

SI Appendix

List of supplementary datasets	2
Supplementary results	3
Supplementary methods	5
Supplementary figures	15
References	72

List of supplementary datasets

Dataset S1: Curated dataset of 136 known RiPP clusters linked to 161 small molecules.

Dataset S2: Curated dataset of all known RiPPs, along with the corresponding producing organism.

Dataset S3: Accuracy metrics of NLPPrecursor model 1 (DL1) which predicts the probability of a given ORF to be RiPP or non-RiPP.

Dataset S4: Accuracy metrics of NLPPrecursor model 2 (DL2) which predicts all possible cleavage sites within candidate RiPP precursor peptides.

Dataset S5: All identified RiPPs by NLPPrecursor and RiPP-PRISM among 65,421 publicly available prokaryotic genomes.

Dataset S6: Known post-translational modifications of all RiPP classes as recognized by GRAPE.

Dataset S7: Details of media and growth conditions used for bacterial fermentation to generate the extract database.

Supplementary results

To position our work in the context of these recent developments in genome mining for RiPPs, we provide a partial comparison of NLPPrecursor to two recently described algorithms, RODEO [1] and RiPPMiner [2], which implement new methods for certain aspects of genome-guided RiPP discovery.

Comparison to RODEO

RODEO has been developed with the primary goal of identifying elusive lasso peptide precursor peptides[1]. While the scope of DeepRiPP, and even NLPPrecursor, is much broader, we sought to compare NLPPrecursor's ability to differentiate valid lasso peptide precursor ORFs from invalid ones using the dataset released by Tietz et al. (2017). This dataset was chosen as it is the largest available set of lasso peptides and suitable negative examples with a total of 9,983 samples (1,315 lasso peptides and 8,668 negative examples). However, because RODEO takes the genomic context of a putative lasso peptide precursor as input to generate predictions, whereas NLPPrecursor does not, we first dereplicated the dataset to include only unique ORF sequences, resulting in 840 lasso peptides and 6,973 negative examples respectively. According to the Methods described by Tietz et al. [1], the class labels of the peptides in this dataset were amended according to the SVM scores themselves, and subsequent analyses, suggesting that the dataset may be biased towards the success of RODEO. To analyse this dataset using DeepRiPP, each of the contigs on which putative precursors were encoded on the dataset were downloaded, and ORFs were identified via Prodigal. If no ORFs were found within 50 bp of the suggested start site, a score of zero was assigned. The ORFs were analysed by the class prediction module of NLPPrecursor. If the top prediction was not a lasso peptide, a score of zero was assigned, otherwise the class prediction probability was used. The results suggest that RODEO does perform better than NLPPrecursor in identifying lasso peptide precursors (Fig. S5), although it is important to note the potential for bias in this dataset towards the success of RODEO, while NLPPrecursor was trained on a dataset with many fewer lasso peptides.

To provide a more fair comparison, we decided to retrain NLPPrecursor specifically for the same task as RODEO using the same dataset and training-test split strategy as identified in their original manuscript. Here, while NLPPrecursor does perform better, it performs close to but slightly below the predictive accuracy of RODEO (AUC 0.987 vs 0.999; $p = 0.027$, two-sided DeLong test; Fig. S5). It is important to note these results suggest that the majority of predictive accuracy for discriminating precursor peptides can be identified from purely protein sequence alone, without taking into account genomic context information as proposed by Tietz et al. These results further demonstrate the utility of a RiPP precursor peptide learning framework that can be adapted to a variety of classification tasks.

Comparison to RiPPMiner

RiPPMiner [2] presents three major processes that can be directly compared to DeepRiPP using ORF sequence alone. First, RiPPMiner classifies ORFs as either being a RiPP precursor

or not; second it classifies RiPP precursor ORFs according to their subclass; and finally, it predicts cleavage sites for RiPPs. To provide a fair comparison, our dataset was used, where 90% of the data was reserved for training and validation while the remaining 10% was used to compare RiPPMiner and NLPPrecursor. In the first comparison, RiPPMiner had an accuracy of 56% (PPV 0.57), while NLPPrecursor had an accuracy of 96% (PPV 0.99) in distinguishing RiPP precursor peptides from non-precursor peptide ORFs (Fig. S6A).

Looking specifically at cases that each classifier correctly identified as RiPPs (n=769 NLPPrecursor; n=500 RiPPMiner) we sought to compare RiPPMiner to NLPPrecursor according to its classification across each RiPP subclass within RiPPMiner's scope (Fig. S6B). Here, RiPPMiner had an accuracy of 84.6% compared to NLPPrecursor's accuracy of 97.9% (MCC 0.819 and 0.975 respectively).

RiPPMiner predicts cleavage sites for lantipeptides, cyanobactins and lasso peptides. For these classes, we chose to compare NLPPrecursor to RiPPMiner using a reserved testing dataset NLPPrecursor was not trained upon resulting in 408 examples. Among these, RiPPMiner reported a cleavage prediction for 221 (54%) of cases. Although NLPPrecursor is able to cleave all of these cases, only these 54% were considered here to facilitate a fair comparison. Overall, both RiPPMiner and NLPPrecursor produce accurate results, with a median and mode prediction distance from N-terminal cleavage of 0 amino acids (Fig. S6C). However, when comparing the distribution of predictions, RiPPMiner had a standard deviation of 6.577, while NLPPrecursor had a standard deviation of 4.754, indicating that NLPPrecursor's prediction are, on average, closer to the true prediction site.

Controlling for ORF size

Because of the short length of RiPP precursor peptides, RiPPs can be distinguished with reasonably good accuracy from protein-coding ORFs using ORF length as a trivial predictor (Fig. S2, median length of 52 amino acids for RiPP ORFs vs. median length of 185.5 for all protein-coding ORFs). Using a ORF size threshold between 20 and 200, we can easily achieve an overall accuracy of 87.5%, but a poor specificity of 0.50 (Fig. S5). By accounting for this effect in our selection of negative training ORFs, we ensure that a potential source of inherent bias in the underlying dataset is removed. Thus, we can be confident that NLPPrecursor has increased our predictive accuracy in identifying RiPP precursor peptide ORFs without mistaking this for simply the accuracy of selecting smaller ORFs, which is especially important considering the ratio of non-RiPP small ORFs is highly unbalanced within a whole genome context.

Supplementary methods

Development of a deep learning model to identify precursor peptides

To create a precursor detection engine that was entirely reliant on sequence data, instead of genomic context, a sequence-based deep learning model was used. We adapted this architecture from the natural language classification model ULMFiT [3] for its reported **state-of-the-art performance on sequence classification tasks in low-data situations, which is achieved through a transfer learning process that leverages large unlabeled datasets from the domain of interest.** Briefly, ULMFiT first performs unsupervised learning through language modeling, and then subsequently transfers the encodings learned from the language modeling objective to a classification task. Here, protein sequences of open reading frames are used as input, whereas the output of the model consists of a classification of each ORF as either a precursor peptide (further subclassified according to RiPP family), or a non-precursor peptide. A total of 14 classes are identified (*n_class*). Protein sequences were tokenized such that each amino acid was considered a unique token, a total of 22 tokens were used (20 amino acids, an ambiguous amino acid token for when gene translations are not definitive, and a padding token), which is subsequently referred to as the vocabulary size.

