Published online 10 August 2019

Nucleic Acids Research, 2019, Vol. 47, No. 18 ***e110***
doi: 10.1093/nar/gkz654

A deep learning genome-mining strategy for biosynthetic gene cluster prediction

Geoffrey D. Hannigan[1,†], David Prihoda[2,3,†], Andrej Palicka[4], Jindrich Soukup[5], Ondrej Klempir[6], Lena Rampula[7], Jindrich Durcak[6], Michael Wurst[4], Jakub Kotowski[4], Dan Chang[8], Rurun Wang[1], Grazia Piizzi[1], Gergely Temesi[6], Daria J. Hazuda[1,9], Christopher H. Woelk[1,*,‡] and Danny A. Bitton[6,*,‡]

[1]Exploratory Science Center, Merck & Co., Inc., Cambridge, Massachusetts, USA, [2]Big Data Solutions, MSD Czech Republic s.r.o., Prague, Czech Republic, [3]Department of Informatics and Chemistry, Faculty of Chemical Technology, University of Chemistry and Technology, Prague, Czech Republic, [4]AI & Big Data Analytics, MSD Czech Republic s.r.o., Prague, Czech Republic, [5]Data Science, MSD Czech Republic s.r.o., Prague, Czech Republic, [6]Bioinformatics & Cheminformatics Solutions, MSD Czech Republic s.r.o., Prague, Czech Republic, [7]NLP, MSD Czech Republic s.r.o., Prague, Czech Republic, [8]Genetics & Pharmacogenomics, Merck & Co., Inc., Boston, MA, USA and [9]Infectious Diseases and Vaccine Research, MRL, Merck & Co., Inc., West Point, PA, USA

Received January 11, 2019; Revised July 09, 2019; Editorial Decision July 16, 2019; Accepted August 08, 2019

ABSTRACT

Natural products represent a rich reservoir of small molecule drug candidates utilized as antimicrobial drugs, anticancer therapies, and immunomodulatory agents. These molecules are microbial secondary metabolites synthesized by co-localized genes termed Biosynthetic Gene Clusters (BGCs). The increase in full microbial genomes and similar resources has led to development of BGC prediction algorithms, although their precision and ability to identify novel BGC classes could be improved. Here we present a deep learning strategy (DeepBGC) that offers reduced false positive rates in BGC identification and an improved ability to extrapolate and identify novel BGC classes compared to existing machine-learning tools. We supplemented this with random forest classifiers that accurately predicted BGC product classes and potential chemical activity. Application of DeepBGC to bacterial genomes uncovered previously undetectable putative BGCs that may code for natural products with novel biologic activities. The improved accuracy and classification ability of DeepBGC represents a major addition to *in-silico* BGC identification.

INTRODUCTION

Natural products are chemical compounds that are found in nature and produced by living organisms. They represent a rich reservoir of drug candidates that have proven utility across multiple therapeutic areas. Between 1981 and 2014, one third (32%) of FDA approved small molecule drugs were either unmodified natural products (6%) or natural product derivatives (26%) (1). These include multiple classes of antibacterials, as well as oncology drugs, diabetes drugs, hypocholesterolemic drugs and immunomodulatory agents (1,2). The global rise in antibiotic resistance (3,4), the increased promise of immunomodulatory agents in cancer treatment (5), and the continued need for development of new drugs across novel and complex biology is contingent upon the identification of structurally diverse bioactive compounds (6–8).

Early genetic work in the field of natural product discovery showed that bioactive molecules are microbial secondary metabolites whose synthesis is primarily orchestrated by genomically co-localized genes termed Biosynthetic Gene Clusters (BGCs) (9–11). While these early insights were born out of forward genetic approaches (progressing from phenotype to sequence), the advent of next generation sequencing technologies and genomic approaches provided opportunities for reverse genetic approaches (progression from sequence to phenotype) in BGC discovery, synthesis, and characterization (2). The surge of

*To whom correspondence should be addressed. Tel: +420 277026475; Email: danny.bitton@merck.com
Correspondence may also be addressed to Christopher H. Woelk. Tel: +1 617 995 9010; Email: christopher.woelk@merck.com
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
‡Equal senior author contribution.

microbial genomic resources, including completed genome sequences of cultured and uncultured organisms, has enabled a paradigm shift in how computational methods have been used in natural product drug candidate discovery.

Numerous bioinformatics tools have leveraged the increasingly abundant genomic data to facilitate natural product genome mining (12). Early approaches implemented simple BGC reference alignment techniques using programs like BLAST (13), and were often paired with manual curation. Rule-based algorithms (14,15) improved on their predecessors by using human-coded ('hard coded') rule sets to define BGCs based on their similarity to reference genes and protein domain composition. While some recent approaches have continued to employ these 'reference-based' techniques, other algorithmic advances have embraced more generalizable machine learning approaches that provide a greater ability to discover new BGC genomic elements. One such widely used machine learning approach named ClusterFinder (16) employs a Hidden Markov Model (HMM) instead of the multiple sequence alignment based profile-HMM (17) methods seen in other approaches such as AntiSMASH (ANTIbiotics & Secondary Metabolite Analysis SHell) (14) and PRISM (18).

While they have been effective, HMMs like ClusterFinder do not preserve (i.e. remember) position dependency effects between distant entities or order information (19–21). This means HMM-based tools are unable to capture higher order information among entities (19–21), thus limiting their ability to detect BGCs. We addressed this algorithmic limitation by implementing a deep learning approach using Recurrent Neural Networks (RNNs) and vector representations of protein family (Pfam) (22) domains which together, unlike HMMs, are capable of intrinsically sensing short- and long-term dependency effects between adjacent and distant genomic entities (23). This implementation yielded performance higher than another leading algorithm (ClusterFinder), including improved BGC detection accuracy from genome sequences and improved ability to identify BGCs of novel classes.

Here, we introduce DeepBGC, a novel utilization of deep learning and natural language processing (NLP) strategy for improved identification of BGCs in bacterial genomes (Figure 1). DeepBGC employs a Bidirectional Long Short-Term Memory (BiLSTM) RNN (24,25) and a word2vec-like word embedding skip-gram neural network we call pfam2vec. Compared to Clusterfinder (16), DeepBGC improves detection of BGCs of known classes from bacterial genomes, and harnesses great potential to detect novel classes of BGCs. We supplement this with generic random forest classifiers that enable classifications of BGCs based on the product class and molecular activity of the compounds. We applied DeepBGC to bacterial reference genomes to identify BGC candidates coding for molecules with putative antibiotic activity that could not be identified using other existing methods. In addition to bacterial reference genomes, we expect this approach to be important in microbiome metagenomic analyses, in which the improved BGC detection may empower new functional insights. To facilitate these and other analytical applications, DeepBGC is available at https://github.com/Merck/deepbgc.

MATERIALS AND METHODS

Open reading frame identification

Open reading frames were predicted in 3376 reference bacterial genomes (26) using Prodigal (27) version 2.6.3 with default parameters. All other sequences were downloaded with annotations and gene locations.

Protein family identification

Protein family domains were identified using HMMER (17), hmmscan version 3.1b2, and the Pfam database version 31 (22). This Pfam database was used for all applications except for the original ClusterFinder algorithm, where it was preserved at legacy version 27. Hmmscan tabular output was filtered using BioPython SearchIO module (version 1.70) to preserve only highest scoring Pfam regions with *e*-value <0.01. The resulting list of Pfam domains was sorted by the gene and the domain start location.

Pfam2vec implementation

Pfam2vec embedding was generated using the original word2vec implementation (28) wrapped in the word2vec python package (version 0.9.2). After bootstrap evaluation, the following hyper-parameters were chosen: 100 dimensions, 8 training iterations and skipgram architecture. The training corpus consisted of 3376 documents (bacteria) and 23 425 967 words (15 686 unique Pfam identifiers). Each document contained a space-separated list of Pfam identifiers representing all Pfam domains of a specific bacterial genome maintained in their genomic order. To evaluate the pfam2vec embedding, cosine similarity of domain vectors were used in comparison with domain membership to one of 604 Pfam superfamilies (clans) from the Pfam database version 31 (22). First, cosine similarity of all (non-identity) pairs of domains from all Pfam superfamilies was compared (using Wilcoxon rank-sum test) to cosine similarity of all pairs from Pfam superfamilies that were randomly shuffled. Second, the same calculation was performed with pfam2vec vectors replaced by random numeric vectors. Third, Levenshtein distance of Pfam domain descriptions was calculated for each pfam2vec vector and its nearest neighbor by cosine similarity and compared (using Wilcoxon rank-sum test) to the description similarity of each pfam2vec vector and a randomly selected pfam2vec vector. Finally, average pfam2vec vector representation of each BGC from the MIBiG set was calculated as an average of its list of pfam2vec vectors, with averages computed separately for each dimension. This BGC vector representation was reduced to two dimensions for visualization using the scikit-learn manifold.tSNE package with cosine metric, random initialization and default perplexity of 30.

