).

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](#).

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

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

Model	Gene metrics			Cluster metrics			Average F-m	% gold genes	
	P	R	F-m	P	R	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%

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 a 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 S1](#). Similarly to *A.niger*, [Supplementary Figure S1A](#) 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 S1B](#), the agent kept most non-BGC genes, potentially resulting in overpredicted boundaries. Approximately 50% of protein domains from non-BGC genes in [Supplementary Figure S1B](#) were associated to pseudo-functional annotations in *A.nidulans*, while only 20% of domains from non-BGC genes in [Supplementary Figure S1A](#) 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 ([Cimermanic et al., 2014](#); [Hannigan et al., 2019](#); [Khaldi et al., 2010](#)).

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. In addition, a preliminary analysis, shown in [Supplementary Figure S2](#), 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 toward supporting the development of robust *in silico* approaches for fungal BGC discovery.

Acknowledgements

The authors 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).

Conflict of Interest: none declared.

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.

References

- Aguilar-Pontes, M. *et al.* (2018) The gold-standard genome of *Aspergillus niger* NRRL 3 enables a detailed view of the diversity of sugar catabolism in fungi. *Stud. Mycol.*, **91**, 61–78.
- Almeida, H. *et al.* (2019) Supporting supervised learning in fungal biosynthetic gene cluster discovery: new benchmark datasets. In: *Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, San Diego, CA, November 18–21 2019. IEEE, pp. 1280–1287.
- Almeida, H. *et al.* (2020) TOUCAN: a framework for fungal biosynthetic gene cluster discovery. *NAR Genom. Bioinform.*, **2**, lqaa098.
- Angermueller, C. *et al.* (2020) Model-based reinforcement learning for biological sequence design. In: *Proceedings of the 8th International Conference on Learning Representations (ICLR), Virtual Conference, Apr 26–May 1st, 2020*.
- Blin, K. *et al.* (2021) antiSMASH 6.0: improving cluster detection and comparison capabilities. *Nucleic Acids Res.*, **49**, W29–W35.
- Chavali, A.K. and Rhee, S.Y. (2018) Bioinformatics tools for the identification of gene clusters that biosynthesize specialized metabolites. *Brief. Bioinform.*, **19**, 1022–1034.
- Cimermancic, P. *et al.* (2014) Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. *Cell*, **158**, 412–421.
- de Vries, R.P. *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.
- Drott, M. *et al.* (2020) Diversity of secondary metabolism in *Aspergillus nidulans* clinical isolates. *mSphere*, **5**, e00153-20.
- El-Gebali, S. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.
- Evdokias, G. *et al.* (2021) Identification of a novel biosynthetic gene cluster in *Aspergillus niger* using comparative genomics. *JoF*, **7**, 374.
- Gottipati, S.K. *et al.* (2020) Learning to navigate the synthetically accessible chemical space using reinforcement learning. In: *Proceedings of the 37th International Conference on Machine Learning (ICML), Virtual Conference, July 13–18 2020*, pp. 3668–3679.
- Hannigan, G.D. *et al.* (2019) A deep learning genome-mining strategy for biosynthetic gene cluster prediction. *Nucleic Acids Res.*, **47**, e110.
- Imani, M. and Braga-Neto, U.M. (2019) Control of gene regulatory networks using Bayesian inverse reinforcement learning. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **16**, 1250–1261.
- Inglis, D.O. *et al.* (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.
- Kautsar, S.A. *et al.* (2020) MIBiG 2.0: a repository for biosynthetic gene clusters of known function. *Nucleic Acids Res.*, **48**, D454–D458.
- Keller, N.P. (2015) Translating biosynthetic gene clusters into fungal armor and weaponry. *Nat. Chem. Biol.*, **11**, 671–677.
- Keller, N.P. (2019) Fungal secondary metabolism: regulation, function and drug discovery. *Nat. Rev. Microbiol.*, **17**, 167–180.
- Khaldi, N. *et al.* (2010) SMURF: genomic mapping of fungal secondary metabolite clusters. *Fungal Genet. Biol.*, **47**, 736–741.
- Kjærboelling, I. *et al.* (2019) Strategies to establish the link between biosynthetic gene clusters and secondary metabolites. *Fungal Genet. Biol.*, **130**, 107–121.
- Kjærboelling, I. *et al.* (2020) A comparative genomics study of 23 *Aspergillus* species from section Flavi. *Nat. Commun.*, **11**, 1106.
- Krivtseva, E.V. *et al.* (2019) 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.
- Mahmud, M. *et al.* (2018) Applications of deep learning and reinforcement learning to biological data. *IEEE Trans. Neural Netw. Learn. Syst.*, **29**, 2063–2079.
- Mircea, I.-G. *et al.* (2018) A reinforcement learning based approach to multiple sequence alignment. In: *Proceedings of the 7th International Workshop Soft Computing Applications (SOFA 2016)*, Arad, Romania, August 24–26, 2016, Vol. 1. Springer, pp. 54–70.
- Montiel, D. *et al.* (2015) Yeast homologous recombination-based promoter engineering for the activation of silent natural product biosynthetic gene clusters. *Proc. Natl. Acad. Sci. USA*, **112**, 8953–8958.
- Neftci, E.O. and Averbek, B.B. (2019) Reinforcement learning in artificial and biological systems. *Nat. Mach. Intell.*, **1**, 133–143.
- Sutton, R.S. and Barto, A.G. (2018) *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.
- Takeda, I. *et al.* (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.
- Watkins, C.J. and Dayan, P. (1992) Q-learning. *Mach. Learn.*, **8**, 279–292.
- Wolf, T. *et al.* (2016) CASSIS and SMIPS: promoter-based prediction of secondary metabolite gene clusters in eukaryotic genomes. *Bioinformatics*, **32**, 1138–1143.
- Zhang, X. and Elliot, M.A. (2019) Unlocking the trove of metabolic treasures: activating silent biosynthetic gene clusters in bacteria and fungi. *Curr. Opin. Microbiol.*, **51**, 9–15.