Each sequence is broken into stretches of roughly 70 in length (exactly 70, 95% of the time, otherwise the length sampled from a random normal distribution with a mean of 70 and a standard deviation of 5), this length is referred to as *bptt*.

The core structure of this model is a recurrent neural network (RNN) encoder, where an embedding layer first converts each token to a 400-length vector (*emb_sz* = 400). This vector is subsequently fed into a three-layer stacked long short-term memory (LSTM) network, where the first two layers have a hidden state of size 1140, and the final LSTM layer has a hidden state equivalent to the embedding size of 400. Since ORF sequences are passed in chunks of size *bptt* through these layers, the final hidden state for each chunk is passed onto to initialize the first hidden state of the next successive chunk.

To train the model in an unsupervised fashion, a language model is first created where the goal of this model is to predict the amino acid, or token, at position *bptt* + 1. Thus, this data is passed into the RNN encoder as described above, as a vector of length *bptt*. This is then converted to a matrix of shape (*bptt*, *emb_sz*) through the embedding layer. After passed through the stacked LSTMs, the final hidden state (vector of size *emb_sz*) is carried forward towards a single layer linear decoder with an output of vocabulary size. This output is compared to the token at *bptt* + 1 using binary cross entropy as a loss function. The weights of the single layer linear decoder are tied to the encoder weights to ensure knowledge is captured within the encoder, and not simply the decoder. During training, a number of dropout layers were defined within this model to prevent overfitting: specifically, an embedding dropout with $p = 0.02$, and a first dropout of $p = 0.25$ along the *bptt* axis. For each of the stacked RNNs, two dropout procedures were applied, first along the hidden states at $p = 0.15$, and for the LSTM weights at $p = 0.2$. A dropout of $p = 0.2$ was applied to the final output of the stacked RNNs, prior to decoding.

The output layer of the trained encoder was then used as input to a classifier to predict

whether a given ORF sequence was a RiPP precursor or not. Similar to the language model, the sequence is chunked into token vectors of length b_{ptt} , and subsequently passed through the RNN encoder, keeping the hidden states at each position across these chunks. This is concatenated to result in a matrix of size $(emb_sz, sequence_length)$. This was further reduced to three vectors conveying (i) the final hidden state of the RNN encoder, the (ii) maximum, and (iii) mean values across the $sequence_length$ dimension resulting in a concatenated vector of size $3 \times emb_sz$. This vector is subsequently sent towards a neural network of two layers. The first layer has ReLU activation, and has an output size of 50. The second layer has a linear activation, with an input size of 50, and output size of n_class . These two layers employ batch normalization, with the first of these applying a dropout of $p = 0.4$ and the second $p = 0.1$. This output is compared to the true label using a weighted binary cross entropy to generate a loss value for backpropagation.

The data used to train the language model was the entirety of unique precursor peptides identified previously by RiPP-PRISM [4], resulting in a total of 3,013 unique ORF sequences. For non-precursor peptides, a selection of diverse microbial genomes was gathered and determined by RiPP-PRISM to not encode RiPP BGCs. ORF sequences were identified from these genomes, and were sampled according to the distribution of ORF sizes seen within RiPP precursors, to obtain a total of 3,013 unique non-RiPP ORF sequences (Fig. S5). The total data available for training was 6,026 ORF sequences (available at: https://github.com/magarveylab/NLPPrecursor/tree/master/training_data/classification). The language model was trained on 90% of this data (excluding class labels), and was validated against the remaining 10%. The model was trained for a total of fifty epochs, using a cyclic learning rate between 10^{-4} and 10^{-2} as described [3]. For the classification model, this same dataset was used, but split into training, validation and testing sets (81%, 9%, and 10%, respectively). The classifier was trained using gradual unfreezing, to ensure the weights learned from the encoder were not drastically forgotten. Using the same cyclic learning rate described above, the classifier was trained for 50 epochs in each of four stages. The first three stages involved stepwise unfreezing of a layer from the bottom, while the final stage unfroze the entire model's weights for training. A batch size of 64 was used throughout training. The Adam optimizer was used, with betas of 0.9 and 0.999 [5]. All hyperparameters here were selected as default as those used within ULMFiT to best capture the transfer learning properties and high performance observed in NLP tasks using this architecture. A visualization of training and validation loss over the course of NLPPrecursor's training regime is visualized in Fig. S14A-B. A step-by-step tutorial to reproduce our results is found online (<https://github.com/magarveylab/nlpprecursor/>) to aid reproducibility.

Development of a deep learning model to predict precursor cleavage sites

From the sequences identified and predicted to be precursor RiPPs, a second model was trained to predict cleavage site(s) within the precursor peptide. This can be framed as an annotation problem, where each amino acid within the ORF sequence is labelled as either part of the final peptide, or not. In natural language processing, several models have been developed for a similar task, labelling parts of speech within a sentence. In particular, this model takes in a tokenized ORF sequence, and for each position within the sequence, predicts a label

such that the input and output sequences are of the same length.

Protein sequences were tokenized such that a start and end tag were added, alongside with the twenty proteinogenic amino acids, and an ambiguous amino acid, and a padding token, resulting in a total vocabulary size of 24. For the output, the following annotations were made: a *start* and *end* tag (matching the input tokens), *before* (annotates each amino acid token upstream of cleavage site), *propeptide* (annotates each amino acid of the mature peptide), *after* (annotates each amino acid token downstream of propeptide sequence) and a padding index, resulting in a total vocabulary size of 6 for the output. The following transitions were allowed within the model: (*start-before*), (*before-before*), (*before-propeptide*), (*propeptide-propeptide*), (*propeptide-after*), (*propeptide-stop*), (*stop, pad*), (*after-after*), (*after-stop*), (*pad-pad*). All sequences were padded with pad tokens at the end such that their final length was equivalent to 200.

The model was architected in three main components, starting with embedding, a bidirectional LSTM, and a final linear chain conditional random field (CRF)[6]. Specifically, the embedding layer transformed each token to a vector of length 100, resulting in each input having a shape of (200, 100) or (*sequence_length*, *emb_sz*). A dropout layer was applied here with $p = 0.1$. This data was subsequently fed through a single layer bidirectional LSTM with a hidden state of size 100 times 2, representing the bidirectionality of the model. All hidden states are carried forward to generate a matrix of shape (*sequence_length*, *hidden_size* × 2) equivalent to (200, 200). This is passed through a single linear layer with an output size of (200, 6), representing the predictions for each annotation at each sequence position. This data is then passed through a CRF layer. Using the forward-backward algorithm, the CRF is able to generate a negative log likelihood of the given predictions matching the true annotations. This value is used as the loss to train the network. During prediction, the Viterbi algorithm is used to determine the most probable path of annotations through the sequence length. The data used to train this model was the entirety of unique precursor peptides identified previously by RiPP-PRISM [4], resulting in a total of 3,013 unique ORF sequences (released here: https://github.com/magarveylab/NLPPrecursor/tree/master/training_data/annotation). This model was trained on 90% of this data and tested against the remaining 10%. A total of 50 epochs were trained (Fig. S14C), using a batch size of 64, and the Adam optimizer with betas of 0.9 and 0.999 [5]. A step-by-step tutorial to recreate these results is found online (<https://github.com/magarveylab/nlpprecursor/>) to aid reproducibility.