DeepBGC training set

DeepBGC was trained on a subset of the original ClusterFinder positive training set and on a negative set generated based on similar principles as the ClusterFinder negative training set. To collect the original positive training set, 681 accession IDs were obtained from the ClusterFinder

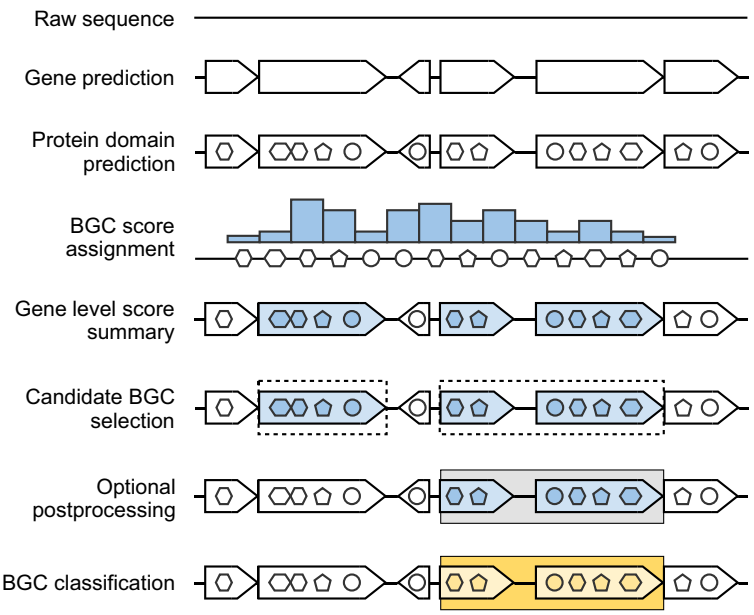

Figure 1. Overview of the deep learning strategy for detection of Biosynthetic Gene Clusters in bacterial genomes. (From top to bottom) raw genomic sequences (solid line) are used for gene (arrowhead structures) prediction by Prodigal (27). Pfam domains (circles, penta- and hexagons) are assigned to each ORF using hmmscan (17). The BiLSTM outputs classification score (blue bars) for each domain. Domain scores are summarized across genes, which are selected accordingly (blue arrowhead structures). Consecutive candidate BGC genes are assembled to putative BGCs (dashed rectangles). An optional post-processing step allowed merging of neighboring BGCs based on the presence of a known biosynthetic pathway, minimum cluster length, and gaps between adjacent BGCs (gray rectangles). BGCs were classified using random forest classifiers based on compound class and molecular activity (yellow rectangles).

supplementary table and searched on NCBI, which returned 617 sequences. To generate the negative set, we collected and preprocessed a public reference set of 3376 bacteria from EMBL-EBI. For each reference bacterium, regions similar to known MIBiG BGC (version 1.3) were removed (Blastn (13) with 95% threshold). To generate a single negative sample, a random reference bacterium and a random sample from the positive ClusterFinder set were selected. Each gene in the positive sample was replaced with a random gene from the reference bacteria, while considering only 1% of genes that were most similar in number of Pfam domains. In total, three samples were generated from each reference bacteria, producing 10 128 negative samples.

DeepBGC implementation

The BiLSTM model was implemented using Keras (version 2.1.6) with TensorFlow backend (version 1.6.0). The architecture consisted of a single Keras Sequential model with two layers. First, the model contained a stateful BiLSTM layer with 128 units and dropout of 0.2. Second, the model contained a time-distributed dense layer with sigmoid activation and 1 output unit. The input was a sequence of Pfam domains represented by 102-dimensional vectors consisting of the 100-dimensional pfam2vec embedding and two binary flags marking domains found at the beginning or end of proteins. The output was a sequence of values between 0 and 1 representing the prediction score of given domain to be part of a BGC. In each training epoch, all positive and negative samples were shuffled randomly and concatenated to create an artificial genome. Training was configured with 256 timesteps and a batch size of 64. Thus, the training sequence of each epoch was separated into 64 subsequences, each trained in parallel in batches of 256 timesteps, processing a single training vector at each timestep. The final model was trained for 328 epochs using the Adam optimizer with learning rate of 1e–4 and weighted binary cross-entropy loss function (weights are inversely proportional to number of positive and negative samples in our training dataset, therefore giving more weight to positive samples).

To obtain BGC regions used for BGC-level analysis, first, predicted scores were averaged in each gene, BGC genes were selected using any given threshold and consecutive BGC genes were merged. Optionally, postprocessed BGC regions were created by applying filters defined in Cimermancic *et al.* (16): merging BGC regions at most one gene apart and filtering out regions with less than 2000 nucleotides and regions with no known biosynthetic domains from the current list of 133 domains published in the ClusterFinder (16) submodule of antiSMASH (14).

DeepBGC validation

The primary evaluation metric published in Cimermancic *et al.*, (16) was a ROC curve based on 10 reference genomes that are fully annotated with BGC and non-BGC regions. The associated genomes were retrieved based on the list of gene loci provided in the supplementary table (16). By querying the gene loci on NCBI, 9 out of 10 of the original genomes were obtained. The original *Streptomyces roseosporus* genome could not be retrieved. In three of the genomes the genes were not found in a single contig, but in multiple contigs (two for *Streptomyces ghanaensis*, two for *Streptomyces* sp. AA4 and three for *Streptomyces*

sp. C). The sequences were updated since the release of the paper, which resulted in a location shift of 7% of genes and removal of 10 BGC genes from the *Streptomyces pristinaespiralis* genome.

To ensure an accurate comparison between DeepBGC and ClusterFinder, DeepBGC was trained with the 617 positive and 10 128 negative samples described above. Since this dataset is artificially created it lacks some features of real-world data (potential nonrandom distribution of BGC across the genome, real distribution and order of the genes etc.). Therefore, there was no guarantee that model trained on our artificial training set will perform with the same accuracy on real genomes. To address that we chose to tune hyperparameters using a subset of real world data. To avoid the reduction of our validation set and prevent a leakage from training set to validation set we chose a 'bootstrap' approach. It consists of creating multiple models, where each model will utilize a small part of validation set for hyperparameter tuning and the rest for testing. Averaging the accuracy of those multiple models is proven to converge to an unbiased estimation.

During hyperparameter tuning we considered following parameters and values: learning rate (0.001, 0.0001), number of pfam2vec training iterations (4,8,16,32), number of pfam2vec dimensions (50, 100, 200) and positive training sample weight (1, 16.415). Keras EarlyStopping method was used with minimum delta of 0.0005 on validation ROC AUC within 100 epochs. We realized that majority of the learned models, regardless of the validation genome selection, preferred following parameters: 0.0001 for learning rate, 8 pfam2vec training iterations, 100 pfam2vec dimensions and positive sample weight based on the negative/positive ratio of 16.415.

BGC-level coverage evaluation was performed on the test set of the first bootstrap split. Predictions of each model were converted into BGC regions (with and without postprocessing) using the method defined above. True BGC coverage of each model was calculated for each annotated true BGC region as the fraction of the region that was covered by all its overlapping predicted BGC regions of given model. A BGC was considered detected when its coverage was above a given coverage threshold. Coverage distribution was calculated by evaluating all coverage thresholds from 0% up to 100% in steps of 0.1%. Next, each predicted BGC region is marked as true positive if it overlaps with a true BGC region and as a false positive if it does not. Finally, BGC-level precision was calculated as the number of true positive regions divided by the total number of predicted regions.

The secondary evaluation metric published in Cimermancic *et al.* (16) was a True Positive Rate (TPR) evaluation based on 65 BGCs in their genomic context of 6 genomes. All BGC locations were provided along with names of source bacteria in the supplementary table in Cimermancic *et al.* (16). First, genomes were found on NCBI by manually querying the organism names. Second, BGC start and end locations were validated to match with start and end locations of annotated genes present in the retrieved genomes. Finally, we obtained BGC predictions using DeepBGC, original ClusterFinder and retrained ClusterFinder. The original evaluation metric was based on calculating TPR in terms of fraction of BGCs detected with median ClusterFinder HMM prediction >0.4 threshold. After the inspection of ClusterFinder predictions it was found that >42% of the domains outside the defined BGC regions were detected above the given threshold, relying on heavy further postprocessing and manual annotation to filter out false positives. Therefore, the sequences were evaluated using a ROC curve, which considers all unannotated regions to be negative, producing a lower bound of the AUC value which is unbiased to either of the two models.

