Exploring the chemical space of post-translationally modified peptides in Streptomyces with machine learning

Kloosterman, A.M.

Author: Kloosterman, A.M.
Title: Exploring the chemical space of post-translationally modified peptides in Streptomyces with machine learning
Issue Date: 2021-05-12

3

Expansion of RiPP biosynthetic space through integration of pan-genomics and machine learning uncovers novel candidate RiPP families

Alexander M. Kloosterman
Peter Cimermancic
Michalis Hadjithomas
Mohamed S. Donia
Michael A. Fischbach
Gilles P. van Wezel
Marnix H. Medema

The work described in this chapter is part of the publication:

Kloosterman, et al., *Expansion of RiPP biosynthetic space through integration of pan-genomics and machine learning uncovers a novel class of lantibiotics.* PLOS Biol, 2020. **18**(12): e3002016.

Abstract

Microbial natural products constitute a wide variety of chemical compounds, many which can have antibiotic, antiviral or anticancer properties that make them interesting for clinical purposes. Natural product classes include polyketides (PKS), non-ribosomal peptides (NRPS) and ribosomally synthesized and post-translationally modified peptides (RiPPs). While variants of biosynthetic gene clusters (BGCs) for known classes of natural products are easy to identify in genome sequences, BGCs for new compound classes escape attention. In particular, evidence is accumulating that for RiPPs, subclasses known thus far may only represent the tip of an iceberg. Here, we present decRiPPter (Data-driven Exploratory Class-independent RiPP TrackER), a RiPP genome mining algorithm aimed at the discovery of novel RiPP subclasses. DecRiPPter combines a classifier based on Support Vector Machines (SVMs) that identifies candidate RiPP precursors, with pan-genomic analyses to identify which of these are encoded within operon-like structures that are part of the accessory genome of a genus. Subsequently, it prioritizes such regions based on the presence of new enzymology and based on patterns of gene cluster and precursor peptide conservation across species. We then applied decRiPPter to mine 1,295 *Streptomyces* genomes, which led to the identification of 42 new candidate RiPP families that could not be found by existing programs. The BGCs of these families encode enzyme families not previously associated with RiPP biosynthesis, or precursors with interesting repeating patterns. These results highlight how novel natural product families can be discovered by methods going beyond sequence similarity searches to integrate multiple pathway discovery criteria.

Code and data availability

The source code of decRiPPter is freely available online at https://github.com/Alexamk/decRiPPter. Results of the data analysis are available online at https://decrippter.bioinformatics.nl. All training data and code used to generate these, as well as outputs of the data analyses, are available on Zenodo at doi:10.5281/zenodo.3834818.

Introduction

The introduction of antibiotics in the 20th century contributed hugely to extend the human life span. However, the increase in antibiotic resistance and the concomitant steep decline in the number of new compounds discovered via high-throughput screening [22, 25], means that we again face huge challenges to treat infections by multi-drug resistant bacteria [157]. The low return of investment of high throughput screening is due to dereplication, in other words, the rediscovery of bioactive compounds that have been identified before [23, 24]. A revolution in our understanding was brought about by the development of next-generation sequencing technologies. Actinobacteria are the most prolific producers of bioactive compounds, including some two-thirds of the clinical antibiotics [32, 158]. Mining of the genome sequences of these bacteria revealed a huge repository of previously unseen biosynthetic gene clusters (BGCs), highlighting that their potential as producers of bioactive molecules had been grossly underestimated [27, 32, 159]. However, these BGCs are often not expressed under laboratory conditions, most likely because the environmental cues that activate their expression in their original habitat are missing [26, 30]. To circumvent these issues, a common strategy is to select a candidate BGC and force its expression by expression of the pathway-specific activator or via expression of the BGC in a heterologous host [33]. However, these methods are time-consuming, while it is hard to predict the novelty and utility of the compounds they produce.

To improve the success of genome mining-based drug discovery, many bioinformatic tools have been developed for identification and prioritization of BGCs. These tools often rely on conserved genetic markers present in BGCs of certain natural products, such as polyketides (PKs), non-ribosomal peptides (NRPs) and terpenes [39, 40, 62]. While these methods have unearthed vast amounts of uncharacterized BGCs, they further expand on previously characterized classes of natural products. This raises the question of whether entirely novel classes of natural products could still be discovered. A few genome mining methods, such as ClusterFinder [41] and EvoMining [160, 161], have tried to tackle this problem. These methods either use criteria true of all BGCs or build around the evolutionary properties of gene families found in BGCs, rather than using BGC-class-specific genetic markers. While the lack of clear genetic markers may result in a higher number of false positives, these methods have indeed charted previously uncovered biochemical space and led to the discovery of new natural products.

One class of natural products whose expansion has been fueled by the increased amount of genomic sequences available is that of the ribosomally synthesized and post-translationally modified peptides (RiPPs) [42]. RiPPs are characterized by a unifying biosynthetic theme: a small gene encodes a short precursor peptide, which is extensively modified by a series of enzymes that typically recognize the N-terminal part of the precursor called the leader peptide, and finally cleaved to yield the mature product [43]. Despite this common biosynthetic logic, RiPP modifications are highly diverse. The latest comprehensive review categorizes RiPPs into roughly 20 different subclasses [42], such as lanthipeptides, lasso peptides and thiopeptides. Each of these subclasses is characterized by one or more specific modifications, such as the thioether bridge in lanthipeptides or the knot-like structure of lasso peptides. Despite the extensive list of known subclasses and modifications, new RiPP subclasses are still being found. These often carry unusual modifications, such as D-amino acids [98], addition of unnatural amino acids [162, 163], β-amino acids [103], or new variants of thioether crosslinks [55, 106]. These discoveries strongly indicate that the RiPP genomic landscape remains far from completely charted, and that novel types of RiPPs with new and unique biological activities may yet be uncovered. However, RiPPs pose a unique and major challenge to genome-based pathway identification attempts: unlike in the case of NRPSs and PKSs, there are no universally conserved enzyme families or enzymatic domains that are found across all RiPP pathways. Rather, each subclass of RiPPs comprises its own unique set of enzyme families to post-translationally modify the precursor peptides belonging to that subclass. Hence, while biosynthetic gene clusters (BGCs) for known RiPP subclasses can be identified using conventional genome mining algorithms, a much more elaborate strategy is required to automate the identification of novel RiPP subclasses.

Several methods have made progress in tackling this challenge. 'Bait-based' approaches such as RODEO [45, 55, 72-74, 86] and RiPPer [52] identify

RiPP BGCs by looking for homologues of RiPP modifying enzymes of interest, and facilitate identifying the genes encoding these enzymes in novel contexts to find many new RiPP BGCs. A study was also described using a transporter gene as a query that is less dependent on a specific RiPP subclass [164]. However, these methods still require a known query gene from a known RiPP subclass. Another tool recently described, NeuRiPP, is capable of predicting precursors independent of RiPP subclass, but is limited to precursor analysis [88]. Yet another tool, DeepRiPP, can detect novel RiPP BGCs that are chemically far removed from known examples, but is mainly designed to identify new members of known subclasses [89]. In the end, an algorithm for the discovery of BGCs encoding novel RiPP subclasses will need to integrate various sources of information to reliably identify genomic regions that are likely to encode RiPP precursors along with previously undiscovered modifying enzymes.

Here, we present decRiPPter (Data-driven Exploratory Class-independent RiPP TrackER), an integrative algorithm for the discovery of novel subclasses of RiPPs, without requiring prior knowledge of their specific modifications or core enzymatic machinery. DecRiPPter employs a classifier based on Support Vector Machines (SVMs) that predicts RiPP precursors regardless of RiPP subclass, and combines this with pan-genomic analysis to identify which putative precursor genes are located within specialized genomic regions that encode multiple enzymes and are part of the accessory genome of a genus. Sequence similarity networking of the resulting precursors and gene clusters then facilitates further prioritization. Applying this method to the gifted natural product producer genus *Streptomyces*, we identified 42 new RiPP family candidates. Experimental characterization of a widely distributed candidate RiPP BGC led to the discovery of a novel lanthipeptide that was produced by a previously unknown enzymatic machinery.

Results and Discussion

RiPP BGC discovery by detection of genomic islands with characteristics typical of RiPP BGCs

Given the promise of RiPPs as a source for novel natural products, we set out to construct a platform to facilitate identification of novel RiPP subclasses. Since no criteria could be used that are specific for individual RiPPsub classes, we used three criteria that generally apply to RiPP BGCs: 1) they contain one or more open reading frames (ORFs) for a precursor peptide; 2) they contain genes encoding modifying machinery in an operon-like gene cluster together with precursor gene(s); 3) they have a sparse distribution within the wider taxonomic group in which they are found. To focus on novel RiPP subclasses, we added a fourth criterion: 4) they have no direct similarity to BGCs of known classes (Figure 1).

For the first criterion, we trained several SVM classifiers to distinguish between RiPP precursors and other peptides. A collection of 175 known RiPP precursors, gathered from RiPP clusters from the MIBiG repository [29, 140] was used as a positive training set (Table S1). For the negative training set, we generated a set of 20,000 short non-precursor sequences, consisting of 10,000 randomly selected short proteins (<175 amino acids long) from Uniprot without measurable similarity to RiPP precursors (representative of gene encoding proteins but not RiPP precursors), and 10,000 translated intergenic sequences between a stop codon and the next start codon of sizes 30-300 nt taken from 10 genomes across the bacterial tree of life (representative of spurious ORFs that do not encode proteins). From both positive and negative training set sequences, 36 different features were extracted describing the amino acid composition and physicochemical properties of the protein/peptide sequences, as well as localized enrichment of amino acids prone to modification by modifying enzymes. Based on these, several SVMs were trained with different parameters and kernel functions, of which the average was taken as a final score (Materials and Methods). To make sure that this classifier could predict precursors independent of RiPP subclass, we trained it on all possible subsets of the positive training set in which one of the RiPP subclasses was entirely left out.