Construction of GRAPE

Previously, we developed a retrobiosynthetic algorithm, GRAPE [7], which decomposes the chemical structures of nonribosomal peptide or polyketide natural products, in SMILES format, into their corresponding amino acids together with a list of associated biosynthetic tailoring reactions. However, the published version of GRAPE was limited to polyketides and nonribosomal peptides. To construct BARLEY, we first extended GRAPE with 60 RiPP-specific tailoring reactions to process RiPP scaffolds (Dataset S6). Specifically, GRAPE retrobiosynthesizes RiPPs according to the same logic as previously described [7] in four major steps. First, macrocycle forming bridges are broken. In this step, we have updated GRAPE to include RiPP-specific crosslinks such as, but not limited to, the lanthionine linkage, duramycin

linkage and thiopeptide pyridine formation. Second, reactions forming heterocyclic rings and amino acid modifications are retrobiosynthesized. Examples include oxazol(in)es, thiazol(in)es and dehydrated amino acids. In the third steps, amide and thioamide bonds are cleaved to reveal the primary amino acid structure of RiPPs. Finally, additional groups are cleaved such as sugars and indoles. Through these steps, GRAPE is able to identify the specific tailoring reactions that derivatize the precursor peptide identified as well as its resulting primary amino acid structure. In certain cases, a definitive amino acid cannot be derived from a chemical structure, such as the case of lanthionine linkages, where it cannot be determined which amino acid represented serine or cysteine without corresponding genomic information. In these cases, an ambiguous amino acid (denoted SerCys in Fig. S13) is generated that can be matched to either serine or cysteine in subsequent alignments.

BARLEY chemical distance

A chemical distance metric describes the theoretical distance or similarity between two chemical structures; a widely used example is the Tanimoto coefficient, as applied to a pair of binary chemical fingerprints. BARLEY calculates a chemical distance tailored to the comparison of RiPP structures by using GRAPE to deconstruct chemical scaffolds into their corresponding amino acids and tailoring reactions. For a comparison between a query chemical scaffold to a subject, a Smith-Waterman alignment is calculated using an identity matrix, scoring +1 for exact matches, and a gap opening and extension penalty of -2. From this alignment, two scores are denoted for the local alignment, the total number of amino acids in the query that were exactly matched and mismatched to the subject, which are weighted +1 and -1, respectively. From the identified tailoring reactions, three metrics are derived with respect to the query: the number of tailoring reactions observed in both the query and the subject, the number of tailoring reactions observed in either the query or the subject but not in both, and the number of tailoring reactions between query and subject that were marked as similar. The following tailoring reactions were considered to be similar: lanthionine and methyl-lanthionine, labionin and methyl-labionin, oxazole and oxazoline, thiazole and thiazolines, phenyloxazole and phenyloxazoline, dehydroalanine and dehydrobutyric acid. These three scores (tailoring reaction match, tailoring reaction mismatch and similar tailoring reactions) are weighted +5, -5, and +5, respectively. The sum of these five weighted scores are used to determine a total score. To generate a relative score between zero and one, a self-score is generated between the query and itself. The total score is divided by the self-score to determine the relative similarity of two RiPP chemical scaffolds. Thus, the relative score represents how close the subject molecule is to the query, in a range between zero and one, where one represents the exact same molecule. This relative score is asymmetric as it is normalized according to the query molecule.

BARLEY genomic distance

Akin to chemical distances, the genomic distances by BARLEY aim to describe the similarity of RiPPs in a scale between zero and one. However, genomic distances are generated by solely examining a pair of RiPP biosynthetic gene clusters, without using any structural information. BARLEY uses the genes identified by RiPP-PRISM to build a model of the cleaved precursor peptide core amino acid sequence, and the total tailoring reactions possibly encoded. RiPP-PRISM can identify up to 112 genes that encode distinct tailoring reactions for RiPPs [4]. For a

comparison between a query gene cluster to a subject, all potential tailoring reactions are gathered for both query and subject (as described below). The corresponding core amino acid sequence for each BGC are then identified using the predictions by RiPP-PRISM. In cases where multiple precursor peptides are identified within a RiPP gene cluster, each is considered as a unique entity, and all tailoring reactions are generated independently. The query and subject are then scored in the same manner as described above for chemical distance.

Integration of known RiPP tailoring reactions within BARLEY

Genomically encoded tailoring reactions act upon a precursor peptide in a stepwise manner (using the products of prior reactions as substrate for subsequent reactions) and in an uncertain manner (able to act upon multiple substrates) to generate a wide array of possible tailoring reactions, not all of which are truly detected in nature. To account for this, BARLEY incorporates a unique algorithm to guarantee that all possible tailoring reactions are predicted for each RiPP. Each reaction is encoded into BARLEY with three main parameters: (i) a list of genes required to perform the reaction, (ii) a list of precursor chemical moieties and (iii) the resulting modification. Using the available precursors, all possible reactions are iteratively applied in a fixed order, until the number of possible modifications converges upon a maximum. Precursors for a tailoring reaction include both amino acids and other modifications as a result of tailoring reactions, and are not consumed during the execution of a reaction, thus allowing for all combinatorial possibilities.

An example is the formation of lanthionine linkages by LanB and LanC. A dehydration reaction can be encoded as (i) requiring LanB, (ii) requiring 1 of either serine or threonine and (iii) resulting in either dehydrated serine (Dha) or dehydrated threonine (Dhab). A lanthionine linkage reaction is then encoded as (i) requiring LanC, (ii) requiring 1 free cysteine and 1 of either Dha or Dhab and (iii) resulting in a lanthionine or methyl-lanthionine linkage. For a given gene cluster containing LanB and LanC, with a precursor peptide that contains Ser, Thr, and Cys, applying this algorithm results in a total of four modifications: Dha, Dhab, lanthionine and methyl-lanthionine. While not all of these may truly be present in the final product, this ensures that all possibilities of tailoring reactions are considered.

BARLEY structural novelty index

To assess the novelty of a genomically encoded RiPP, it is necessary to quantify its similarity to the complete set of previously isolated and characterized RiPPs, including both RiPPs with known clusters as well as those whose structures alone are known. As described above, both genomically and chemically encoded RiPPs can be deconstructed into their core amino acid sequence and tailoring reactions, and compared according to five scoring metrics (three describing core amino acid alignments, and two representing tailoring reactions match and mismatch from both query and subject perspectives). Without weighting, these five metrics are used as features to describe a comparison, which is subsequently trained using a random forest regression model.

The random forest is structured to input these five parameters, and output a continuous value ranging between -1 and $+1$, where -1 represents two unrelated products, 0 represents members of the same RiPP family, and $+1$ represents an exact match.

The dataset of 138 RiPP gene clusters, matched with 161 chemical scaffolds was used to

train and validate the performance of this model. First, 75% of these gene clusters, and their corresponding chemical scaffolds, were split into a training set, while the remaining 25% was used to test performance. Within the training set, a 10-fold cross validation was employed to optimize hyperparameters within the random forest, including the number of trees (Fig. S15), the maximum number of randomly sampled features (Fig. S16), and the minimum leaf size (Fig. S17). Cross-validation indicated that the optimum number of trees was 400, the number of randomly sampled features was 5, and the minimum node size to be one. The relative importance of each feature, as assessed by the mean decrease in impurity, is visualized in Fig. S18.