To perform cross validation and leave-class-out validation we obtained all 1406 BGC samples from MIBiG (version 1.3) and our negative set of 10128 samples. Each sample was represented as a list of Pfam domain identifiers. For cross validation, samples were randomly distributed into 10 splits. In each of the 10 cross-validation folds, models were trained on nine splits and evaluated on one split. Training and testing samples were shuffled and concatenated to create artificial genomes. An average ROC was computed by concatenating all test split predictions. In leave-class-out validation, 1003 samples from six non-hybrid classes (Polyketide, NRP, RiPP, Saccharide, Terpene, Alkaloid) were selected. For each class, the models were trained on all other classes and random two thirds of negative samples. Thereafter, models were tested on the given class (upsampled to 500 samples by sampling with replacement) and the remaining third of negative samples. Again, training and testing samples were shuffled and concatenated to create artificial genomes. This was performed three times for each class with different random splits and random shuffles to minimize the influence of any random initialization. An average ROC was computed by concatenating all test predictions.

ClusterFinder implementation

ClusterFinder predictions were produced using antiSMASH (version 4.1.0) with ClusterFinder enabled and with default parameters. Raw prediction scores for each Pfam domain were parsed from the final Genbank output files. These scores were used to produce domain-level ROC curves. Raw and postprocessed BGC regions used for BGC-level analysis were obtained using gene-level score averaging as in the DeepBGC implementation described above.

To retrain the model for cross validation and leave-class-out validation, the ClusterFinder HMM was reimplemented using the hmmlearn python module (version 0.2.0). The transition and starting probability matrices of the original model were used. The emission probability matrix was re-computed using the new positive and negative training set preprocessed with Pfam database version 31 (same as the DeepBGC model).

Random forest Multi-Label product classification

Biosynthetic product class and activity training data were obtained from the MIBiG database (version 1.3) in JSON format. Product classes were extracted from the 'biosyn_class' field, producing 1355 labelled training samples. Product activities were extracted from the 'chem_act' field of each compound in the 'compounds' field, excluding BGCs with no known product activities, producing

370 training samples. A separate random forest classifier was trained for both domains using the scikit-learn python module (version 0.19.1). The classes were predicted using multi-label classification, where each sample is labelled with a binary vector representing presence of zero or more classes. Global feature importance was obtained using native scikit-learn method, class-specific feature importance was calculated by training a separate temporary random forest classifier for each class. To evaluate model performance, 5-fold cross validation was used, producing 5 sets of real-valued prediction scores which were merged and compared with expected output at different thresholds to produce a ROC curve. A confusion matrix was generated by treating each occurring combination of biosynthetic classes as a single separate hybrid class. To produce the antiSMASH ROC curve, all MIBiG BGC genbank files were processed through antiSMASH (version 4.1.0) with default parameters. The predicted product types were parsed from the text output files and mapped to the more high-level MIBiG classification. In ambiguous cases where the antiSMASH product type could not be mapped to a single MIBiG class, BGCs of given product type were discarded from the evaluation of both models. For antiSMASH, the resulting class predictions were used to generate a ROC curve with one point, since a classification score is not available.

RESULTS

Curation yielded diverse training & validation datasets

BGC prediction is a classification task requiring the use of labelled BGC and non-BGC sequences (a 'positive' and 'negative' set, respectively) to train and validate the classifier. To ensure adequate comparison between the existing HMM approach and our deep learning strategy, we trained and validated our model using a training set similar to that used in Cimermancic *et al.* (16). We built our positive training set by retrieving 617 out of 667 published labeled BGCs from Cimermancic *et al.* (16) (Supplementary Figure S1, Supplementary Table S1). We constructed a negative training set of 10128 random gene clusters based on similar principles to those described in Cimermancic *et al.* (16). We additionally retrieved a second, supplementary dataset consisting of 1406 BGCs found in the Minimum Information about a Biosynthetic Gene cluster (MIBiG) database (29), which were used for 10-fold cross-validation, leave-class-out validation and training random forest classifiers (Supplementary Figure S1, Supplementary Table S2).

In addition to the BGC sequence datasets, we utilized whole bacterial genomes that had been manually annotated with BGC and non-BGC regions. We used the set of nine bacterial genomes that contained 291 manually annotated BGCs from Cimermancic *et al.* (16) and used them for validation, hyperparameter tuning, and testing (Supplementary Figure S1, Supplementary Table S3). We also used a second set of 65 experimentally validated BGCs in six bacterial genomes (termed the validated Cimermancic *et al.* (16) set, Supplementary Figure S1, Supplementary Table S4), which allowed for a supplemental model testing. Additionally, we retrieved a corpus of 3376 unannotated bacterial genomes that we used for generating negative samples, for pfam2vec corpus curation and for explorative application of DeepBGC to detect novel BGCs (Supplementary Figure S1, Supplementary Table S5).

Pfam2Vec captures biological signal for DeepBGC input

One major challenge in BGC identification was defining informative genomic input for the algorithm. Input sequences of biological entities can be represented at different genomic levels including nucleotides, amino acids, and genes. Of these options, sequential protein family (Pfam) domain representations have been highly informative for BGC identification (16,30) because they represent functional elements within genes. We extended this approach by converting sequences of Pfam domain identifiers to numeric vector representations using the word2vec algorithm (28). The resultant vectors of real numbers encapsulated domain properties based on their genomic context, allowing us to leverage contextual (and below we show functional) similarities between Pfam domains and BGCs.

We validated the ability of pfam2vec to produce functionally meaningful numeric representations of Pfam domains within genomes by calculating the average vector cosine similarity between members of superfamilies. The cosine similarities within the same Pfam superfamilies (clans) were significantly higher than the similarities between random domain vector pairs ($P < 2.2 \times 10^{-16}$, Supplementary Figure S2a). The average cosine similarity between superfamily pairs and random pairs was centered around 0 when random representation vectors were used ($P = 0.09$, Supplementary Figure S2b). The known domain functional annotations of the nearest domain vector pairs were more similar compared to random pairs ($P < 2.2 \times 10^{-16}$, Supplementary Figure S2c) and N- and C-terminal domain pairs appeared to be more similar in vector space compared to other domains (Supplementary Table S6). These findings suggested that pfam2vec produced functionally meaningful numeric representations of Pfam domain sequences by reflecting known superfamily similarities.

We further confirmed the functional relevance of pfam2vec vectors by evaluating their ability to discriminate BGCs by their Pfam repertoires. We accomplished this by condensing the many Pfam domains within each given BGC into a single representative BGC vector using two alternative approaches. In the first approach we created representative BGC vectors by averaging the pfam2vec vector values, and in the second approach we created a binary vector indicating the presence of each specific domain (domain set vector representation). We assessed the biological relevance of these two approaches by means of a t-Distributed Stochastic Neighbor Embedding (t-SNE) of all BGCs from the MIBiG database (29), and showed that both approaches preserved similarity between BGC subclasses (Supplementary Figure S3a and b). Separation of BGCs was reduced when assessing their taxonomic discriminative abilities, further suggesting that BGCs were largely defined by their functional domain architecture and less by their bacterial species (Supplementary Figure S3c and d). We also note that while this is true at the domain level, it does not reflect taxon specificity at the chemical class level, which we would expect increased taxon specificity (16). While domain

set vectors meaningfully represented BGCs, they could not be used for individual domain representations that could be used in sequence as input to a RNN model. On the contrary pfam2vec vectors provided condensed and meaningful representation of individual domains and therefore represented a functionally relevant input for RNN that could enhance BGC identification.

Unique model architecture & bootstrapping improves BGC prediction

The DeepBGC BiLSTM neural network was comprised of three layers: the input layer, the BiLSTM unit, and the output layer (Figure 2). The *input layer* encoded a sequence of numerical vectors representing Pfam domains in their genomic order. The *BiLSTM layer* consisted of forward and backward LSTM network layers, each consisting of a basic LSTM unit (a memory cell) with a 128-dimensional hidden state vector. The memory cell was fed with a single input vector as well as the cell's state from the previous time step in the genome. The *output* from all LSTM memory cells was processed through a single fully connected output layer with a sigmoid activation function. This yielded a single value for each genomic Pfam entity, which represented BGC classification score for that Pfam domain. This model was trained using our positive (labeled BGCs) and negative (random gene clusters) training samples, which were converted into their respective sequences of pfam2vec vectors. Positive and negative samples were repeatedly shuffled and concatenated to simulate real genomic context in which BGCs were scattered randomly throughout the genome and surrounded by non-BGC sequences.