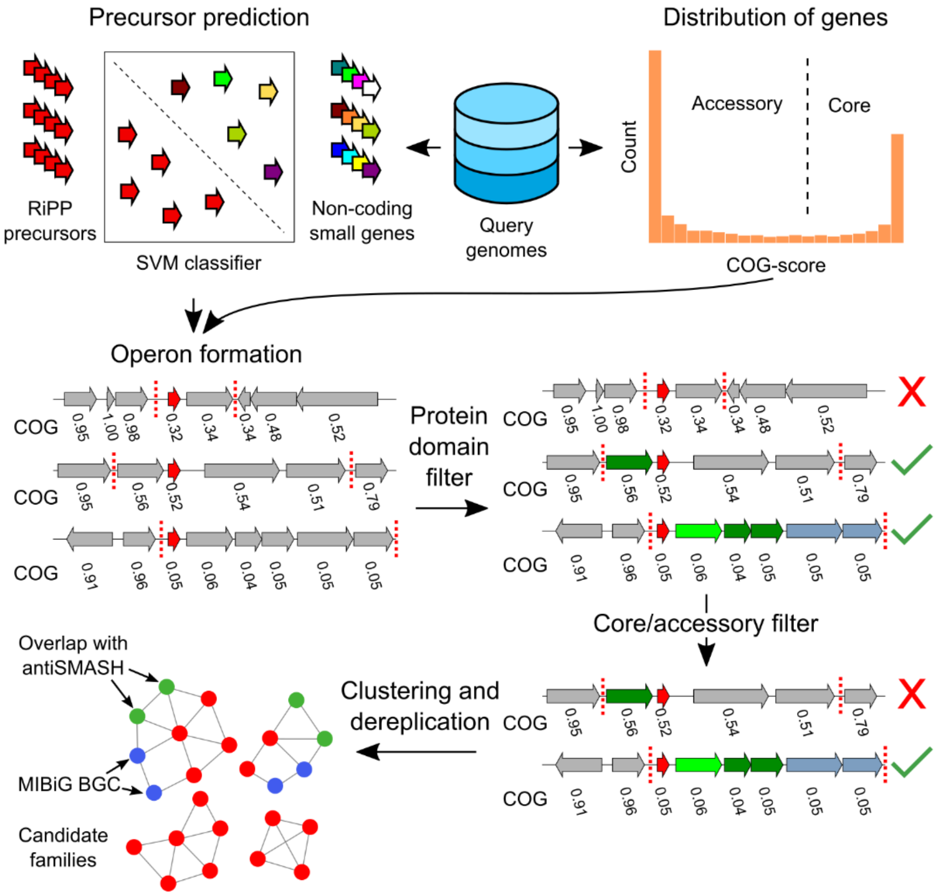


Figure 1. **decRiPPter pipeline for the detection of novel RiPP families.** The SVM classifier is used to identify all candidate RiPP precursors in a given group of genomes, using all predicted proteins smaller than 100 amino acids. The gene clusters formed around the precursors are analyzed for specific protein domains. In addition, all COG scores are calculated to act as an additional filter, and to aid in gene cluster detection. The remaining gene clusters are clustered together and with MIBiG gene clusters to dereplicate and organize the results. In addition, overlap with antiSMASH detected BGCs is analyzed.

We termed this strategy leave-one-class-out cross-validation. Typically, the classifier was still capable of predicting the subclass that was left out. To validate the classifier, we used it to score precursor hits from the various RiPP mining studies performed using RODEO [45, 55, 72-74, 86]. In general, 66.7% of all precursors identified by RODEO's SVMs were scored as positive by decRiPPter's classifier (Table S2). This shows that, for known RiPP subclasses, the classifier described here is well capable of detecting the majority of precursor peptides, although it is, unsurprisingly, outperformed by the dedicated, subclass-specific SVMs of RODEO.

For the second criterion, we made use of the fact that the majority of RiPP BGCs appear to contain the genes encoding the precursor and the core biosynthetic enzymes in the same strand orientation within close intergenic distance (81.6% of MIBiG RiPPs). Therefore, candidate gene clusters are formed from the genes that appear to reside in an operon with predicted precursor genes, based on intergenic distance and the COG scores calculated (Cluster of Orthologous Genes, see description below, Materials and Methods, Figure 2 and Figure S1). These gene clusters were then analyzed for protein domains that could constitute the modifying machinery (Figure 1B). Rather than restricting ourselves to specific protein domains, we constructed a broad dataset of Pfam and TIGRFAM domains that are linked to an E.C. number using InterPro mappings [165]. This dataset was extended with a previously curated set of Pfam domains found to be prevalent in the positive training set of the ClusterFinder algorithm [41], and manually curated, resulting in a set of 4,131 protein domains. We also constructed Pfam [75] and TIGRFAM [76] domain datasets of transporters, regulators and peptidases, as well as a dataset consisting of known RiPP modifying domains to provide more detailed annotation and allow specific filtering of RiPP BGCs based on the presence of each of these types of Pfam domains (Data S1, available from https://github.com/Alexamk/decRiPPter/tree/master/data/domains/).

For the third criterion, we sought to distinguish specialized genomic regions from conserved genomic regions. Indeed, most BGCs are sparingly distributed among genomes, with even closely related strains showing differences in their BGC repertoires [3-5]. We therefore developed an algorithm that separates the ‘core’ genome from the ‘accessory’ genome, by comparing all genes in a group of query genomes from the same taxon (typically a genus), and identifying the frequency of occurrence of each gene within that group of genomes (Figure 1C and Figure 2).

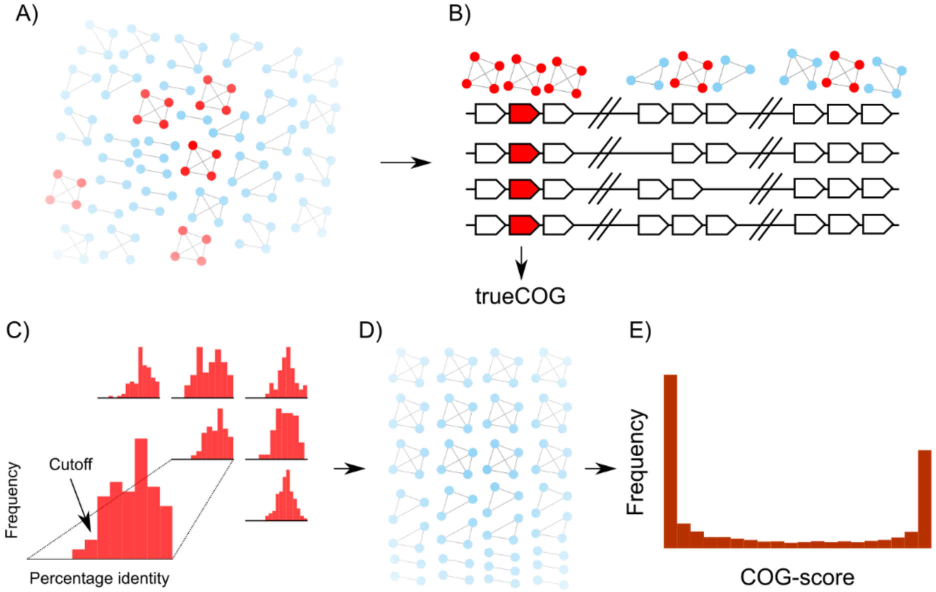


Figure 2. **decRiPPter determines the frequencies of occurrence of genes to calculate the COG score.** In this example, the COG scores of four genomes are calculated. A) All encoded proteins are aligned to find bidirectional best hits (BBHs; edges). All clusters of BBHs conserved across all genomes are displayed as red. If one genome does not contain a homologous gene, or the gene in question is not a BBH with all genes from the cluster from other genomes, it is not considered a conserved group of BBHs. B) If the flanking genes of the clusters of BBHs are also part of clusters of BBHs, the center genes are considered to form a true Cluster of Orthologous Genes (trueCOG). Of the three cases displayed here, only the leftmost group passes this criterion; for the center group, not all genes are conserved, and for the right group, not all genes are BBHs with one another in the flanking groups. C) The distribution of sequence similarities is used to calculate a sequence identity cutoff to use for each pair of genomes. D) All genes are paired using the sequence identity cutoffs determined in the previous step. E) The COG-score is calculated for each gene. Typically, a bimodal distribution can be seen, with many genes either conserved across all genomes, or only present in a single organism.

For the purpose of comparing genes between genomes, we reasoned that it was more straightforward to identify groups of functionally closely related genes that also include recent paralogues, due to the complexities of dealing with orthology relationships across large numbers of genomes (especially for biosynthetic genes that are known to have a discontinuous taxonomic distribution and may undergo frequent duplications [166]). Therefore, decRiPPter first identifies the distribution of sequence identity values of protein-coding genes that can confidently be assigned to be orthologs, and uses this distribution to find groups of genes across genomes with ortholog-like mutual similarity. First, a set of high-confidence orthologs, called true conserved orthologous genes (trueCOGs) are identified based on two criteria: 1) they should be bidirectional best hits (BBH) between all genome pairs, and 2) their two flanking genes should also be BBHs between all genome pairs [167]. In other words, decRiPPter looks for sets of three contiguous genes that are highly conserved in both sequence identity and synteny among all analyzed genomes, using DIAMOND [168]. The center genes of these gene triplets are themselves conserved, and have conserved surrounding genes, making it highly likely that they are orthologous to one another. These center genes were therefore considered trueCOGs. While this list of trueCOGs contains high-confidence orthologs, the criteria for orthology set here are strict, and many orthologs are missed by only considering orthologs based on BBHs [169]. We therefore further expanded the list of homologs with ortholog-like similarity by dynamically determining a cutoff between each genome pair based on the similarity of the trueCOGs shared between those genomes. This cutoff is used to find all highly similar gene pairs. Considering that only sequence identity is used as a cutoff here, these gene pairs are either orthologs or paralogs. The identified gene pairs are then clustered with the Markov Clustering Algorithm (MCL [170, 171]) into 'clusters of orthologous genes' (COGs). The number of COG members found for each gene is divided by the number of genomes in the query to get a COG score ranging from 0 to 1, reflecting how widespread the gene is across the set of query genomes (Materials and Methods, Figure 2).