Metabolomic mass spectral analysis

To generate a profile of MS1 ions per strain, all detected MS1 ions above a baseline intensity of 1,000 from each experimental analysis across multiple media conditions were compared. Ions within 5 PPM and a 30 second retention time window were considered the same metabolite. Of these overlapping ions across media conditions, a single candidate ion was chosen based on its relative intensity as a representative for subsequent analysis. All MS1 ions in this representative set were then compared to a dataset of 118 blank media extractions to remove any compounds not associated to bacterial metabolism using the same PPM and retention time tolerance described above. The remaining MS1 ions were then compared to all analytical experiments from 463 strains with associated genomic data. BARLEY was used to determine strains with identical encoded RiPPs, any MS1 ions from the candidate strain that were overlapped with non-RiPP carrying strains were eliminated using the same PPM and retention time tolerances described above. The remaining MS1 ions were analysed according to our RiPP structure prediction and fragmentation module explained below.

RiPP structure prediction and peak matching

The structure prediction engine of RiPP-PRISM was extracted to create a separate computational interface accepting cleaved precursor peptide sequences and a list of their corresponding RiPP modification enzymes (without any limit on the length of tailoring reactions), along with a list of structure prediction settings to help manage computational time. By default, in all experiments here and the DeepRiPP web platform, we set a maximum scaffold limit (per cleavage site) of 50, a maximum run time of 10 minutes with a maximum of 500 plans per reaction, and a maximum of 1000 combinatorial plans. With these settings, we were able to balance overall computational time, while still providing enough sampling time across the breadth of combinatorial possibilities. As described previously [8,9], a library of fragment masses were generated *in silico* from a randomly chosen set of 100 predicted scaffolds. The top 1,000 most frequently observed fragment masses were saved. At the end of this process, each RiPP is annotated with a library of predicted scaffolds at each potential cleavage site \pm 5 AAs from NLPPrecursor's predicted cleavage site, alongside their *in silico* fragmentation masses. While these settings were evaluated for accuracy and reasonable computational efficiency (Fig. S3), the platform itself is available to generate any arbitrarily large structure prediction library given appropriate computational time and infrastructure.

All MS1 ions, after passing several filters described above, were then evaluated for their

similarity to each RiPP. For each MS1 ion, a true mass is calculated according to their charge, and their potential to exist as one of six adduct types ($[M+H]^+$, $[M+Na]^+$, $[M+K]^+$, $[M+2H]^{2+}$, $[M+H+Na]^{2+}$, $[M+3H]^{3+}$), subsetted according to peak charge. An MS1 match is considered when any of these adduct adjusted masses match the predicted mass for a RiPP with a tolerance of 20 PPM. For MS2 match scores, the top 1,000 most frequently observed fragment masses were then compared to the MS2 spectra for each candidate peak using a 20 PPM tolerance, where a score was generated representing the fraction of MS2 ions that were matched to an *in silico* prediction.

DeepRiPP web application

The DeepRiPP web application is implemented as two major projects, representing the front-end and back-end respectively. The front end application is written entirely in Javascript using the Vue.js framework and the Material Design system (as implemented in Vuetify). This front end application communicates via HTTP requests to the back-end server to manage logins, job submissions, tracking and visualizations. The HighCharts library is used specifically to visualize CLAMS results. The back-end of this application is built using Python 3.7 to provide three major functionalities: the integration of scientific applications presented in this work, a RESTful web server and finally a database and job management system. Specifically, each DeepRiPP module (NLPPrecursor, BARLEY and CLAMS) is integrated in a modular manner so that jobs can interact with these either individually or in the context of a combined genomic and metabolomic analysis (the entire DeepRiPP workflow). NLPPrecursor is installed as a Python package (publicly available, along with all data used to train the model, from <https://github.com/magarveylab/nlpprecursor>) and directly accessed within the Python environment. However, BARLEY (Java v8) and CLAMS (R v3.6.0: publicly available at <https://github.com/magarveylab/clams-release>) are installed globally on the webserver and accessed via Python's subprocess module. The backend uses SQLAlchemy, Flask and Redis not only to maintain a registration and login system, but also to manage job submission and leverage a multitude of workers available asynchronously. All passwords are hashed using sha256 with a salt length of eight and stored within an SQLite database (Fig. S19). Support for the entire DeepRiPP web application is provided at <https://github.com/magarveylab/NLPPrecursor/issues>. A collection of screenshots are provided here guiding users through sample data (Figs. S20-32) with all data corresponding to these samples provided in the homepage of NLPPrecursor for users to explore within the context of the web application.

Comparisons to existing software

The genomic modules of BARLEY were compared to the structure prediction engine of RiPP-PRISM [4] and the cluster comparison software BiG-SCAPE [10]. The goal of genomic comparisons is to provide a proxy for chemical similarity, and in this case, we compare the genomic scores generated by BARLEY, the similarity of predicted chemistry from RiPP-PRISM and BiG-SCAPE BGC similarity scores to the chemical distances produced by BARLEY for the set of known biosynthetic clusters in Dataset S1. Specifically, Spearman's rank correlation coefficient is used to evaluate the rank-wise similarity between genomic and chemical distance measurements (Fig. S9B). To evaluate whether BARLEY (genomic) correlation coefficients

were significantly higher than RiPP-PRISM and BiG-SCAPE and further test whether this difference could have appeared spuriously due to random chance within the dataset, we tested the null hypothesis that the correlation coefficients are identical using the Fisher z transformation (Fig S9C) [11,12]:

$$z = F(\rho) = \operatorname{arctanh}(\rho) = \frac{1}{2} \ln \left(\frac{1+\rho}{1-\rho} \right)$$

The standard deviation can be calculated for Spearman's ρ as follows [13]:

$$SD(\rho_1 - \rho_2) = \sqrt{\frac{2}{N-3}}$$

where N represents the sample size. With this, using the normal distribution, we can perform a one tailed test using this formula:

$$P(\rho_1 > \rho_2) = 1 - CDF_{NORM} \left(\mu = \left| F(\rho_1) - F(\rho_2) \right|, \sigma = SD(\rho_1 - \rho_2) \right)$$

General Experimental Procedures

For analytical separation and to record high resolution LC-MS/MS spectra, a SciEX 5600+ TripleTOF mass spectrometer (ABSciEX) with an electrospray ionization (ESI) source was used. The system operates using CID with helium for fragmentation, coupled with an Agilent 1100 series HPLC system using an luna C18 column (150 mm × 2.1 mm, Phenomenex). For preparative separation we used Dionex UltiMate 3000 HPLC system, coupled with a Luna C18 column (250 mm × 15 mm, Phenomenex). For both analytical and preparative separation, the mobile phase consists of gradient mixture of double distilled H₂O with 0.1% formic acid and acetonitrile. 0.1% formic acid was used as buffer for both solvents.

To record nuclear magnetic resonance (NMR) spectra of deepstreptin we used Bruker AVIII 700 MHz. deepstreptin was dissolved in methanol-d₃ (Sigma-Aldrich). Recorded spectra included, 1D (¹H and DEPTq), 2D (¹H-¹H) COSY, TOCSY, ROESY, and NOESY, and 2D (¹H-¹³C) HSQC, and HMBC.

Microbial Strains and Culturing

To generate the microbial extract database, 463 microbes have been cultured on 26 different media. The majority of the fermentations have been conducted in shaking liquid cultures grown for a period of 2-5 days until reaching the optimum OD₆₀₀. Detailed media and extraction conditions for each strain are shown in Dataset S7.