To compensate for the lack of a large independent validation set that could be used for optimizing model architecture, input features, and hyperparameters, we employed a bootstrap sampling technique. Because our model was trained on artificially created genomes, we bootstrapped a real-world dataset to enable hyperparameter tuning and to simultaneously prevent data leakage that would bias our estimation of accuracy in our algorithm. We performed bootstrapping using our nine manually BGC-annotated whole-genome dataset, wherein each of five iterations we randomly selected two genomes randomly with replacement for hyperparameter validation and used the remaining genomes for testing (Supplementary Figure S1). We obtained an averaged Receiver Operating Characteristic (ROC) curve by combining the 5 iteration test set predictions, which revealed an improvement in precision and recall compared to the original ClusterFinder HMM model as well as to ClusterFinder HMM retrained with up-to-date data (AUC: 0.946, 0.837, 0.912, respectively, average precision: 0.75, 0.28, 0.63, respectively, Figure 3A and B).

DeepBGC accuracy outperforms existing machine learning model

We formally evaluated the performance of the DeepBGC model and compared it to the ClusterFinder model by testing its ability to (i) accurately identify BGC positions within whole bacterial genomes, (ii) discriminate between BGCs and artificially created non-BGC sequences and (iii) identify 'novel' BGC classes to which it has not been exposed. First, to address whether DeepBGC could accurately identify BGC positions within whole genomes, we evaluated the model positional accuracy using our 65 experimentally validated BGC set from six bacterial genomes (16). This revealed an improved performance of DeepBGC (AUC = 0.923) over ClusterFinder (AUC = 0.847, Figure 3C). Second, we used 10-fold cross validation to evaluate whether our final DeepBGC model could better discriminate between BGC and artificially created non-BGC sequences compared to the existing ClusterFinder algorithm. We used BGCs in the MIBiG database (29) as a positive set and the random gene cluster negative set. Both sets were randomly distributed across 10 bins, with 9 bins used for training the model (with the optimal settings) and 1 used for testing. DeepBGC (AUC = 0.984) outperformed ClusterFinder (AUC = 0.936) in differentiating between positive and negative samples (Supplementary Figure S4). Third, to evaluate DeepBGC's ability to identify 'novel' BGCs, we carried out a 'Leave-class-out' validation in which we assessed the models' abilities to identify a single BGC class in the test set that was intentionally omitted from the training set. DeepBGC yielded more accurate identification of classes it had not encountered (AUC = 0.946) compared to ClusterFinder (AUC = 0.865; Figure 3D, Supplementary Figure S5). Overall DeepBGC yielded improved BGC identification accuracy, and was better able to accurately extrapolate to identify BGCs of classes it had not encountered previously.

DeepBGC yields improved precision & BGC coverage

The most common use case for BGC identification models such as DeepBGC is locating BGCs within bacterial genomes. It was therefore important for us to validate DeepBGC as having improved positional predictive accuracy of BGCs, in addition to an improved precision and recall. We achieved this by comparing the accuracy of DeepBGC and ClusterFinder BGC detection in a subset of manually annotated bacterial genomes. To account for potential differential impacts of chosen BGC score thresholds, we applied two distinct thresholds based on the respective domain-level ROCs, a stringent domain level of 10% false positive rate (FPR) as well as a lenient cutoff of 80% true positive rate (TPR, Supplementary Table S7, Supplementary Figure S6a and b).

With the stringent cutoff of 10% FPR, the number of BGCs predicted by DeepBGC was consistently higher than those predicted by ClusterFinder, regardless of the BGC coverage threshold (Figure 4A and B). ClusterFinder displayed a sharp decline in the number of predicted BGCs as the coverage threshold increased (Figure 4B), indicating that BGCs predicted by this approach were typically short (Supplementary Figure S7, Supplementary Table S8). The overall precision at the BGC level remained comparable between the two models (precision = 34%, 26%, Figure 4C). Under a more lenient TPR threshold of 80%, ClusterFinder predicted more BGCs than DeepBGC when coverage threshold remained <68% (Figure 4D and E). But we found that ClusterFinder predictions were of low precision and were composed of many false positives (Fig-

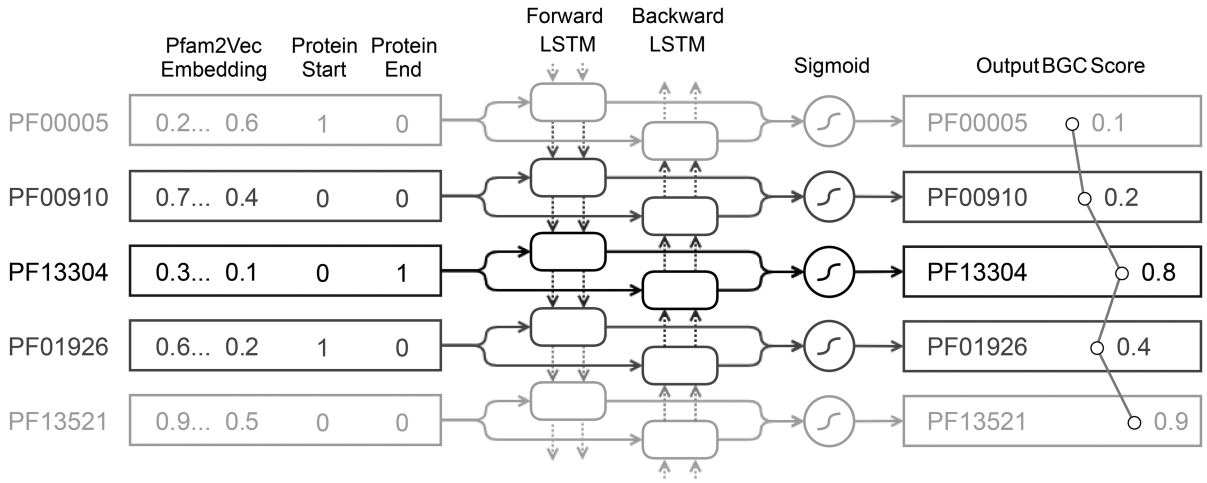

Figure 2. Bidirectional Long-Short Term Memory (BiLSTM) neural network architecture (left to right blocks). The network consists of three layers: input, BiLSTM network, and output layer. (Top to bottom) Each row represents a time step where the BiLSTM model processes a single Pfam domain from the input sequence that is maintained in genomic order. Each Pfam domain is represented as a vector of precomputed 100-dimensional pfam2vec skip-gram embedding and two binary flags indicating whether the domain is found at the beginning or at the end of a given protein. Each LSTM memory cell receives the vector from input layer (full arrows) as well as the cell's internal state that represents all previously seen Pfam domains (dashed arrows). The backward LSTM layer processes the vectors in reverse order, hence bi-directional. In each timestep, output from both LSTM memory cells (boxes) is processed through a single fully-connected node with sigmoid activation function (circle) that outputs a single BGC classification score for the given Pfam domain.