To validate our calculations, we analyzed the COG-scores of the highly conserved single-copy BUSCO (Benchmarking set of Universal Single-Copy Orthologs) gene set from OrthoDB [172-174], as well as the COG-scores of the genes in the gene clusters predicted by antiSMASH. In line with our expectations, homologs of the BUSCO gene set averaged COG-scores of 0.95 (Figure S2D), while the COG-scores of the antiSMASH gene clusters were much lower, averaging 0.311 +- 0.249 for all BGCs, and 0.234 +- 0.166 for RiPP BGCs (Figure S2C). While the COG-scoring method requires a group of genomes to be analyzed rather than a single genome, we believe that the extra calculation significantly contributes in filtering false positives (Table 1). In addition, the COG scores aid in the gene cluster identification based on the assumption that gene clusters are generally sets of genes with similar absence/presence patterns across species (Materials and Methods).

For the final criterion, the algorithm dereplicates the identified clusters by comparing them to known RiPP BGCs. All putative BGCs are clustered based on domain content and precursor similarity using sequence similarity networking [175], and compared to known RiPP BGCs from MIBiG [29, 140]. In addition, the overlap between predicted RiPP BGCs and gene clusters found by antiSMASH [39, 77] is determined (Figure 1).

decRiPPter identifies 42 candidate novel RiPP subclasses in *Streptomyces*

While RiPPs are found in many different microorganisms, their presence in streptomycetes reflects perhaps the most diverse array of RiPP subclasses within a single genus. Streptomycete*s* produce a broad spectrum of RiPPs, such as lanthipeptides [176], lasso peptides [45], linear azol(in)e-containing peptides (LAPs) [177], thiopeptides [46], thioamide-containing peptides [52] and bottromycins [97, 178, 179]. Their potential as RiPP producers is further highlighted by a recent study showcasing the diversity of lanthipeptide BGCs in *Streptomyces* and other actinobacteria [68]. Even though any genus or set of genomes can be analyzed by the decRiPPter pipeline, we hypothesized streptomycetes to be a likely source of novel RiPP subclasses, and sought to exhaustively mine it.

We started by running the pipeline described above on all publicly available *Streptomyces* genomes (1,295 genomes) from NCBI (Data S2). Due to computational limits, the genomes were split into ten randomly selected groups to calculate the frequency of distribution of each gene (COG-scores). In general, the number of genomes that could be grouped together and the resulting cutoffs were found to vary with the amount of minimum trueCOGs required (Figure S3A). To make sure that as many genomes as possible could be compared at once, we set the cutoff for minimum number of trueCOGs at 10. Despite the low cutoff, the distribution of similarity scores between genome pairs still resembled a Gaussian distribution (Figure S3B). The bimodal distribution of the resulting COG-scores showed that the majority of the genes were either conserved in only a small portion of the genomes, or present in

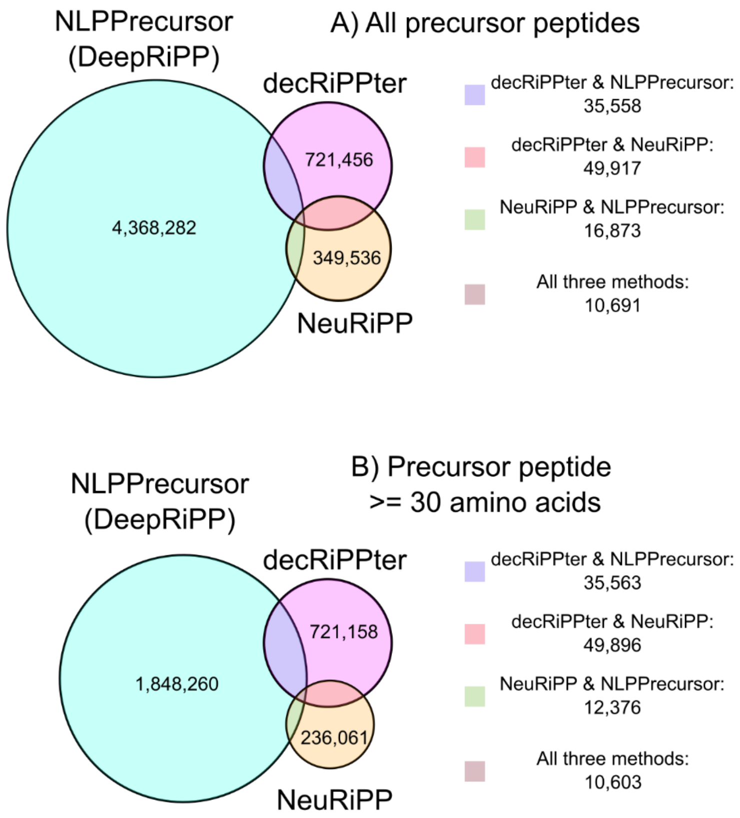


Figure 3. Three machine-learning-based RiPP precursor classifiers give highly different results. All small ORFs from the 1,295 *Streptomyces* genomes were classified by DeepRiPP's NLPPrecursor [89] module, NeuRiPP [88] and decRiPPter. The three tools have only a small overlap (10,691 hits). NLPPrecursor scored six times more hits as positive, and NeuRiPP roughly half when compared to decRiPPter. Many of these hits were very small ORFs (≤ 30 amino acids; (B)), though, while most of decRiPPters predicted precursors were larger than that. The exact accuracy of these tools cannot be determined, as it is unclear which of these hits are false positives, and which are hits in novel RiPP BGCs.

almost all genomes (Figure S3A). We then scored all predicted products of genes as well as predicted ORFs in intergenic regions shorter than 100 amino acids (total $7.19*10^{7}$) with the SVM-based classifier. While by far most of the queries scored below 0.5, a peak of queries scoring from 0.9 to 1.0 was observed (Figure S2B). Seeking to be inclusive at this stage, we set the cutoff at 0.9, resulting in $1.32*10^{6}$ candidate precursors passing this initial filter, thus filtering out 98.2 % of all candidates. Eliminating candidate precursors whose genes were completely overlapping reduced the number to 8.17*10^5 precursors (1.1 %). As a comparison, all ORFs were also analyzed by NLPPrecursor and NeuRiPP (Figure 3) [88, 89], and overlapping hits were removed as was done with decRiPPter's hits. For all three tools, a large number of candidate precursors were hits: NLPPrecursor scored the most (4.4*10^6), and NeuRiPP the least (4.3*10^5). Surprisingly, the three tools showed little overlap in positive hits (1,1*10^4). Considering that NLPPrecursor was parametrized for the detection of precursors of known subclasses and NeuRiPP appeared to be more strict (while our goal was to be more exploratory), we continued with decRiPPter's hits. In principle, the precursor-peptide-finding module of decRiPPter could easily be replaced by, e.g., NeuRiPP in future analyses for which this would be desirable.

We noticed that the majority of the precursor hits of decRiPPter were not found by Prodigal, but were extracted from intergenic regions (6.6*10^5 intergenic, 1.6*10^5 from Prodigal). A GC-plot analysis of 112 hits of both intergenic and Prodigal-detected genes showed that only 5-10% of the intergenic hits showed a GC-plot with clear distinctions between the first, second and third codon position, while the majority of Prodigal-detected genes had the same distinction (Figure S4). These intergenic regions are likely a source of many false positives, and for a more conservative approach one could choose to ignore intergenic hits altogether. Since our aim was to conduct an explorative study to detect novel subclasses, and gene-finding algorithms do frequently miss precursor genes, we chose to continue with all the precursors hits found here.

In our analyses, we found that the majority of RiPP BGCs contain the majority of biosynthetic genes on the same strand orientation as the precursor (MIBiG: 81.6%; antiSMASH RiPP BGCs: 73.1%). We therefore formed gene clusters using only the genes on the same strand as the predicted precursor. As a comparison, we divided all known RiPP BGCs and all antiSMASH RiPP BGCs found in the analyzed genome sequences into sections containing only adjacent genes on the same strand. The core section was defined as the section that contained the most biosynthetic genes as detected by antiSMASH or as annotated in the MIBiG database. These sections were used as validation sets to fine-tune distance and COG cutoffs for two gene cluster formation methods, which we called the 'simple method' and the 'island method'.

In the simple gene cluster method, genes were joined only using the intergenic distances as a cutoff. Using this method, we found that at a distance of 750 nucleotides, all MIBiG core sections were covered, and 91% of all antiSMASH core sections (Figure S5AB). However, using only distance may cause the gene cluster formation to overshoot into regions not associated with the BGC (e.g. Figure S1). We therefore created an alternative method, called the 'island method'. In this method, each gene is first joined with immediately adjacent genes that lie in the same strand orientation and have very small intergenic regions (≤50 nucleotides), to form islands. These islands may subsequently be combined if they have similar average COG-scores (Materials and Methods). We found that with this method, we could confidently cover our validation set, while slightly reducing the average size of the gene clusters (number of genes: 3.73 ± 3.75 vs 3.44 ± 3.53; Figure S5CDE). In addition the variation of the COG scores within the gene clusters decreased, suggesting that fewer housekeeping genes would be added to detected biosynthetic gene clusters (Figure S5F).

Overlapping gene clusters were fused, resulting in $7.18*10^5$ gene clusters. To organize the results, all gene clusters were paired to other gene clusters with similar protein domain content (Jaccard index of protein domains; cutoff: 0.5) and containing at least one predicted precursor gene with sequence similarity (NCBI blastp; bitscore cutoff: 30). These cutoffs were shown to distinguish between different RiPP subclasses (Figure S6). Clustering these pairs with MCL created 45,727 'families' of gene clusters, containing 312,163 gene clusters, while the remaining 406,105 gene clusters were left ungrouped.

Analysis of overlap between decRiPPter clusters and BGCs predicted by antiSMASH revealed that 5,908 clusters overlapped, constituting 78% of antiSMASH hits. The majority of BGCs previously detected by RODEO were also found to overlap (84%, Table S3). Most of the antiSMASH BGCs missed by decRiPPter belonged to the bacteriocin family, which do not necessarily encode a small precursor peptide (Table S3). The remainder of missed BGCs are likely due to precursor genes not being on the same strand as the genes encoding the biosynthetic machinery or due to precursor genes missed by decRiPPter's classifier.

Table 1. Correlation between the strictness of the filter used on the identified gene clusters and the saturation of RiPP BGCs. Genes were considered as being around the gene cluster if within five genes.