Streptomyces sp. BTA 0171 was maintained on ISP3 agar, or KE or GGYM broth with shaking at 200 RPM, at 28 °C. ISP3 medium consists of 4 g/L yeast extract, 10 g/L malt extract, 4 g/L dextrose, and 15 g/L agar. KE medium consists of 1 g/L glucose, 10 g/L potato dextrin, 5 g/L NZ-amine, 5 g/L yeast extract, 3 g/L beef extract, 0.5 g/L CaCO₃, 0.05 g/L MgSO₄·7H₂O, 2 mL/L filter-sterilized phosphate buffer, added after autoclaving (consists of 91 g/L potassium phosphate monobasic and 95 g/L potassium phosphate dibasic at pH 7). GGYM medium

consists of 4 g/L glucose, 4 g/L yeast extract, 10 g/L malt extract, and 5 g/L glycine. *Flavobacterium ginsengiterrae* JCM 17337 was obtained from the Japan Collection of Microorganisms and maintained on nutrient agar or Caitones –Yeast (CY) broth. CY medium consists of 5 g/L casitones and 1 g/L yeast extract. For antimicrobial assay, *P. aeruginosa* PAO1, *S. aureus* Newman, and *C. albicans* ATCC 90028 were maintained and cultured on cation-adjusted Mueller Hinton, CAMH (4 g/L, Difco™, USA), tryptic soy broth, TSB (24 g/L, Difco™, USA), and potato dextrose broth, PDB (30g/L, Sigma-Aldrich, USA), respectively.

Production and purification of deepstreptin

Streptomyces sp. BTA 0171 was cultured in small 250 mL Erlenmeyer flasks to produce seed culture. A 48 h single colony of strain BTA 0171 grown on ISP3 agar was used to inoculate 50 ml GGYM medium. The seed cultures were incubated for 48 h at 28 °C with continuous shaking at 200 rpm. Thereafter, 10 ml of the seed cultures were aseptically transferred to 2.8-L Fernbach flask containing one liter KE broth and incubated for 5 days at 28 °C with continuous shaking at 200 RPM. A total of 18 liters KE media was used for large scale production of deepstreptin. Cells were harvested by centrifugation at 4000 rpm for 20 min at 4 °C then pellets were extracted with one liter of methanol for 4 h. Simultaneously, a resin mixture of 1:1 HP20 and XAD7 were added to the supernatant at ratio of 3:100 W/V, shaken at 100 RPM for 3 h at room temperature, filtered under vacuum, washed with water. The resin was then extracted three times with methanol (1:4 W/V) followed by a final wash in acetone (1:4 W/V). The methanol extract from the mycelial pellets and resin was combined and dried under vacuum using a rotary evaporator followed by nitrogen air for complete drying to yield 10 g of crude extract. The dried extract was then suspended in water and subjected to liquid-liquid partition between 1:1 n-butanol and water. The n-butanol fraction was dried under vacuum using rotary evaporator then nitrogen gas to yield 600 mg dry residue. The residue was suspended in 2 ml methanol and applied on the top of size exclusion column (Sephadex LH20, 1.6 × 80 cm), methanol was used as the mobile phase with flow rate of 1 ml/min. Fractions containing deepstreptin were pooled and dried under nitrogen to yield 120 mg dry mass which was dissolved in 5 ml methanol and subjected to a semi-preparative reversed-phase HPLC with a luna 5 µm C18 column (Phenomenex, 250 mm × 15 mm) using water and acetonitrile with 0.1% formic acid as mobile phase employing a linear gradient of 5% to 80% acetonitrile over 30 min followed by a wash of 100% acetonitrile for 10 min. Fractions were collected at 5 sec interval with deepstreptin being eluted at 18 min. HPLC fractions contain deepstreptin were pooled and dried under nitrogen line to yield 10 mg of pure compound which is then dissolved in methanol-d3 and subjected to NMR analysis. Detailed NMR spectra and assignment of chemical shifts are detailed in [Figs. S33-39](#).

Production and purification of deepflavo and deepginsen

Flavobacterium ginsengiterrae JCM 17337 was allowed to grow in 250 mL Erlenmeyer flasks to produce seed culture. A 48 h single colony of strain JCM 17337 grown on nutrient agar was used to inoculate 50 ml nutrient medium. The seed cultures were incubated for 48 h at 28 °C with continuous shaking at 200 rpm. Thereafter, 10 ml of the seed cultures were aseptically transferred to 2.8-L Fernbach flask containing one liter CY broth and incubated for 5 days at 28

°C with continuous shaking at 200 RPM. A total of 18 liters CY media was used for large scale production of deepflavo and deepginsen. A resin mixture of 1:1 HP20 and XAD7 were added to the fermented broth at ratio of 3:100 W/V, shaken at 100 RPM for 3 h at room temperature, filtered under vacuum, washed with water. The resin was then extracted three times with methanol (1:4 W/V) followed by a final wash in acetone (1:4 W/V). The mixture of methanol and acetone was dried under vacuum to yield 7.9 g of crude extract, which was dissolved in water (500 mL) and partitioned with EtOAc (3 × 500 mL) to yield an EtOAc fraction of 4.2 g. The EtOAc fraction (4.2 g) was then subjected to a flash column chromatography (Teledyne) with a 30 g SNAP Ultra C18 column (Biotage) using water and acetonitrile as mobile phase at 35 ml/min employing a linear gradient of 10% to 100 % acetonitrile over 18 mins followed by 5 mins of 100% acetonitrile wash. Fractions containing deepflavo and deepginsen were pooled to result a dry mass of 230 and 160 mg, respectively. Deepflavo containing fraction (230 mg) was then subjected to a semi-preparative reverse phase HPLC with a Luna 5 µm C18 column (Phenomenex, 250 mm × 10 mm) using water and acetonitrile with 0.1% formic acid as mobile phase employing a linear gradient of 20% to 45% acetonitrile over 12.5 min followed by 5 mins isocratic run with 45% acetonitrile then a wash of 100% acetonitrile for 10 min. deepflavo is eluted at 17.6 min. HPLC fractions contain deepflavo were pooled and dried under nitrogen to yield 1.4 mg of pure compound. Deepginsen containing fraction (160 mg) was subjected to reverse phase HPLC using water and acetonitrile with 0.1% formic acid as mobile phase employing a linear gradient of 40% to 60% acetonitrile over 17.5 min followed a wash of 100% acetonitrile for 10 min. Deepginsen was eluted at 19.9 min with a total of 2.1 mg. Both deepflavo and deepginsen were then dissolved in DMSO-d6 and subjected to NMR analysis. Detailed NMR spectra and assignment of chemical shifts for deepflavo and deepginsen are detailed in Figs. S41-55.