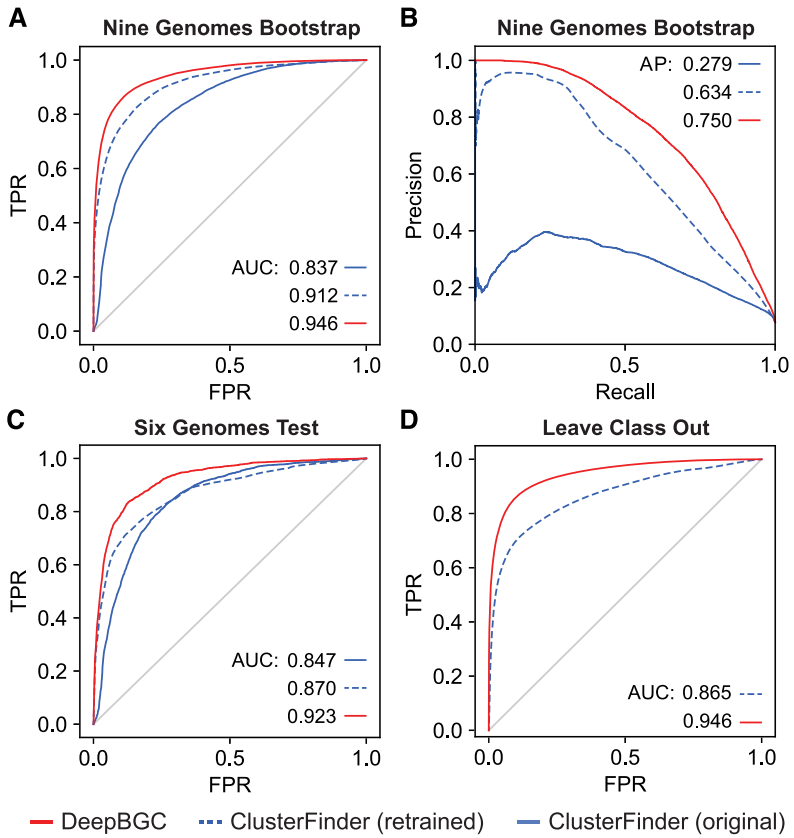

Figure 3. Model validation and testing on Pfam domain level using the (**A**) Receiver Operating Characteristic (ROC) curves and (**B**) Precision (Y-axis) Recall (X-axis) Curve reflecting performance of: (blue) original ClusterFinder HMM model, (dashed blue) ClusterFinder HMM model retrained with latest training data and latest Pfam database, and (red) DeepBGC. A total of 291 BGCs in nine bacterial genomes were used for testing, none of them were included in the training set. The DeepBGC ROC represents combination of 5 test set predictions following bootstrap. AUC (Area Under the Curve) values are as indicated: FPR – False Positive Rate (X-axis); TPR – True positive rate (Y-axis). (**C**) ROC curves reflecting performance using a total of 65 experimentally validated BGCs that were used for testing, none of them were included in the training set. (**D**) ROC curves reflecting average performance in 'Leave-Class-Out' analysis. The mean AUC for all classes is given. For individual classes performance see Supplementary Figure S5.

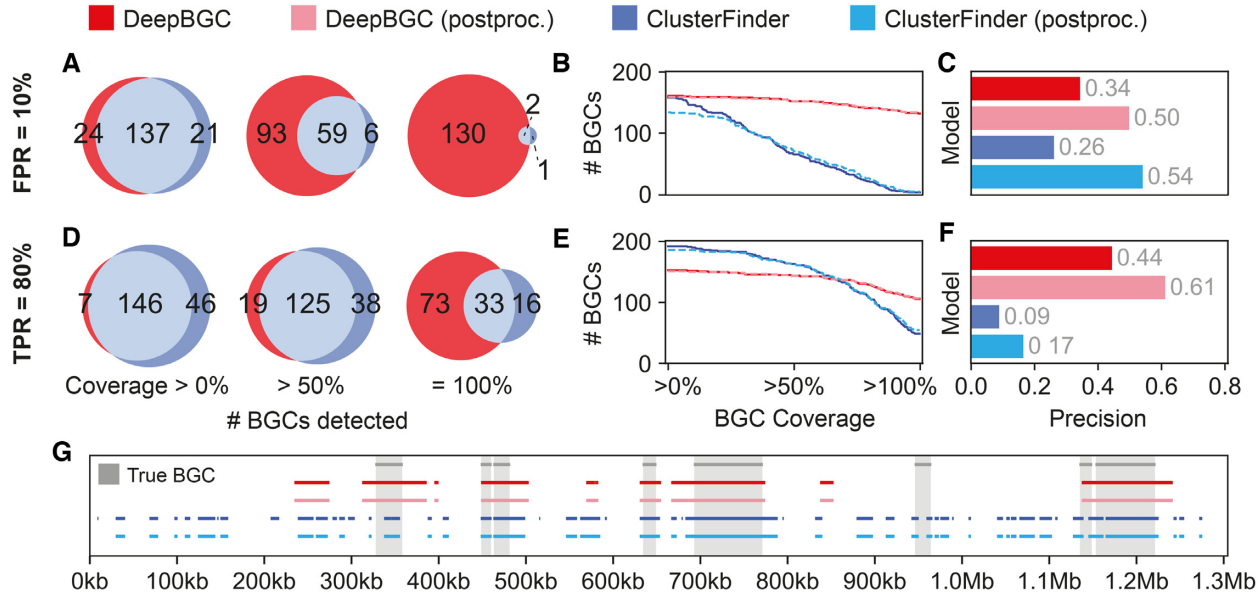

Figure 4. Precision and coverage of DeepBGC and ClusterFinder algorithms. (**A**) Number of true BGCs detected by DeepBGC (red), ClusterFinder (blue) and both models (grey), based on three BGC coverage thresholds: any (>0%), majority (>50%), and full (100%). Coverage of each annotated true BGC is defined as the fraction of its nucleotide sequence overlapping with co-located predicted BGCs. The first bootstrap test split of seven out of nine genomes was used for comparison. Domains were retrieved based on a fixed False Positive Rate (FPR) of 10%. Genes containing candidate Pfam domains were summarized to produce putative BGCs that were compared to the actual BGCs in the split data. (**B**) Cumulative coverage plot of actual BGCs by predicted BGCs for DeepBGC (red) and ClusterFinder (blue) also following post-processing (dashed). (**C**) BGC level precision for DeepBGC (red) and ClusterFinder (blue) also following post-processing (light colors) at FPR 10%. Precision was calculated as follows: the number of true positives (any overlap between actual and predicted BGCs) divided by total number of predicted BGCs. (**D–F**) Same as 'A–C' but at 80% TPR cutoff (**G**) A snapshot of contig view (X-axis genomic coordinates of *Micromonospora* sp.), manually confirmed BGCs (grey shade and bar), ClusterFinder raw and post-processed predictions (dark and light blue), DeepBGC raw and post-processed (dark and light red). For simplicity only part of the contig is shown and only at 80% TPR threshold. For all contigs, thresholds and models, see Supplementary Figures S7 and S8.

ure 4F, BGC level precision = 9%, Supplementary Figure S8, Supplementary Table S8). On the contrary, DeepBGC displayed >4-fold increase in precision compared to ClusterFinder (precision = 44%, Figure 4F).

To correct for short BGCs predicted by ClusterFinder, Cimermancic *et al.* (16) applied a post-processing step whereby neighboring clusters were merged if they were separated only by a single gene. Putative clusters that were below 2 kb in length, as well as those not containing a known biosynthetic domain, were filtered out. We implemented this approach and found that while this step dramatically improved precision for both models (Figure 4C and F), it also inevitably removed a subset of true positive predictions, most notably for ClusterFinder at 10% FPR (Figure 4B). We thus concluded that DeepBGC not only reduced the number of false predictions compared to ClusterFinder, but it also located BGCs within genomes more accurately.

Random forests provide product & activity classification

To identify the biosynthetic products derived from predicted BGCs, we classified BGC sequences by training and testing a random forest classifier using the MIBiG database, which contains classification of BGCs to one or more compound classes (Table 1). Five-fold cross-validation revealed that our random forest classifier exhibited an average AUC of 0.80 (Table 1, Supplementary Figure S9), broadly comparable with antiSMASH product type prediction accuracy when mapped to common MIBiG classification (Supplementary Table S9) with AUC of 0.78 (Table 1). Our approach can also reveal the most influential Pfam domains that drive the classifier decisions (Supplementary Figure S10). Our DeepBGC random forest classifier therefore provided a data-driven alternative to the rule-based antiSMASH classification approach. It is also important to note that the main difference between the DeepBGC and antiSMASH algorithms was saccharide classification and this was due to antiSMASH having made the design choice to refrain from confidently identifying these clusters during its processing.

In addition to identifying BGC classes, we also evaluated our ability to predict BGC molecular activity information using the 370 molecular activity labeled MIBiG BGC subset. Due to the small sample size, the classifier accounted only for the four most common compound activity classes: antibacterial, cytotoxic, inhibitor, and antifungal. Using 5-fold cross-validation, BGCs were classified according to their compound activity with modest precision (average AUC 0.61, Table 1). Larger training sets will be needed for improved performance in future work.