Filter	Filter details	Number of detected gene clusters	Gene clusters overlapping antiSMASH RiPP BGCs (percentage)
None	-	718,268	5,908 (0.8)
Mild	Gene cluster COG score: <= 0.25 In the gene cluster: • >= 3 genes • >= 2 biosynthetic genes In or around the gene cluster: • >= 1 transporter gene	21,419	1,678 (7.8)
Strict	Gene cluster COG score: <= 0.10 In the gene cluster: • >= 3 genes • >= 2 biosynthetic genes In or around the gene cluster: • >= 1 transporter gene • >= 1 regulatory gene • >= 1 peptidase gene	2,471	357 (14.4)

The hits overlapping with antiSMASH constituted only 0.8% of all decRiPPter clusters (Table 1, row 2). To further narrow down our results, we applied several filters to increase the saturation of RiPP BGCs in our dataset. A mild filter, limiting the average COG score to 0.25 and requiring two biosynthetic genes and a gene encoding a transporter, increased the fraction of overlapping RiPP BGCs to 7.8% (Table 1, row 2). When only clusters associated with genes for a predicted peptidase and a predicted regulator were considered, and the average COG score was limited to 0.1, the fraction increased further to 14.4% (Table 1, row 3). While many antiSMASH RiPP BGCs were filtered out in the process (and, by extension, many unknown RiPP BGCs were likely also filtered out this way), we felt our odds of discovering novel RiPP families were highest when focusing on the dataset with the highest fraction of RiPP BGCs, and therefore applied the strict filter. The remaining 2,471 clusters of genes were clustered as described above. Since our efforts were aimed at finding new gene cluster families, we discarded groups of clusters with fewer than three members, leaving 1,036 gene clusters in 187 families. Families in which more than half of the gene clusters overlapped with antiSMASH non-RiPP BGCs were discarded as well, leaving only known RiPP families and new candidate RiPP families (893 gene clusters in 151 families; Figure 4). While this step eliminated BGCs for hybrids of RiPP and non-RiPP pathways, we felt this filter was necessary to reduce the number of false positives in our dataset, especially considering the rarity of these hybrid BGCs.

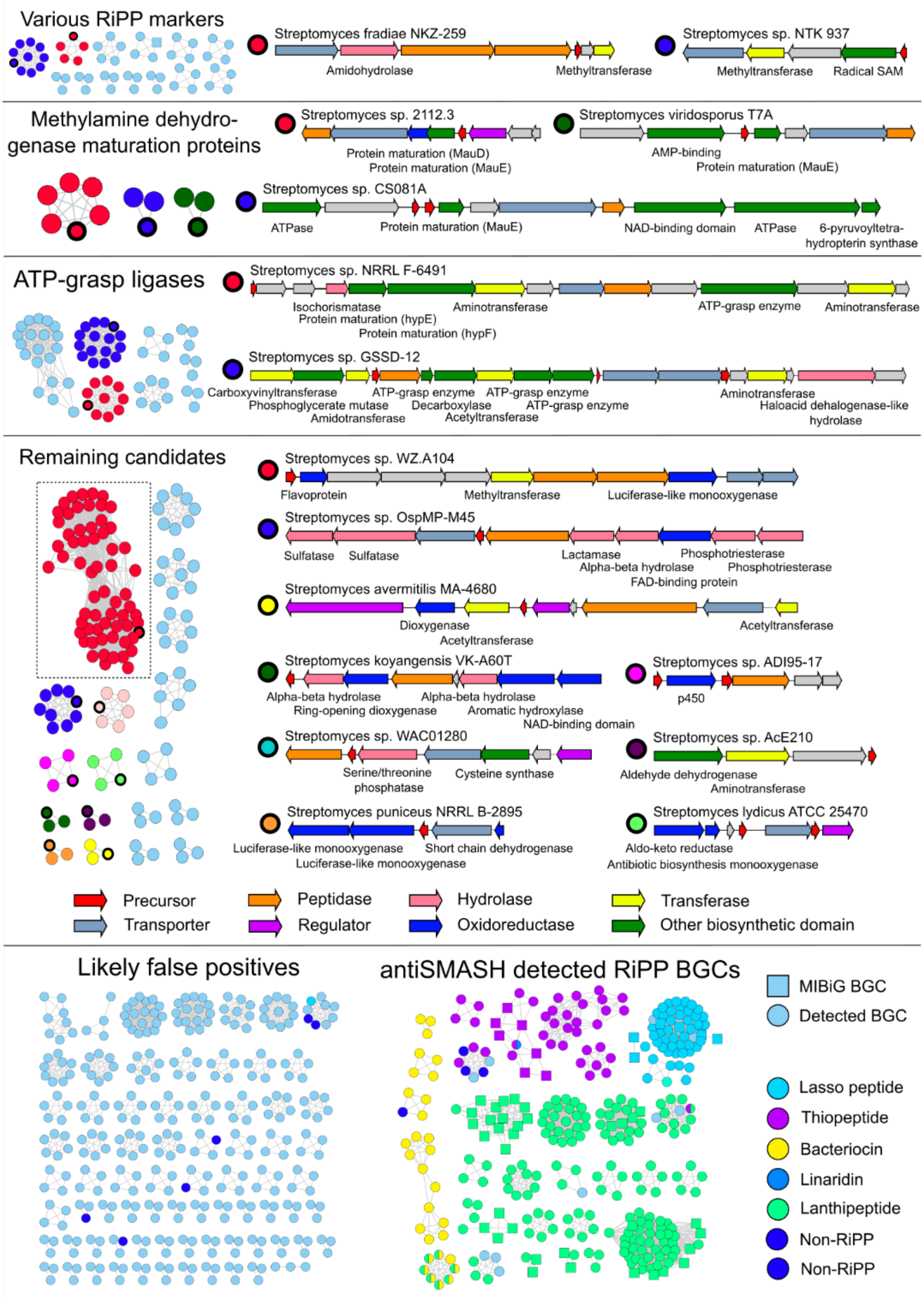
Roughly a third (280) of the remaining gene clusters were members of known families of RiPPs, including lasso peptides, lanthipeptides, thiopeptides, bacteriocins and microcins. In addition, many of the other candidate clusters (54) contained genes common to known RiPP BGCs, such as those encoding YcaO cyclodehydratases and radical SAM-utilizing proteins (Figure 4) These gene clusters were not annotated as RiPP gene clusters by antiSMASH, but the presence of these genes alone or in combination with a suitable precursor can be used as a lead to find novel RiPP gene clusters [52, 103].

Each remaining family of gene clusters was manually investigated to filter out likely false positives from the candidates. A set of general guidelines followed can be found in the Materials and Methods. Common reasons to discard gene clusters were functional annotations of candidate precursors as having a non-precursor function (e.g. homologous to ferredoxin or LysW [180]), annotations of multiple genes within a gene cluster related to primary metabolism (e.g. genes for cell-wall modifying enzymes), or other abnormalities (e.g. large intergenic gaps or very large gene clusters of more than 50 genes). Several modifying enzymes belonging to the candidate families were homologous to gene products involved in primary metabolism, such as 6-pyruvoyltetrahydropterin synthase or phosphoglycerate mutase. Given the low distribution (COG scores) of the genes encoding these enzymes, it seemed more likely to us that they were adapted from primary metabolism to play a role in secondary metabolism [160]. We therefore only discarded a gene cluster family if multiple clear relations to a known pathway were found. The remaining 42 candidate families, containing were further grouped together into broader families depending on whether a common enzyme was found (Figure 4).

Among our candidate families, a large group of families all contained one or more genes for ATP-grasp enzymes. ATP-grasp enzymes are all characterized by a typical ATP-grasp-fold, which binds ATP, which is hydrolyzed to catalyze a number of different reactions. These enzymes have a wide variety of functions in both primary and secondary metabolism, and their genes are present in a many different genomic contexts [181]. Involvement of ATP-grasp enzymes in RiPP biosynthesis has been reported for microviridin [83] and other omega-ester containing peptides (OEPs) [84], and for pheganomycin [162], where they catalyze macrocyclization and peptide ligation, respectively. The ATP-grasp enzymes involved in the biosynthesis of these products did not show direct similarity to any of the ATP-grasp ligases of these candidates, however, suggesting that these belong to yet to be uncovered biosynthetic pathways.

Among the candidate families were three families that contained homologs to *mauE*, and one that additionally contained a homolog of *mauD*. The proteins encoded by these genes are known to be involved in the maturation of of methylamine dehydrogenase, required for methylamine metabolism. MauE in particular has been speculated to play a role in the formation of disulfide bridges in the β-subunit of the protein, while the exact function of MauD remains unclear [182]. As no other orthologs of the *mau* cluster were found within the genomes of *Streptomyces* sp. 2112.3, *Streptomyces viridosporus* T7A or *Streptomyces* sp. CS081A, it is unlikely that these proteins carry out this function. Rather, the presence of these genes in a putative RiPP BGC suggests that they play a role in modification of RiPP precursors. Supporting this hypothesis, each of these gene clusters contained a gene predicted to a encode for a precursor containing at least eight cysteine residues (Table 2).

Similarly, homologs of *hypE* and *hypF* were detected in a gene cluster containing another gene encoding an ATP-grasp ligase. Genes encoding these proteins are typically part of the *hyp* operon, which is involved in the maturation of hydrogenase. Specifically, the two proteins cooperate to synthesize a thiocyanate ligand, which is transferred onto an iron center and used as a catalyst [183]. No other homologs of genes in the *hyp* operon were detected, suggesting that these protein-coding genes have adopted a novel function.