Supplementary figures

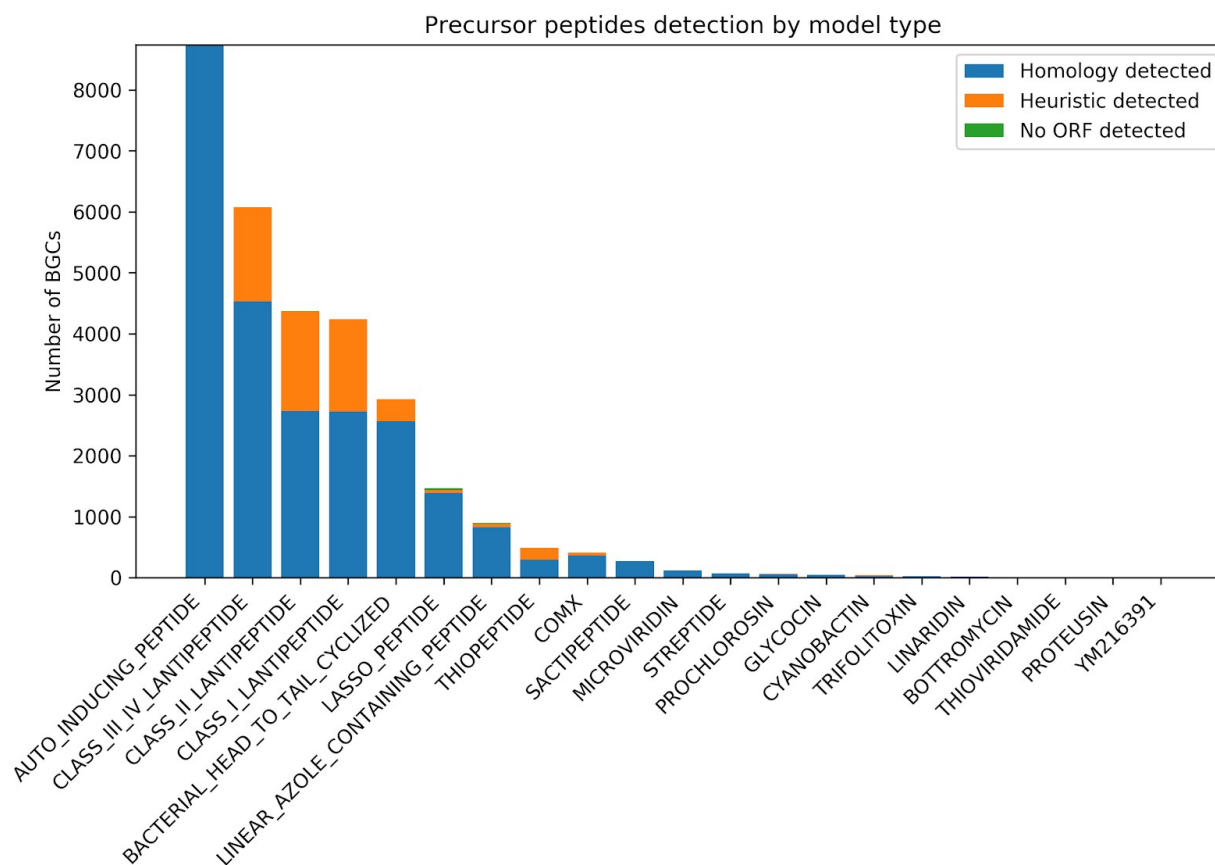


Fig. S1: Limitation of precursor peptide detection across RiPP classes as identified by RiPP-PRISM. Stacked bar chart showing the total number of RiPP biosynthetic gene clusters stratified according to the detection methods of their precursor peptides. Precursor peptides identified by homology (shown in blue) have predicted leader cleavage according to RiPP-PRISM's library of motifs, and can be subsequently used for structure prediction and novelty assessment by BARLEY. Peptides identified by heuristic rules (shown in orange) are typically not cleaved by RiPP-PRISM, and are subsequently not used for downstream structure prediction or novelty assessment. Rarely, some RiPP biosynthetic gene clusters were not found with any predicted precursor peptide ORFs (shown in green).

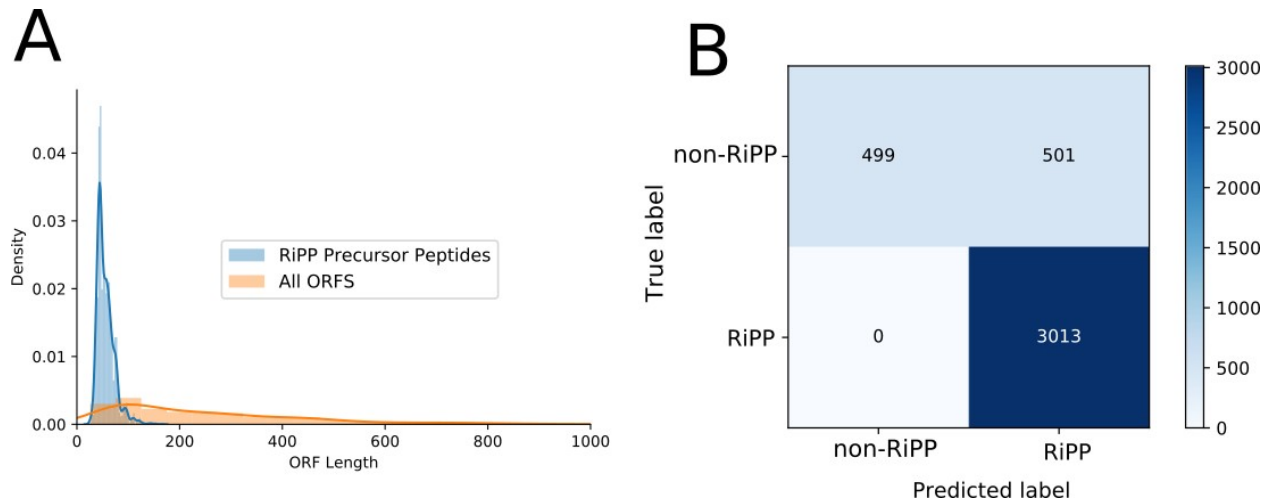


Fig. S2: ORF size is a confounding variable in predicting RiPP precursor peptides. (A)

Density plot comparing the lengths of ORFs, in amino acids, encoding RiPPs or randomly sampled ORFs from complete bacterial genomes. (B) Confusion matrix depicting predictive accuracy of discriminating RiPP precursor peptides solely by ORF sizes, using a minimum and maximum size of 20 and 200 amino acids, respectively. A dataset consisting of 3,013 predicted RiPP ORF sequences and 1,000 randomly sampled ORF sequences from non-RiPP-producing bacterial genomes was used to model the predictive power of ORF length.