DeepBGC predicts novel antibacterial BGCs

Above we showed that DeepBGC, together with random forest classifiers, could effectively identify BGCs and classify their compound class and molecular activity. We therefore applied this model to unearth novel BGCs that could not be predicted by other approaches. We accomplished this

Table 1. Random forest classifiers and antiSMASH performance for classifying BGCs based on their products and their underlying activity

		RandomForest			antiSMASH		
	Samples	AUC	Precision	Recall	AUC	Precision	Recall
Polyketide	644	0.903	0.876	0.898	0.870	0.901	0.806
NRP	433	0.907	0.904	0.850	0.915	0.939	0.852
RiPP	199	0.907	0.935	0.823	0.897	0.958	0.799
Saccharide	179	0.811	0.906	0.631	0.607	0.769	0.223
Other	154	0.583	0.876	0.171	0.671	0.594	0.370
Terpene	120	0.824	0.867	0.658	0.744	0.908	0.492
Alkaloid	39	0.607	0.733	0.216	0.785	0.434	0.590
Average	252	**0.792**	**0.871**	**0.607**	0.784	0.786	0.590
Antibacterial	180	0.629	0.615	0.508			
Cytotoxic	140	0.706	0.694	0.542			
Inhibitor	81	0.545	0.473	0.115			
Antifungal	71	0.532	0.360	0.073			
Average	118	**0.603**	**0.536**	**0.310**			

The classifier was trained using 1355 MIBiG labeled BGCs belonging to one or more compound classes including polyketides (PKS), non-ribosomally synthesized peptides (NRP), ribosomally synthesized and post-translationally modified peptides (RiPP), saccharides, terpenes, alkaloids, and those belonging to other rarer classes ('other'). Areas under the curve (AUC) was determined using 5-fold cross-validation. Respective confusion matrix and important domain features are provided in Supplementary Figures S9 and S10. For molecular activity classification random forest was used as before on 370 molecular activity labeled MIBiG BGCs. Only antibacterial, cytotoxic, inhibitor or antifungal classes are accounted for.

by using a bacterial reference set of 3376 RefSeq bacterial genomes (26) and subjected these to DeepBGC, antiSMASH and ClusterFinder analyses, followed by a systematic comparison of their predictions (Supplementary Table S10). To avoid an artificially inflated number of putative novel predictions, we maximized the ability of ClusterFinder and antiSMASH to detect BGCs by accepting their default (lenient) settings, while conversely applying a strict cutoff only for DeepBGC (2% FPR at the domain level). Under these criteria, ClusterFinder predicted >4.5 times more BGCs (62491) than antiSMASH (13 865) and >5.5 times more than DeepBGC (10926). As expected, the majority of BGCs that were identified by ClusterFinder (∼75%) could not be identified by DeepBGC or antiSMASH (Figure 5a). ClusterFinder predictions showed comparable overlap with DeepBGC and antiSMASH (18% and 15% respectively). On the contrary, DeepBGC comparison revealed that the majority of DeepBGC predictions overlapped with ClusterFinder (∼90%), and ∼39% with antiSMASH of which 35% overlapped with both (Figure 5A). Approximately 5% (566) of DeepBGC predictions could not be uncovered by any other method, using our conservative thresholds against false positive detection. Evaluation of the rule-based antiSMASH predictions revealed that the BGCs missed by DeepBGC were overall of diverse classes, and many of these are expected to be recovered by using more lenient DeepBGC parameters (Supplementary Table S11, Supplementary Figure S11). A specific example of *S. coelicolor* BGC detection (a well studied organism in the field) confirmed that looser DeepBGC thresholds allowed for detection of more BGCs that were otherwise identified by rule-based antiSMASH, with the number of antiSMASH BGCs identified by DeepBGC ranging from 21 to 9 out of 28, across thresholds of 0.1–0.9, respectively (Supplementary Figures S12 and S13).

We further explored these novel signatures by interrogating the ∼5% novel BGCs that DeepBGC detected. We considered 227 BGCs that consisted of at least 5 Pfam domains and classified them based on their compound class and molecular activity. We found that the result was enriched for BGCs with no confident class (∼70%) and for those that originated from the *Mycobacterium* genus (∼49%, Supplementary Figure S14). When visualized using t-SNE the novel BGCs that could not be confidently assigned to a single compound class straddled the borders between distinct classes (grey plus signs, Figure 5b), while the remaining were tightly clustered with known BGCs according to their respective class (Figure 5b).

To further highlight the performance of DeepBGC functional classification on real data, we evaluated the *S. coelicolor* reference genome used above. We found that some of the *S. coelicolor* BGCs classified as RIPPs did not appear to contain genetic signatures of RIPPs upon manual inspection (Table S12). DeepBGC also identified a lycopene-like carotinoid cluster but mis-identified it as a terpenoid cluster, while antiSMASH considered this a primary metabolite. These examples further support our benchmarking findings above, which tell us that while DeepBGC performs well compared to other methods, it is not able to perfectly identify and classify BGCs.

To begin evaluating the individual BGCs, we ranked our novel predictions based on their total number of Pfam domain counts, their predicted antibacterial activity score and their similarity to known BGC in the MIBiG database (Supplementary Table S13). We used this list to manually identify a candidate BGC that could not be assigned to a specific compound class, displayed a low similarity score to known BGCs and had a high antibacterial prediction score. This novel BGC resided in the genome of the pathogenic bacterium *Mycobacterium tuberculosis*, in which BGCs are known to be abundant (31–35). The cluster was distant from any other neighboring predictions (∼9 kb, Figure 5C) and rich in a diverse set of regulatory, transport and modifying enzymes (Figure 5d, Supplementary Table S14) including acetyltransferase and glyoxalase/bleomycin resistance protein, which have been known to catalyze diverse biochemical reactions (36). The cluster also encoded for a type II toxin-antitoxin system, further supporting its poten-

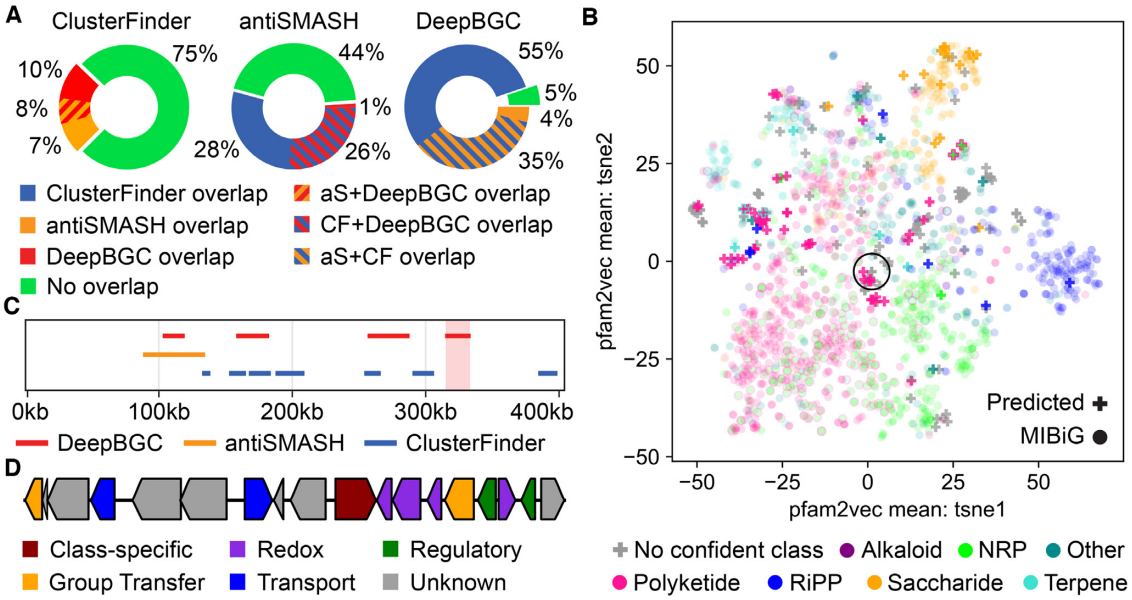

Figure 5. DeepBGC uncovers novel BGCs with antibacterial activity in bacterial genomes. (**A**) Comparison of BGC predictions between (left) ClusterFinder, (middle) antiSMASH and (right) DeepBGC. Default ClusterFinder settings from antiSMASH suite were used. For antiSMASH, rule-based predictions under default settings were considered. In DeepBGC, a 2% FPR at the domain level was applied with no further post-processing. CF – ClusterFinder; aS -antiSMASH. (**B**) t-Distributed Stochastic Neighbor Embedding (t-SNE) of all 1355 class labelled BGCs from the MIBiG database (circles) overlaid with the putative novel 227 BGCs that could be predicted solely by DeepBGC (plus signs). BGCs were represented by the mean value of their pfam2vec domain vectors and are colored by the respective known or predicted class as indicated. (**C**) A snapshot of contig view (X-axis genomic coordinates of *Mycobacterium tuberculosis*) of BGC predictions by (red) DeepBGC, (blue) ClusterFinder, and (orange) antiSMASH combined with ClusterFinder. A novel BGC candidate predicted only by DeepBGC is highlighted (light red shade). (**D**) The novel BGC structure is given, respective genes are colored based on the underlying domain type. For domain IDs see Supplementary Table S14.

tial cytotoxic activity (37). Such wealth of modifying enzymes could potentially grant the final natural product a novel chemistry. A search for BGCs with similar domain architecture revealed a similar cluster (80% similarity) in a different *Mycobacterium tuberculosis* strain. More divergent clusters with >60% similarity scores were discovered in other *Mycobacterium* species, and were also supported by predictions from antiSMASH and ClusterFinder (Supplementary Figure S15). We also highlighted five additional putative BGCs that were uniquely identified by DeepBGC and were from diverse bacterial taxa (Supplementary Figure S16, Supplementary Table S15). Together with our results from the leave-class-out validation and other validation metrics above, these discoveries highlight the value of DeepBGC and its ability to mine bacterial genomes to provide previously unrealized insights into bacterial natural product chemistry. These insights will provide direction for experimental validation and follow up experiments, which will also serve as a robust validation of the predicted BGCs' biological relevance.