Various RiPP markers
Streptomyces fradiae NKZ-259
Amidohydrolase
Methyltransferase
Streptomyces sp. NTK 937
Methyltransferase
Radical SAM
Methylamine dehydrogenase maturation proteins
Streptomyces sp. 2112.3
Protein maturation (MauD)
Protein maturation (MauE)
Streptomyces viridosporus T7A
AMP-binding
Protein maturation (MauE)
Streptomyces sp. CS081A
ATPase
Protein maturation (MauE)
NAD-binding domain
ATPase
6-pyruvoyltetra-hydropterin synthase
ATP-grasp ligases
Streptomyces sp. NRRL F-6491
Isochorismatase
Protein maturation (hypE)
Protein maturation (hypF)
Aminotransferase
ATP-grasp enzyme
Aminotransferase
Streptomyces sp. GSSD-12
Carboxyvinyltransferase
Phosphoglycerate mutase
Amidotransferase
ATP-grasp enzyme
Decarboxylase
ATP-grasp enzyme
Acetyltransferase
ATP-grasp enzyme
Aminotransferase
Haloacid dehalogenase-like hydrolase
Remaining candidates
Streptomyces sp. WZ.A104
Flavoprotein
Methyltransferase
Luciferase-like monooxygenase
Streptomyces sp. OspMP-M45
Sulfatase
Sulfatase
Lactamase
Alpha-beta hydrolase
Phosphotriesterase
FAD-binding protein
Phosphotriesterase
Streptomyces avermitilis MA-4680
Dioxygenase
Acetyltransferase
Acetyltransferase
Streptomyces koyangensis VK-A60T
Alpha-beta hydrolase
Ring-opening dioxygenase
Alpha-beta hydrolase
Aromatic hydroxylase
NAD-binding domain
Streptomyces sp. ADI95-17
p450
Streptomyces sp. WAC01280
Serine/threonine phosphatase
Cysteine synthase
Streptomyces sp. AcE210
Aldehyde dehydrogenase
Aminotransferase
Streptomyces puniceus NRRL B-2895
Luciferase-like monooxygenase
Luciferase-like monooxygenase
Short chain dehydrogenase
Streptomyces lydicus ATCC 25470
Aldo-keto reductase
Antibiotic biosynthesis monooxygenase
Precursor
Transporter
Peptidase
Regulator
Hydrolase
Oxidoreductase
Transferase
Other biosynthetic domain
Likely false positives
antiSMASH detected RiPP BGCs
MIBiG BGC
Detected BGC
Lasso peptide
Thiopeptide
Bacteriocin
Linaridin
Lanthipeptide
Non-RiPP
Non-RiPP

As stated above, 175 gene cluster families, containing a total of 1,036 candidate gene clusters, were left after the strict filter. Of these, 24 families containing 143 gene clusters were removed due to overlap with non-RiPP BGCs. An additional 74 families containing 341 gene clusters were removed by manual curation, making for a total false positive count of 98 families containing 484 gene clusters, just under half of the total (46.7%). A total of 32 families containing 280 gene clusters overlapped with known RiPP BGCs (27.0%), which can be considered true positives. The remaining 272 gene clusters (42 families; 26.3%) are the presented candidates. This means that the actual true positive rate lies between 27.0% and 53.3%, and the false positive rate between 46.7% and 73.0%, depending on the nature of the candidates. For the results from the mild filter, 1,678 gene clusters out of 21,419 were overlapping with known RiPP BGCs (7.8%). How many of the remaining gene clusters (92.2%) are false positives and how many are novel RiPP BGCs can not be determined without a thorough manual examination. From the results of the strict filter, however, it appears there are roughly as many novel RiPP BGCs as there are known ones (272 vs 280). Extending thes ratios to the results of the mild filter would mean that an additional 1,678 gene clusters are novel RiPP BGCs, resulting in an estimated true positive rate of 15.6% and false positive rate of 84.4%. These high false positive rates emphasize that one should interpret the results with caution. However, if even half of the proposed candidates are true RiPP subclasses, this would represent a significant contribution to the total amount discovered.

Figure 4 (opposite page). decRiPPter finds 42 candidate RiPP families with a large variety of encoded modifying enzymes and precursors. Gene clusters found in 1,295 *Streptomyces* genomes were passed through a strict filter and grouped together. Each node of the network represent a candidate BGC, while edges represent similarity in both precurs and enzyme domains. The four panels at the top contain families of interest, grouped by common defining characteristics, if present (top panel: 54 gene clusters in 13 families; second panel: 12 gene clusters in 3 families; third panel: 65 gene clusters in 8 families; fourth panel: 141 gene clusters in 18 families). The bottom panel contains the gene clusters marked as likely false positive (left side, 341 gene clusters in 74 families) and the gene clusters overlapping with antiSMASH-detected RiPP BGCs (right side, 280 gene clusters in 33 families). Examples of 15 gene clusters of candidate families are given (nodes with dark circles). Arrow colors indicate enzyme family of the product, and the description of the putative gene products is given below the arrows. The candidate RiPP family represented by the network outlined with a dashed box is discussed further in Chapter 4.

Table 2. Precursor sequences of selected BGCs of candidate RiPP families shown in Figure 4. Serine and threonine residues are marked in green, and cysteine residues are marked in red.

Family	Strain	Precursor sequence
Known RiPP markers	*Streptomyces* sp. NTK 937	MTENTAPEESPEVEAHSAADDAAQAPEQFHDAAEIICGVYDKEIQV
Known RiPP markers	*Streptomyces fradiae* NKZ-259	MPSGMPNDPSTTDGLSRRRVLGTAAAAAVPLPARGAEDAEAKSGPW
Containing MauE	*Streptomyces viridosporus* T7A	MSRALESLSSRLLGLFVPKVEAAASAQACQCFNECWQCARSACCVNTYCGSINCWRSCPGC
Containing MauE	*Streptomyces* sp. CS081A	MARTVGDGSKGCRPSPVSPYGLDQYGDRAASTWGASSATCGVRGEP
		MVKSLSALAGRAFARVLPQETAAAACACPAGSSSWCSGENLYTRFCCSWNCAAKPTCTVTVVYGAC
Containing MauE	*Streptomyces* sp. 2112.3	MFKKLEAVGSALLERLVPRVDASACGTNCWNDCWQCAHSACKVNTCTGALTCLSGNC
ATP-grasp ligases	*Streptomyces* sp. NRRL F-6491	MARAARNLLAITASAALSFLLVQGTGAQEERAFLAGSGQGKVINDLGWG
ATP-grasp ligases	*Streptomyces* sp. GSSD-12	MSSDPSDAAEQGPVGGFITEPLVAAAATTGGCCGEPRSAPEPARSSCCGEPAAEEAPRSCCGEPAAAG
		MADDMIGSGCCETSGNEDVAEDGTECGCACACCD
		MSETSLGNMFWNAAQQPPAATAEEPKKASSCCGPKPEAKAPAEQAAAPEKASSCCGPKPAAAAEPEGTPAPKKSSCCG
Other	*Streptomyces* sp. WZ.A104	MQNVTEKDLFDGYTAYTSAEELGLHDGKEAAPAFSPTIPWAIRATIISARSSQQCAAALGSLAAKTVENKC
Other	*Streptomyces* sp. OspMP-M45	MTEAGLWEEGDAGRRRPLGVPPENWPVPGGRQGMDGQWSGQSSKTIDHPGGAT
Other	*Streptomyces avermitilis* MA-4680	MSSLDKPGRKKWSGPEKWQVILAASSLGVAVVALVGQFAQFL
Other	*Streptomyces koyangensis* VK-A60T	MGDLDEEVAAPGPGRWIRPSSTAGYGWTTSCRTSVFPPASPDSCQARETVTWCAWVP
Other	*Streptomyces* sp. ADI95-17	MNSLSEAGCWCHERLKSCPSECKFRVKDGGAVMKFLFLLKDKMTPEKSLKAYAWYHWY
		MCEVCRSSRNPGPWGGCCGDGARLGHGWPVSYYETLLCKSQPHEGLDLGASIGEGFEPTPGDLPAGGQSPHKE
Other	*Streptomyces* sp. WAC01280	MLKGGQLGRFSTNSMNDHREQLGIGPPCLLTFDNAARSSQPSQEAAPCARAES
Other	*Streptomyces* sp. AcE210	MAESPTPEAVAEQPTEVAQPHRLVLLGACGCGSGCGCGCQSGAPCQCGGCSG
Other	*Streptomyces puniceus* NRRL B-2895	MRTAAAYASGEPPPVAVVKSHGVAFENRVRYVSPVPSTTHAAASAPGSAEGSAPAATA
Other	*Streptomyces lydicus* ATCC 25470	MLWKSCARARCGISIPWNSFEFDHGGTGVVPCVPGVCEFPARDGKEEVT
		MNQGGGEQRGAEVSIRANVGSWLAVRKSPFEAGGSPVSRWEDLPRGVPCPYETGAHQD

All candidate gene clusters presented here carry the features we selected, typical of RiPP BGCs: a low frequency of occurrence among the scanned genomes, a suitable precursor peptide, candidate modifying enzymes, transporters, regulators and peptidases. However, many known RiPP BGCs were removed, suggesting that there may be more uncharacterized RiPP families among the gene clusters we discarded. While the complete dataset could not be covered here, the command-line application of decRiPPter has been set up to allow users to set their own filters. The pipeline can be run on any set of genomes. We recommend choosing a set of genomes that are sufficiently closely related to share a `core genome` for the COG-score calculations. At the same time, genomes should not be too similar, so that a wide variety of BGCs can be found among them that show variability in their presence/absence pattern across genomes. decRiPPter runs are visualized in an HTML output, in which the results can be further browsed and filtered by Pfam domains and other criteria, allowing users to find candidate families according to their preferences. The results from this analysis of the strict and the mild filter is available at https://decrippter.bioinformatics.nl.

Conclusion and final perspectives

The continued expansion of available genomic sequence data has allowed for discovery of large reservoirs of natural product BGCs, fueled by sophisticated genome mining methods. These methods must make tradeoffs between novelty and accuracy [26]. Tools primarily aimed at accuracy reliably discover large numbers of known natural product BGCs, but are limited by specific genetic markers. On the other hand, while tools aimed at novelty may lead to the discovery of new natural products, these tools have to sacrifice on accuracy, resulting in a larger amount of false positives.

Here, we take a new approach to natural product genome mining, aimed specifically at the discovery of novel types of RiPPs. To this end, we built decRiPPter, an integrative approach to RiPP genome mining, based on general features of RiPP BGCs rather than selective presence of specific types of enzymes and domains. To increase the accuracy of our methods, we base detection of the RiPP BGCs on the one thing all RiPP BGCs have in common: a gene encoding a precursor peptide. With this method, we identify 42 candidate novel RiPP families, mined from only 1,295 *Streptomyces* genomes. These families are undetected by antiSMASH, and show no clear markers identifying them as belonging to previously known RiPP BGC subclasses. While the approach to RiPP genome mining taken here inevitably gives rise to a higher number of false positives, we feel that such a 'low-confidence / high novelty' approach [26] is necessary for the discovery of completely novel RiPP subclasses. Additionally, users are able to set their own filters for the identified gene clusters, allowing them to search candidate RiPP subclasses containing specific enzymes or enzyme types within a much more confined search space compared to manual genome browsing. As such, decRiPPter can function as a platform for explorative RiPP genome mining, enabling a large variety of different search strategies to explore further into RiPP chemical space.