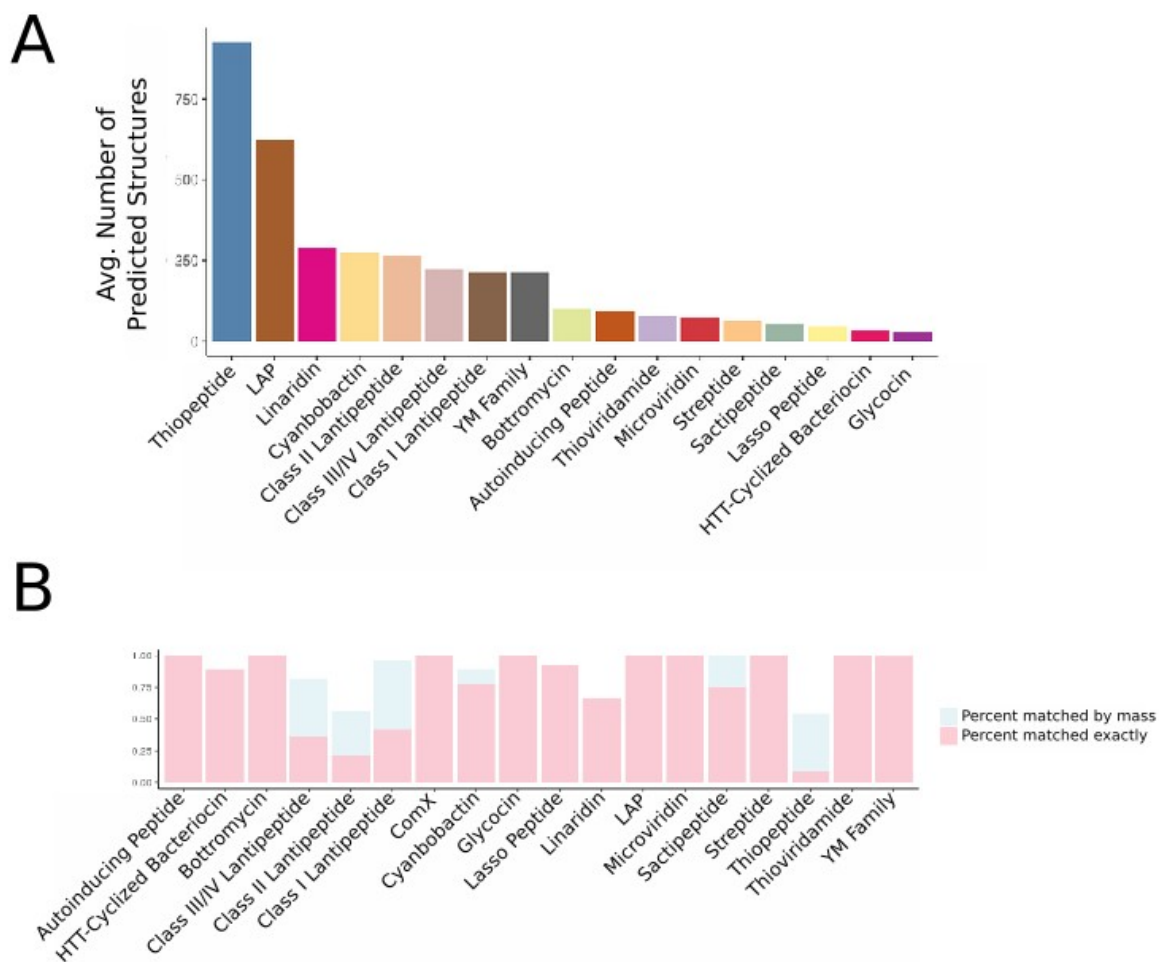


Fig. S3: Structure prediction complexity and accuracy across RiPP families. (A) Using our dataset of known clusters (Dataset S1), we perform structure prediction with an unlimited number of possible predicted structures as output in order to quantify the average number of predicted structures per RiPP family. (B) Using the same dataset, the RiPP structure prediction engine was evaluated based on the proportion of biosynthetic gene clusters for which at least one structure from the predicted library matches either the exact chemical structure of the known product (pink) or its exact mass (blue).

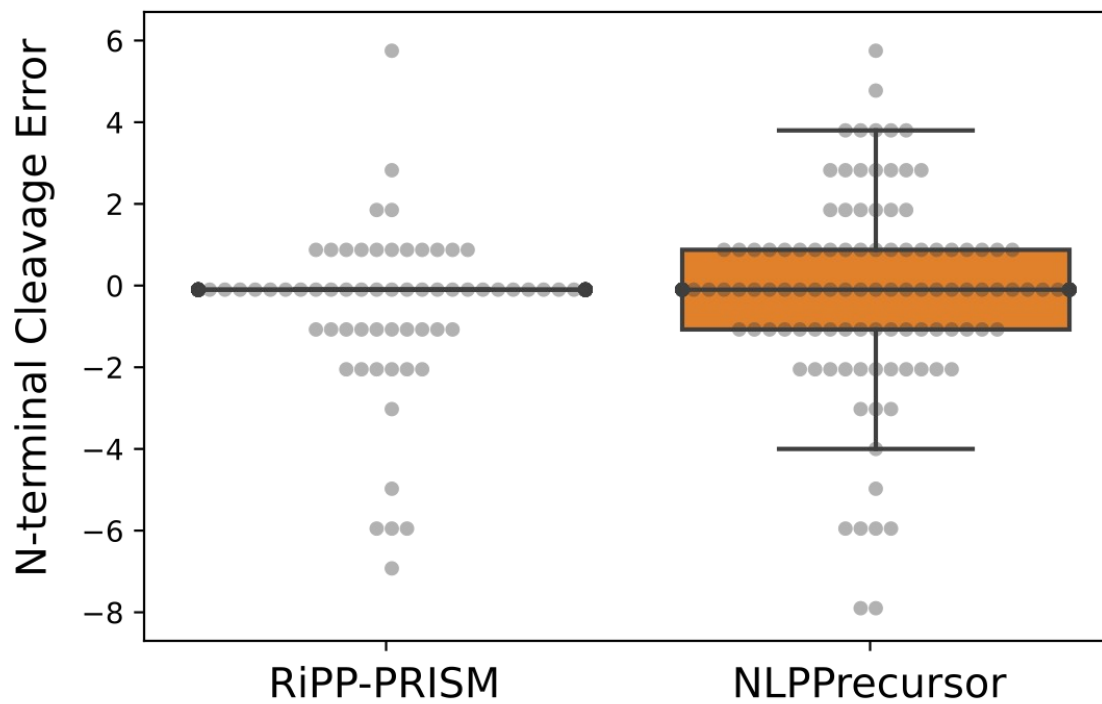
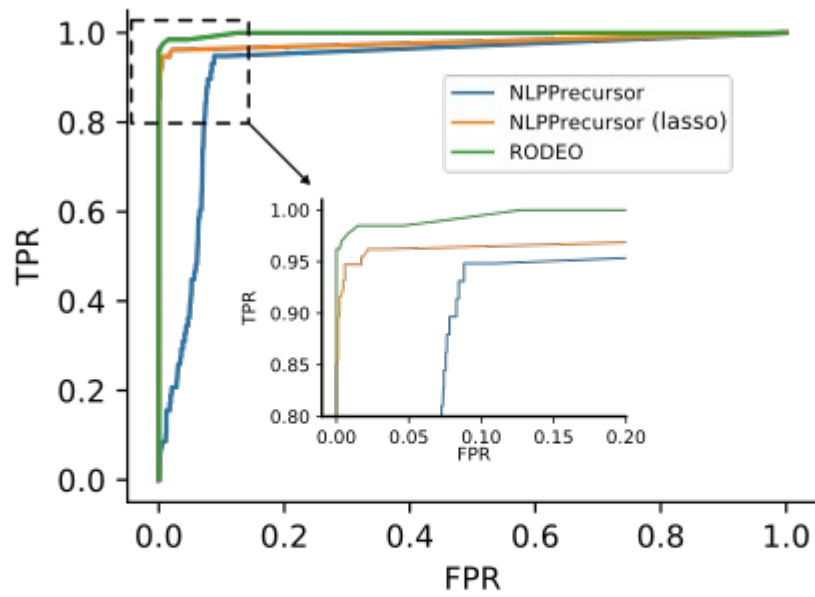


Fig. S4: RiPP-PRISM and NLPPrecursor cleavage prediction accuracy. Boxes represent the interquartile range (IQR) while whiskers represent 1.5 times the IQR.