DISCUSSION

Here we present DeepBGC, a comprehensive deep learning strategy for identifying BGCs from bacterial genomes and classifying them by their product class and chemical activity. Our deep learning approach is based on concepts from the NLP field and builds on existing algorithms that either suffer from a restricted ability to identify novel BGC classes or from a limited ability to accurately identify BGCs within a genome. We demonstrated that our DeepBGC approach outperformed one commonly used machine learning algorithm, ClusterFinder, in its ability to identify BGCs accurately within a genome (Supplementary Table S16). Our leave-class-out analysis suggested that the DeepBGC model also possessed a greater potential to extrapolate and identify BGC classes that it has not encountered before. The supplemental random forest classification approach allowed us to accurately identify BGC classes by their Pfam domain composition, and enabled some prediction of the chemical activity of the resulting secondary metabolites despite the limited sample size available for training. Finally, like other machine learning algorithms, DeepBGC is poised to continue improving over time with the continuous discovery, validation, and labeling of BGCs in microbial genomes.

Machine learning has had a dramatic impact on NLP methodologies, giving rise to powerful word embedding techniques such as word2vec, which allow representation of words as low-dimensionality vectors of real numbers through which they enhance learning by context (38). Asgari *et al.* (39) recently adopted word embedding and neural networks to improve classification of protein families using amino acid sequence vectors and Kim *et al.* (40) has also introduced Mut2Vec for representation of cancerous mutations. In our current study, we fortified our DeepBGC BiLSTM network's ability to learn complex patterns in genomic sequences with novel pfam2vec representation vectors that were generated from a large unlabeled corpus of genomic sequences. By doing so we enabled an improved machine

understanding of the enigmatic genomic context. We believe that our pfam2vec approach could further assist in annotating domains of unknown functions based on their genomic context. To this end, we also provided a set of nearest domain pairs of known and unknown functions (Supplementary Table S17), yet this pursuit was ultimately beyond the scope of this study and we will continue pursuing this in future work.

We applied our model to a real-world dataset (collection of reference bacterial genomes) to highlight its ability to provide unique insights into bacterial BGCs. The inflated number of BGCs predicted by ClusterFinder could be readily explained by its low precision. Our results suggested that many ClusterFinder predictions were false positives, and those that were not false positives only represented small fractions of true BGCs. Throughout the study, parameter settings were conservatively chosen because we preferred underprediction over annotation of bacterial genomes with incorrectly predicted BGCs.

We found that our model identified BGCs of diverse product classes that ClusterFinder could not identify, although the majority of BGC classes could not be confidently assigned. This suggests a potential for identifying novel BGCs, warranting significant future validation to explore those BGC candidates. These unknown BGCs were largely found within Mycobacteria, a genus known to harbor many diverse BGCs (31–35). Therefore, our model not only identifies potential new BGC signatures, but it does so in bacteria with a known prominence for BGCs, and thus the bacteria we might expect *a priori* to have a significant reservoir of novel BGCs.

While we illustrated the application of this tool to reference genomes, we also anticipate DeepBGC applications to the microbiome through shotgun metagenomic datasets. An understanding of differential BGC presence or expression (using metatranscriptomics approaches) could provide new insights into microbiome functionality, underlying mechanisms of disease, and therapeutic approaches. Although beyond the scope of this work, the incorporation of DeepBGC into microbiome sequence analyses is an exciting avenue for future studies.

While our deep learning based, DeepBGC approach outperformed other commonly used models, it is important to note its limitations. Like other existing models, this model was trained on existing BGC databases that are heavily biased towards BGCs from natural product 'workhorses' such as *Streptomyces*. This bias in the training data is likely to limit the ability of the model to identify novel BGCs in bacterial sources that are poorly characterized in the databases, including bacteria found in complex microbial communities (the microbiome). We addressed this extrapolation concern by performing leave-class-out validation to highlight the generalizability of our approach over other existing approaches. Despite our improved performance, further work is needed to curate more diverse BGC databases which can be used to improve the training and validation, and overall performance as a result, of our model.

Another important caveat is the comparison of DeepBGC to antiSMASH, which represents a rule-based approach to BGC identification. Methods that implement domain expert knowledge through human defined rules leverage extensive field information and are therefore clearly valuable in finding BGC signatures that align with our current understanding of the field. Machine learning approaches such as ClusterFinder and DeepBGC excel at identifying novel signatures and continue to scale as our databases improve. However, these models are only as good as the accuracy and depth of the underlying databases used for training. This was reflected by our observation that some well studied classes were recovered regardless of threshold, while others were more threshold-dependent (Supplementary Figure S11). This was further highlighted by our observation that some classes were mis-identified in the *S. coelicolor* reference genome example, which agrees with our other benchmarking results. Overall this highlights that these machine-learning models should be viewed as complimentary instead of redundant, as they provide unique advantages in signal identification.

The resulting classification and misclassifications of our DeepBGC model, as well as other machine learning models, highlight that machine learning approaches identify commonalities in the predictive features of their datasets, and those may not reflect true biology. As we alluded to above, because the machine learning models are trained on relatively small reference datasets, they may learn biases instead of true biological signal such as learning to associated ABC transporter signals with RiPP clusters (an association which does not reflect their true biology). Together this illustrates the importance of incorporating larger datasets, and we built DeepBGC to allow that easy incorporation of data. This will be an important area for future study.

Despite this limitation, our model performs well compared to other methods, and represents a useful algorithm for the field of natural product discovery. Due to our model's improved ability to identify novel BGCs, we showed that we could identify BGC that were missed by other existing models, and thereby identify previously unknown sources for natural products in existing bacterial genome sequences. By reducing the number of fragmented BGCs being identified in bacterial genomes, our improved prediction accuracy will reduce BGC count inflation. Additionally, by providing more accurate BGC border predictions, we will reduce the human triage effort of cleaning up predicted BGCs whose genomic positions were not entirely accurate. Together DeepBGC represents an advancement over the current "state-of-the-art" by improving BGC identification accuracy, BGC genomic location prediction, and identification of potentially novel BGC signatures that were not present in the current training knowledgebase. This will be used to empower follow up genome mining for novel BGCs and their resulting natural products, and the improved extrapolation capabilities will empower BGC mining of microbiome datasets, which still represent an under-explored genomic BGC resource. These microbiome analyses, including associations with disease phenotypes and identification of novel chemical matter in classes such as antibiotics or immunomodulatory agents, could have important clinical impacts for translating microbiome data to therapeutic interventions.

DATA AVAILABILITY

All data used in this work was obtained from the public domain and is specified in the respective methods sections. All code for this publication is available at the following GitHub repository: https://github.com/Merck/bgc-pipeline. DeepBGC is available for installation and use as a Python package at the following GitHub repository: https://github.com/Merck/deepbgc.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank Otakar Smrz, Joseph Lehar and Ivo Lasek for stimulating discussions. We are immensely grateful to David Dzamba, Matthew Tudor, Jyoti Shah and Petr Mejzlik for their comments on earlier version of the manuscript. We are also especially grateful to Mohamed Donia for the stimulating discussions and comments. We also would like to acknowledge and thank Nicole L. Glazer, Jens Christensen and Carol A. Rohl for supporting this work.
Author contributions: G.H., C.W. and D.B. conceived, designed and supervised the study. J.S. and L.R. guided the implementation of BiLSTM network by A.P. J.D., J.S. and D.P. explored protein and BGC similarities approaches. J.S. and D.B. designed the pfam2vec algorithm, J.S. implemented it and DP evaluated its performance. J.S. designed the bootstrap approach and DP implemented it. J.K., M.W. and D.C. advised and assessed the HMM, LSTM and classification methods. D.P. developed and implemented the entire pipeline and performed all data analyses in this study. D.P., O.K. and J.S. designed and implemented the random forest classifier. R.W. evaluated DeepBGC predictions and G.P., G.T. and D.H. advised throughout the study. D.P. and D.B. designed the figures. D.B. drafted the manuscript. G.H., C.W., R.W. and D.P. contributed to the final draft, all authors read and approved the final version of the manuscript.