Materials and Methods

decRiPPter pipeline

Genome data preparation

As input, decRiPPter uses a set of genomes from species that are part of the same taxonomic group (e.g., genus, family), which it requires for its comparative genomic analyses. decRiPPter downloads genomes from NCBI [184] based on NCBI taxonomic identifiers of species, genera or higher orders of classification. Additional requirements for level of assembly (e.g. "Representative genome") can also be given. decRiPPter can reannotate genomes with prodigal 2.6.3[71], and automatically does so when DNA FASTA files are given as input. In addition, users may analyze their own genomes, in isolation or in conjunction with downloaded genomes.

SVM-based classifier

To predict RiPP precursors, we first collected positively and negatively labeled training data. The positive training data was collected from MIBiG [140] and recent literature, resulting in 175 RiPP precursors across ten subclasses. For the negative training set, we generated a set of 20,000 short non-precursor sequences. Half of these were randomly selected from a set of 35,000 short proteins (<175 amino acids long) from Uniprot (queried June 2014) that were not similar to RiPP precursors based on an NCBI blastp search. The other half were randomly selected from a set of 17,000 translated intergenic sequences between a stop codon and the next start codon of sizes 30-300 nt taken from 10 genomes across the bacterial tree of life: Escherichia coli, Bacillus subtilis, Streptomyces coelicolor, Bacteroides fragilis, Rhizobium etli, Chloroflexus aurantiacus, Synechococcus sp. PCC 7002, Opitutus terrae, Acidobacterium capsulatum and Pirellula staleyi. For all sequences from both the positive and negative training sets, we computed several physio-chemical properties, such as its length, hydrophobicity, charge, counts of canonical amino-acid residues and classes of amino acids, and highest counts of, e.g., cysteines and serines within contiguous blocks of 20 or 30 amino acids. The method for computing these properties is part of the decRiPPter pipeline, and can be found in the code repository, at https://github.com/Alexamk/decRiPPter/blob/master/lib/features.py. All training data and data collection scripts are available online (https://zenodo.org/record/3834818#.X7JmIOTsbvs)

We then utilized Scikit-Learn implementations of several different supervised machine-learning algorithms. We varied several parameters associated with a given algorithm (e.g., kernel functions, penalty parameters, penalty functions, etc.). Furthermore, we mapped the accuracy as a function of scaling the dataset or changing class weights to take into account the unbalanced dataset (only ~1% of gene clusters in our dataset represent known RiPPs). The RiPP cluster classification accuracy of each combination of scaling, algorithm, and the corresponding set of parameters was evaluated using accuracy and area under receiver operating characteristics (ROC) curve, and leave-one-class-out cross-validation. SVMs with three different kernel functions were trained: two with polynomal kernel function (SVM3: 3rd degree, coef0 of 2.154, kernel coefficient gamma of 2.78*10^-2, regularization parameter C of 0.158; SVM4: 4th degree, coef0 of 2.154, kernel coefficient gamma of 4.64*10^-3, regularization parameter C of 25.119) and one with a radial basis function kernel (SVMr: kernel coefficient gamma of 1*10^-5, regularization parameter C of 6.310*10^5). For each type, one SVM was trained with all training data, while eighteen more were trained by leaving out the sequences of one RiPP subclass from the positive training data at a time. The average of all 57 SVMs was taken as the final SVM score.

COG scores calculation

To calculate the relative frequency of occurrence of each gene, we constructed a pipeline to find all groups of homologous genes (Figure 2). In the first step, protein-coding genes for which orthology can confidently be assigned are grouped into Clusters of Orthologous Groups (COGs). All proteins are aligned to one another using DIAMOND [168], and all bidirectional best hits (BBHs) are identified that share at least 60.0% similarity (Figure 2A). We established two requirements for genes to be confidently annotated as orthologs, based on recent papers [167, 169]: 1) they should constitute BBHs, and 2) their immediate genomic surroundings should be conserved, i.e. the two flanking genes should also be bidirectional best hits between the two genomes. Genes fulfilling these two criteria are paired together, resulting in groups of orthologous genes. Among these groups, decRiPPter then selects those that are completely conserved across all genomes: each group should contain at least one ortholog in each genome, and all orthologs in the group should all fulfill the same requirements for each genome pair. These groups are considered true Clusters of Orthologous Genes (trueCOGs; Figure 2B).

In the second step, a cutoff for protein-coding gene sequence identity is determined for each genome pair, in order to separate orthologs as well as recently evolved paralogs from more distantly related homologs. For any given pair of genomes, the distribution of sequence identities of all gene pairs of their trueCOGs is calculated. The cutoff is then calculated as the average percentage identity, minus three times the standard deviation (Figure 2C). Any two aligned genes with a percentage identity higher than this cutoff are considered to be functionally closely related to one another and paired up. The resulting groups of homologous genes were clustered with the Markov Cluster Algorithm[170, 171] (Figure 2D). From these groups, the relative frequency of occurrence of groups of homologous genes across all query genomes is calculated, called the COG-score (Figure 2E).

In cases when insufficient numbers of trueCOGs (<= 10) could be found in our analyses (because the set of genomes was too diverse, and/or contained too many draft genomes that each miss some of the trueCOGs), the genomes were rearranged into smaller subgroups. We used two general rules to create the groups: 1) Groups should be as large as possible, so that trueCOGs found are conserved across many species, and represent conserved widespread genes. 2) Genomes should be compared to as many other genomes as possible, so as not to introduce bias into the calculation of the COG-score. To fulfil both requirements, partially overlapping subgroups were formed, with the goal of letting each genome be a part of a collection of subgroups that together covered as many of the genomes as possible. To form the subgroups, a pair of genomes with the highest number of trueCOGs was used as a seed, and genomes were added one at a time until the number of trueCOGs dropped below the set cutoff. All the genomes in the group were said to be linked together by this group. The process of group formation was then repeated, starting with genomes for which no group had yet been formed. If all genomes were already part of at least one subgroup, the genomes were selected which were linked to the fewest genomes via the groups they were part of. The process was terminated when adding additional groups did not increase the number of links between genomes for several successive iterations.

Gene cluster formation

In this stage, decRiPPter identifies putative operon-like gene clusters around each candidate precursor peptide-encoding gene, by either of two different methods (Figure S1): In the first method, called the simple method, genes in the same strand orientation as the candidate precursor peptide-encoding gene are added to the putative gene cluster if the intergenic distance to the previous gene is within a given cutoff. The second method, called the island method, uses both intergenic distance and levels of conservation (COG-score) to determine the gene clusters. First, all genes in the same strand orientation within 750 nucleotides of one another are identified and then grouped into islands. Within islands, genes should be almost directly adjacent (intergenic distance: <= 50 nucleotides). We then fused the islands together using the COG-scores (see above), building on the assumption that genes in a gene cluster should all have similar levels of conservation. Islands were fused together if the average of their COG-scores was within a set range (0.1 plus the sum of the standard deviations of both islands). Not all gene families have similar COG scores when they occur within the gene clusters thus formed; e.g., genes encoding ABC-transporters frequently have close relatives in other biomolecular systems and therefore often have higher COG scores. Hence, to counteract gene cluster formation breaking off prematurely, up to two outlier genes are allowed when fusing islands, if, after adding the outliers, more islands can be added that are within the range for COG-score deviation. Intergenic distances and cutoffs were iteratively finetuned to ensure gene clusters in known RiPP BGCs would be effectively found. Finally, gene clusters that overlap or lie within 50 nucleotides of one another are fused together.

Annotation

For purposes of data exploration (annotation and visualization), each gene cluster is extended to include the 5 flanking genes on either side, and all encoded proteins in the extended gene clusters are annotated with Pfam 31.0 [75] and TIGRFAM [76]. Lists were compiled of all TIGRFAM and Pfam domains associated with either peptidases, transporters, regulators, using a combination of keyword searches on the Pfam and TIGRFAM websites, combined with manual curation. A list of protein domains associated with biosynthetic activity was constructed by linking Pfam domains to E.C. numbers, using InterPro mappings [165]. Biosynthetic TIGRFAM domains were taken directly from the database. Each domain linked to an E.C. number was assumed to have enzymatic activity. The biosynthetic domain list was further expanded with domains used in the ClusterFinder [41] algorithm that were indicative of a biosynthetic gene cluster. The resulting lists are used by decRiPPter to mark proteins either as a regulator, peptidase, transporter or biosynthetic enzyme, in that order, by seeing if any of the identified domains overlapped with the domains in the precompiled lists (Data S1).

Clustering

To cluster the detected gene clusters, the distance between them is calculated in two different ways: 1) amino acid sequences of candidate precursor peptide-encoding genes in the gene clusters are aligned with NCBI BLAST blastp [56] (cutoff: 30 bitscore), and 2) the content of the gene clusters is compared by calculating the Jaccard index of their constituent protein domains (cutoff: 0.5). Gene clusters are paired only if they are paired by both methods. The distance between paired gene clusters is calculated as the average between the Jaccard index and the percentage identity of the aligned precursors. Finally, pairs are clustered using MCL.

Overlap with antiSMASH

Overlap with antiSMASH was determined using antiSMASH 4.0 [77] run in minimal mode.

Availability

The decRiPPter pipeline is available at https://github.com/Alexamk/decRiPPter/. Data from the analysis discussed here is available at https://decrippter.bioinformatics.nl.

Data analysis

Comparison with NeuRiPP and NLPPrecursor

NeuRiPP classifications were performed using the parallel CNN network with the network weights provided by the author [88]. NLPPrecursor was installed and executed with default settings [89]. All open reading frames were analyzed with both methods, and completely overlapping precursor hits on the same frame were removed, as in the decRiPPter pipeline.

Supplementary information for Chapter 3

Data S1. Categorized Pfam and TIGRFAM domains used in decRiPPter pipeline. Available from https://github.com/Alexamk/decRiPPter/tree/master/data/domains/.

Data S2. *Streptomyces* genomes analysed with decRiPPter. Available upon request.