FUNDING

This work supported by Merck Sharp & Dohme Corp., a subsidiary of Merck & Co., Inc., Kenilworth, NJ, USA. Funding for open access charge: Merck & Co. Inc.
Conflict of interest statement. A subset of manuscript authors are inventors on a patent related to this work (patent application number: 62/779.697). All of this work/code is licensed under the MIT permissive free software license.

REFERENCES

1. Newman,D.J. and Cragg,G.M. (2012) Natural products as sources of new drugs over the 30 years from 1981 to 2010. *J. Nat. Prod.*, **75**, 311–335.
2. Milshteyn,A., Schneider,J.S. and Brady,S.F. (2014) Mining the metabiome: identifying novel natural products from microbial communities. *Chem. Biol.*, **21**, 1211–1223.
3. Ventola,C.L. (2015) The antibiotic resistance crisis: part 1: causes and threats. *P T*, **40**, 277–283.
4. Pendleton,J.N., Gorman,S.P. and Gilmore,B.F. (2013) Clinical relevance of the ESKAPE pathogens. *Expert Rev. Anti. Infect. Ther.*, **11**, 297–308.
5. Zhang,H. and Chen,J. (2018) Current status and future directions of cancer immunotherapy. *J. Cancer*, **9**, 1773–1781.
6. Shen,B. (2015) A new golden age of natural products drug discovery. *Cell*, **163**, 1297–1300.
7. DeCorte,B.L. (2016) Underexplored opportunities for natural products in drug discovery. *J. Med. Chem.*, **59**, 9295–9304.
8. Harvey,A.L., Edrada-Ebel,R. and Quinn,R.J. (2015) The re-emergence of natural products for drug discovery in the genomics era. *Nat. Rev. Drug Discov.*, **14**, 111–129.
9. Hopwood,D.A. and Merrick,M.J. (1977) Genetics of antibiotic production. *Bacteriol. Rev.*, **41**, 595–635.
10. Martin,J.F. (1992) Clusters of genes for the biosynthesis of antibiotics: regulatory genes and overproduction of pharmaceuticals. *J. Ind. Microbiol.*, **9**, 73–90.
11. Martín,M.F. and Liras,P. (1989) Organization and expression of genes involved in the biosynthesis of antibiotics and other secondary metabolites. *Annu. Rev. Microbiol.*, **43**, 173–206.
12. Medema,M.H. and Fischbach,M.A. (2015) Computational approaches to natural product discovery. *Nat. Chem. Biol.*, **11**, 639–648.
13. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
14. Medema,M.H., Blin,K., Cimermancic,P., de Jager,V., Zakrzewski,P., Fischbach,M.A., Weber,T., Takano,E. and Breitling,R. (2011) antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences. *Nucleic Acids Res.*, **39**, W339–W346.
15. Weber,T., Rausch,C., Lopez,P., Hoof,I., Gaykova,V., Huson,D.H. and Wohlleben,W. (2009) CLUSEAN: a computer-based framework for the automated analysis of bacterial secondary metabolite biosynthetic gene clusters. *J. Biotechnol.*, **140**, 13–17.
16. Cimermancic,P., Medema,M.H., Claesen,J., Kurita,K., Wieland Brown,L.C., 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.
17. Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.
18. Skinnider,M.A., Merwin,N.J., Johnston,C.W. and Magarvey,N.A. (2017) PRISM 3: expanded prediction of natural product chemical structures from microbial genomes. *Nucleic Acids Res.*, **45**, W49–W54.
19. Yoon,B.-J. (2009) Hidden Markov models and their applications in biological sequence analysis. *Curr. Genomics*, **10**, 402–415.
20. Choo,K.H., Tong,J.C. and Zhang,L. (2004) Recent applications of Hidden Markov Models in computational biology. *Genomics. Proteomics Bioinformatics*, **2**, 84–96.
21. Eddy,S.R. (2004) What is a hidden Markov model? *Nat. Biotechnol.*, **22**, 1315–1316.
22. Finn,R.D., Coggill,P., Eberhardt,R.Y., Eddy,S.R., Mistry,J., Mitchell,A.L., Potter,S.C., Punta,M., Qureshi,M., Sangrador-Vegas,A. *et al.* (2016) The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Res.*, **44**, D279–D285.
23. Hochreiter,S., Heusel,M. and Obermayer,K. (2007) Fast model-based protein homology detection without alignment. *Bioinformatics*, **23**, 1728–1736.
24. Hochreiter,S. and Schmidhuber,J. (1997) Long Short-Term memory. *Neural Comput.*, **9**, 1735–1780.
25. Schuster,M. and Paliwal,K.K. (1997) Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.*, **45**, 2673–2681.
26. O'Leary,N.A., Wright,M.W., Brister,J.R., Ciufo,S., Haddad,D., McVeigh,R., Rajput,B., Robbertse,B., Smith-White,B., Ako-Adjei,D. *et al.* (2016) Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.*, **44**, D733–D745.
27. Hyatt,D., Chen,G.-L., LoCascio,P.F., Land,M.L., Larimer,F.W. and Hauser,L.J. (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, **11**, 119.
28. Mikolov,T., Chen,K., Corrado,G. and Dean,J. (2013) Efficient Estimation of Word Representations in Vector Space.
29. 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. *Nat. Chem. Biol.*, **11**, 625–631.
30. Ziemert,N., Alanjary,M. and Weber,T. (2016) The evolution of genome mining in microbes - a review. *Nat. Prod. Rep.*, **33**, 988–1005.
31. Chavadi,S.S., Stirrett,K.L., Edupuganti,U.R., Vergnolle,O., Sadhanandan,G., Marchiano,E., Martin,C., Qiu,W.-G., Soll,C.E. and Quadri,L.E.N. (2011) Mutational and phylogenetic analyses of the mycobacterial mbt gene cluster. *J. Bacteriol.*, **193**, 5905–5913.
32. Quadri,L.E., Sello,J., Keating,T.A., Weinreb,P.H. and Walsh,C.T. (1998) Identification of a Mycobacterium tuberculosis gene cluster encoding the biosynthetic enzymes for assembly of the virulence-conferring siderophore mycobactin. *Chem. Biol.*, **5**, 631–645.
33. Li,W., He,J., Xie,L., Chen,T. and Xie,J. (2013) Comparative genomic insights into the biosynthesis and regulation of mycobacterial siderophores. *Cell Physiol. Biochem.*, **31**, 1–13.
34. Harris,N.C., Sato,M., Herman,N.A., Twigg,F., Cai,W., Liu,J., Zhu,X., Downey,J., Khalaf,R., Martin,J. *et al.* (2017) Biosynthesis of isonitrile lipopeptides by conserved nonribosomal peptide synthetase gene clusters in Actinobacteria. *Proc. Natl. Acad. Sci. U.S.A.*, **114**, 7025–7030.
35. Tobias,N.J., Doig,K.D., Medema,M.H., Chen,H., Haring,V., Moore,R., Seemann,T. and Stinear,T.P. (2013) Complete genome sequence of the frog pathogen Mycobacterium ulcerans ecovar Liflandii. *J. Bacteriol.*, **195**, 556–564.
36. Armstrong,R.N. (2000) Mechanistic diversity in a metalloenzyme superfamily. *Biochemistry*, **39**, 13625–13632.
37. Anantharaman,V. and Aravind,L. (2003) New connections in the prokaryotic toxin-antitoxin network: relationship with the eukaryotic nonsense-mediated RNA decay system. *Genome Biol.*, **4**, R81.
38. LeCun,Y., Bengio,Y. and Hinton,G. (2015) Deep learning. *Nature*, **521**, 436–444.
39. Asgari,E. and Mofrad,M.R.K. (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS One*, **10**, e0141287.
40. Kim,S., Lee,H., Kim,K. and Kang,J. (2018) Mut2Vec: distributed representation of cancerous mutations. *BMC Med. Genomics*, **11**, 33.