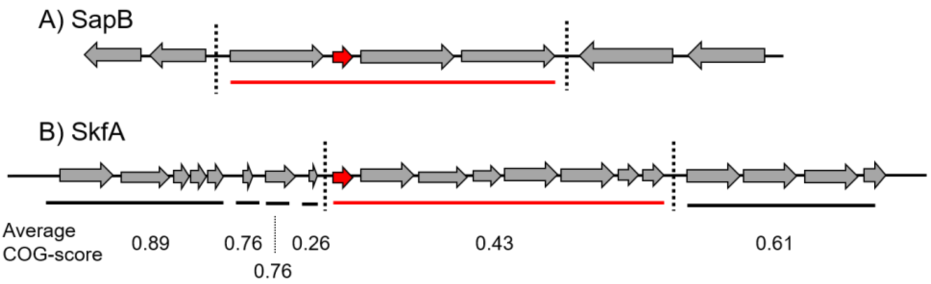


Figure S1. decRiPPter forms putative gene clusters around candidate precursor peptide-encoding genes. Two examples are provided here to illustrate identification of putative gene clusters in decRiPPter. A) In the *sapB* gene cluster, four genes form the main BGC. These four genes are sequential, share the same strand orientation and lie within a small distance of one another (<= 50 nt). They are therefore fused together into a single gene cluster. The flanking genes are on opposite strands, and therefore not considered. B) The *skfA* BGC consists of eight genes sequential genes that share the same strand orientation. However, it is flanked by several other genes that also share the same strand orientation, within relatively short intergenic distances (<= 200 nucleotides). Using the island method, the genes are first fused into six islands, within 50 nucleotides distance of one another (indicated by lines underneath the genes). These islands may then be fused depending on the COG-score, which does not happen here because the difference is too large. The result is that the flanking genes, with a too high COG-score, are not added, and the correct BGC remains.

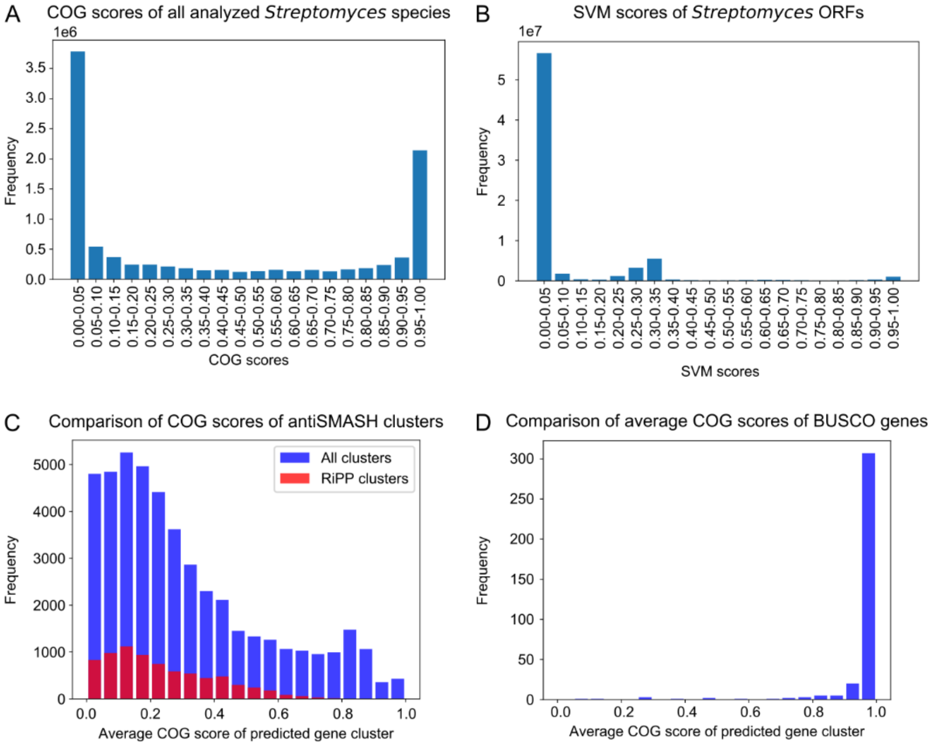


Figure S2. COG and SVM scores in all analyzed 1,295 *Streptomyces* genomes. A) COG scores of all genes in all 1,295 analyzed *Streptomyces* genomes. A high COG score indicates presence of homologs in many different genomes, while a low COG score indicates a more infrequent distribution. COG scores were calculated as described in the methods. B) Distribution of the scores assigned by decRiPPter's SVM-based classifier. A total of 7,1 * 10^7 small ORFs were analyzed. C) Comparison of COG scores of antiSMASH-detected gene clusters. COG scores were averaged over all genes in the predicted gene clusters. COG scores averaged 0.311 +- 0.249 for all gene clusters, and 0.234 +- 0.166 for RiPP gene clusters. D) Comparison of average COG scores of BUSCO genes. The average of each BUSCO [173, 174] gene was calculated for each genome analyzed.

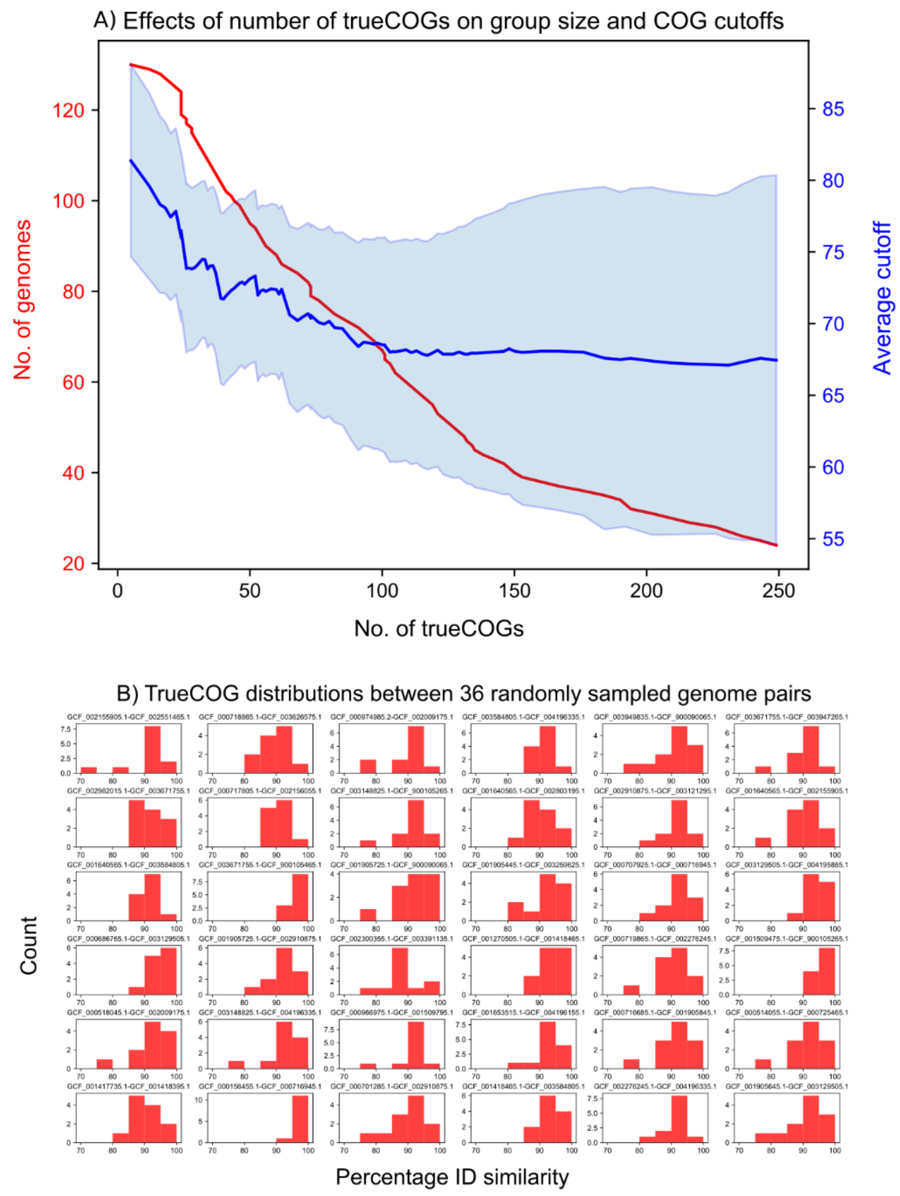


Figure S3. COG-scores calculations depend on genome group size. A) As the minimum number of trueCOGs increases, the number of genomes that can be analyzed together (red line) decreases. In addition, the average COG cutoff (blue line) decreases when more trueCOGs are added, and the spread of COG cutoffs (shaded area; average cutoff +- the standard deviation) increases, suggesting that additional trueCOGs that were added were less conserved and showed higher variability in sequence similarity. B) TrueCOG distribution between 36 randomly sampled genome pairs. Based on these distributions, COG cutoffs were determined.

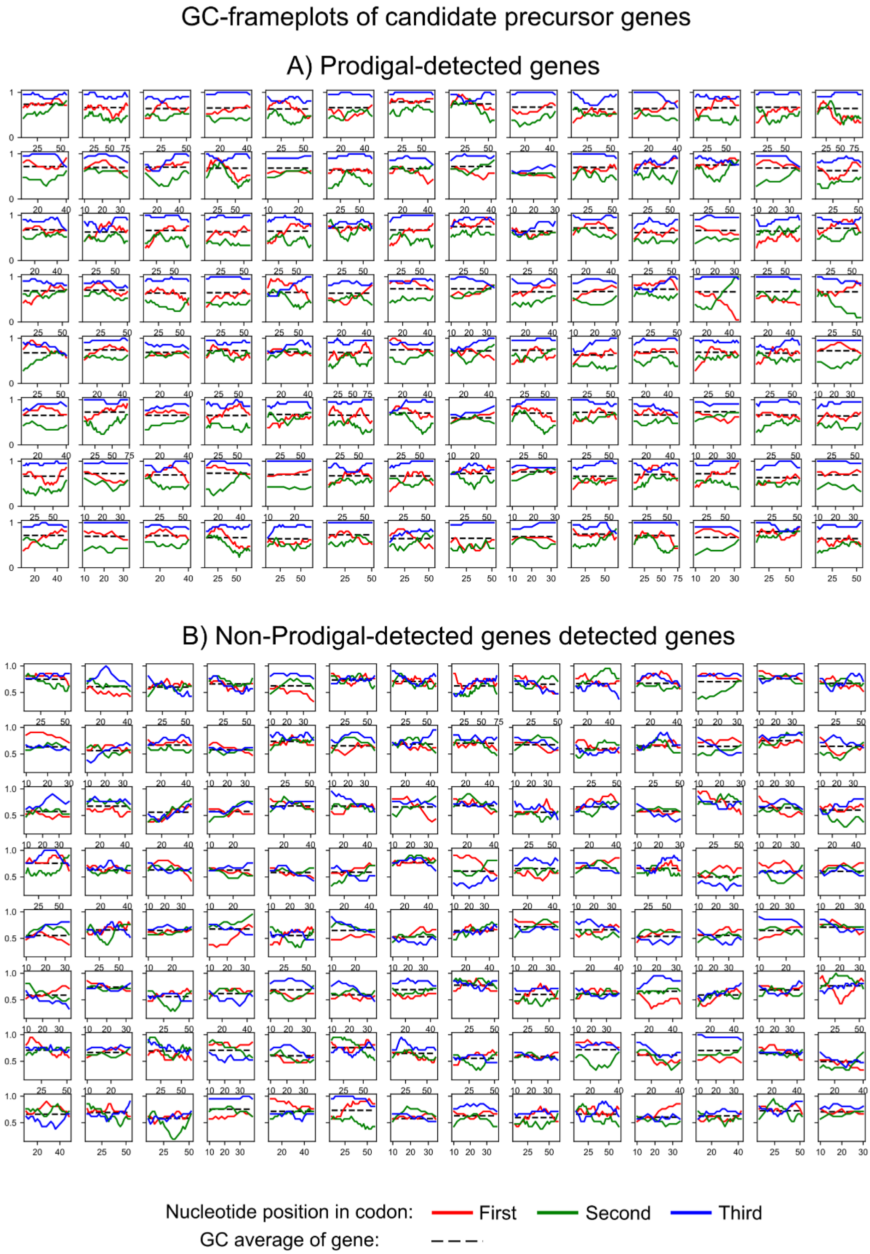


Figure S4. GC-plots of randomly sampled Prodigal-detected precursor hits (A) and intergenic precursor hits (B). GC values are shown as the moving average of the first, second and third positions, using a window-size of 5 and a step-size of 2. Only a small percentage of intergenic hits showed clear distinction between the three moving averages as in the Prodigal-detected hits, suggesting the majority of these are not encoding genes.

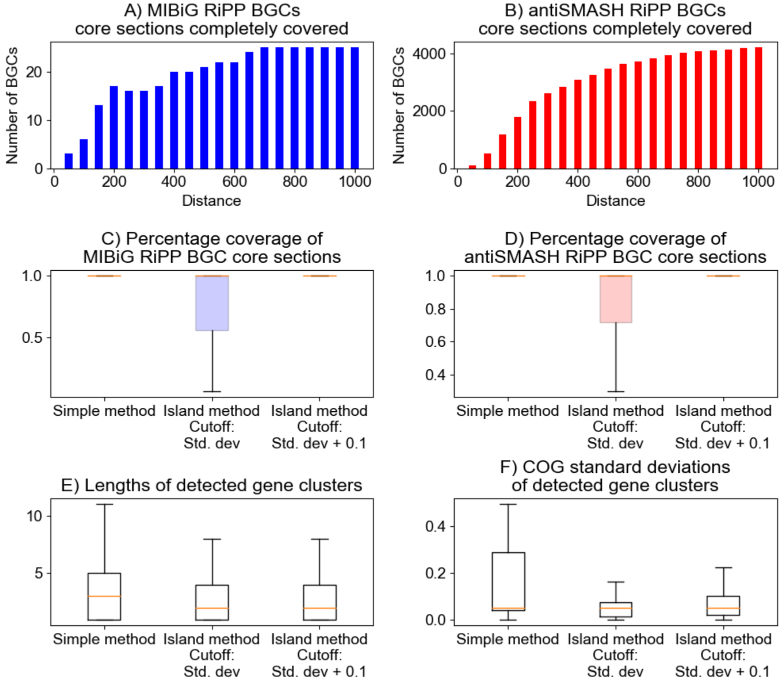


Figure S5. Gene cluster formation effectively covers antiSMASH and MIBiG BGC core gene sections. In the simple gene cluster formation method, genes are sequentially added as long as they are in the same strand orientation, within a certain distance. At a distance of 700 nucleotides, all MIBiG core gene sections are covered (A), as well as 91% (3947/4321) of antiSMASH core gene sections. (B). In the 'island method', genes are first fused into islands, which may be further fused if their average COG-scores are within a cutoff. Using just the standard deviation of the islands as a cutoff resulted in incomplete coverage of both the MIBiG and the antiSMASH core sections (C, D, middle boxes). Increasing the cutoff to the standard deviation plus 0.1 resulted in comparable coverage (C, D, right boxes) of these sections when compared to the simple method (C, D, left boxes). In addition, the overall gene cluster length (E) and variation of COG scores (F) within all formed gene clusters decreased.

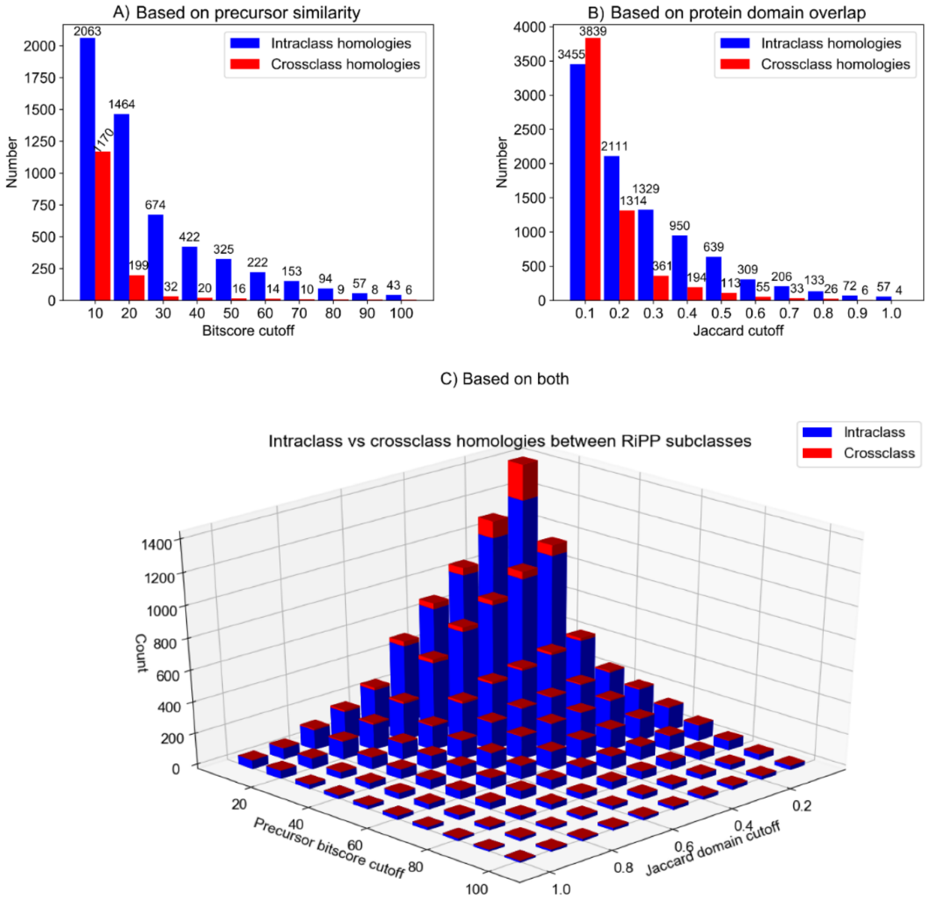


Figure S6. Combining precursor similarity with domain similarity is an effective strategy to group RiPP subclasses. Starting at precursor similarity bitscore cutoffs of 20 and Jaccard scores of overlapping protein domains found in MIBiG RiPP BGCs of 0.4, the number of intraclass homologies is larger than the number of crossclass homologies. Combining the two methods greatly decreases the number of cross-class homologies found, proving it as an effective method to group RiPP BGCs of different subtypes.

Table S1. RiPP classes in positive training data of decRiPPter.

RiPP class	Amount of precursors
Bottromycin	3
Cyanobactin	14
Glycocin	1
head-to-tail cyclized peptide	10
Lanthipeptide	79
LAP	4
Hybrid	4
lasso peptide	13
Linaridin	2
Microcin	7
Microviridin	4
Proteusin	1
Sactipeptide	4
Thiopeptide	12
Unclassified	17

Table S2. decRiPPter detects most RiPP precursors of known classes found by RODEO. RODEO results were extracted from previous studies [55, 72-74, 86].

RiPP Class	Number detected by RODEO	Scored ≥ 0.9 by decRiPPter
Lanthipeptide	453	329
Lasso peptide	5270	3738
Linaridin	2152	1127
Sactipeptide/ranthipeptide	1524	953
Thiopeptide	399	387
Total	9798	6534

Table S3. Comparison of detected BGCs with antiSMASH and RODEO. Note that not all genomes were analyzed by RODEO. Results from earlier RODEO genome mining [55, 72-74, 86] where only used if within the 1,295 *Streptomyces* genomes.

RiPP Class	RODEO BGCs	Overlap (no filter)	Overlap (mild filter)	Overlap (strict filter)	antiSMASH BGCs	Overlap (no filter)	Overlap (mild filter)	Overlap (strict filter)
Lanthipeptide	1530	1447	421	102	2768	2570	850	175
Lasso peptide	397	175	112	14	878	742	315	59
Linaridin	97	85	33	4	229	199	82	5
Thiopeptide	71	45	23	4	612	584	264	57
Sactipeptide/ ranthipeptide	1	1	1	0				
Bacteriocin					2735	1402	184	41
Bottromycin					2	2	0	0
Cyanobactin					31	27	3	1
Proteusin					2	2	2	0
RiPP hybrid					321	312	96	32