	RODEO	NLPPrecursor	NLPPrecursor (lasso)
Accuracy	0.99921	0.97951	0.98699
AUC	0.9988	0.9244	0.95718
PPV	0.99921	0.97951	0.98374

Fig. S5: Comparing NLPPrecursor to RODEO for identifying lasso precursor peptides. (A) Receiver operating characteristic curve of NLPPrecursor's prediction probabilities, trained on the DeepRiPP training set (NLPPrecursor) or the dereplicated, lasso peptide-specific training set provided with RODEO (NLPPrecursor-lasso), and RODEO linear combination scores. (B) Classification metrics of RODEO and DeepRiPP (AUC = Area under receiver operating characteristic curve, PPV = Positive predictive value).

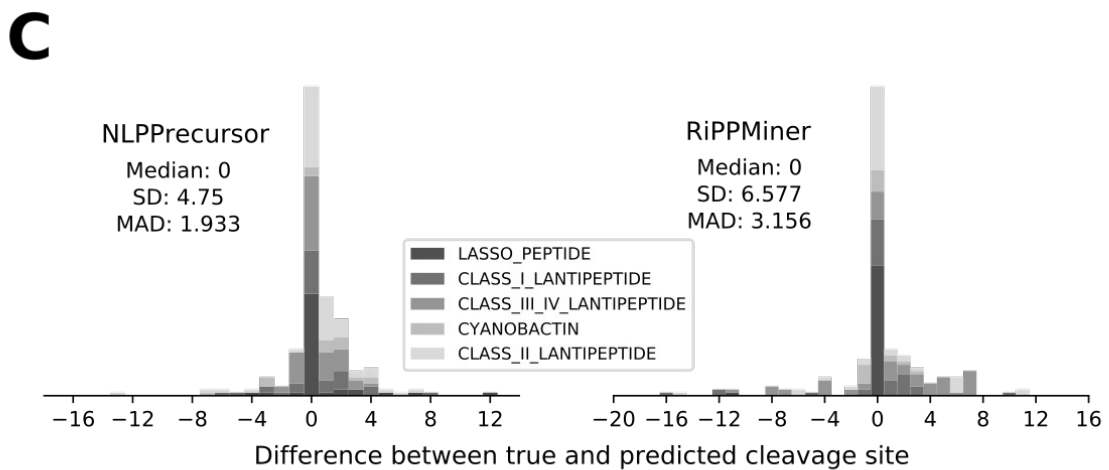
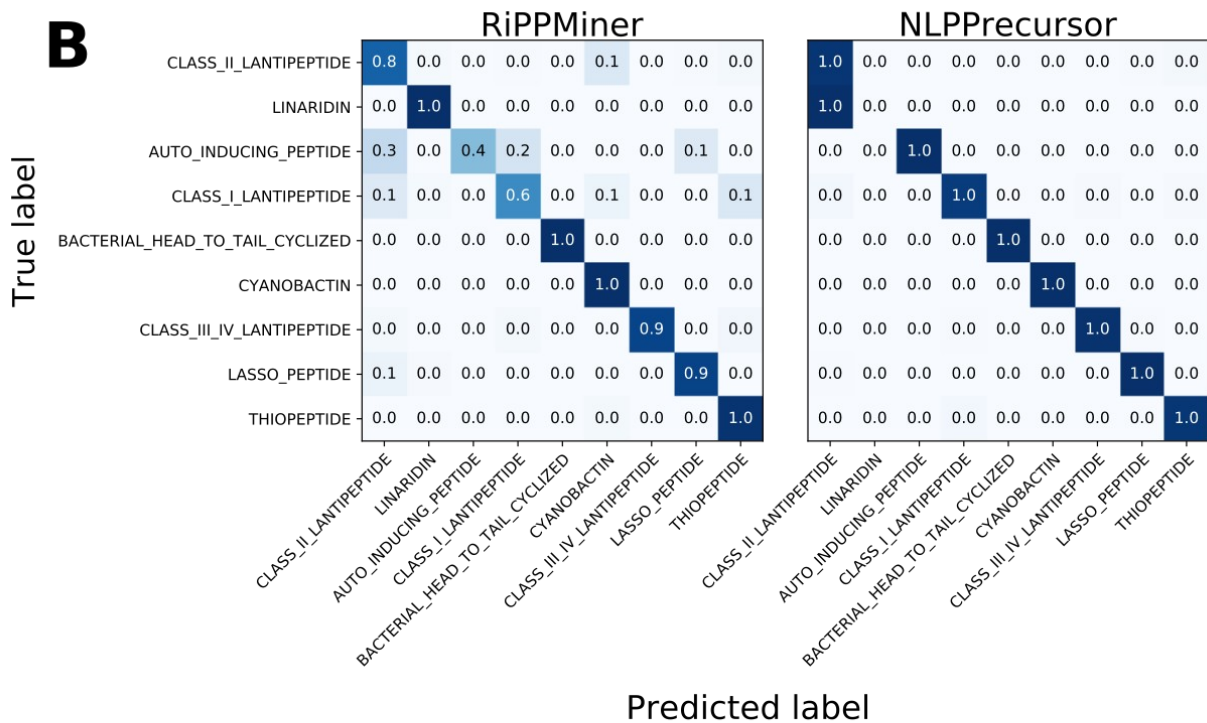
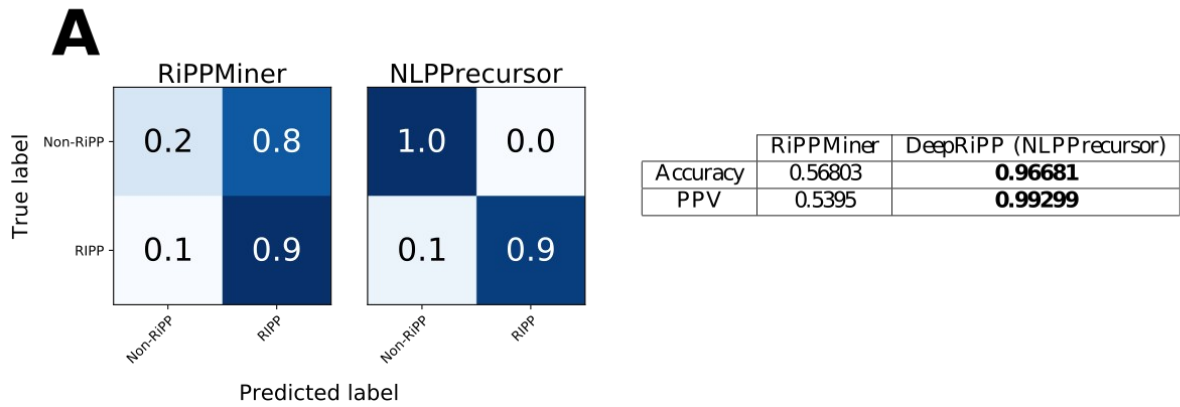


Fig. S6: Comparing RiPPMiner and NLPPrecursor. (A) Confusion matrix and summary statistics comparing NLPPrecursor and RiPPMiner according to their ability to identify RiPP precursor peptides. (B) Confusion matrix comparing NLPPrecursor and RiPPMiner according to their ability to discriminate between RiPP precursor peptide classes (n = 500, RiPPMiner, and n = 769, NLPPrecursor). (C) Comparison of cleavage prediction accuracy between NLPPrecursor and RiPPMiner (MAD, mean absolute deviance).

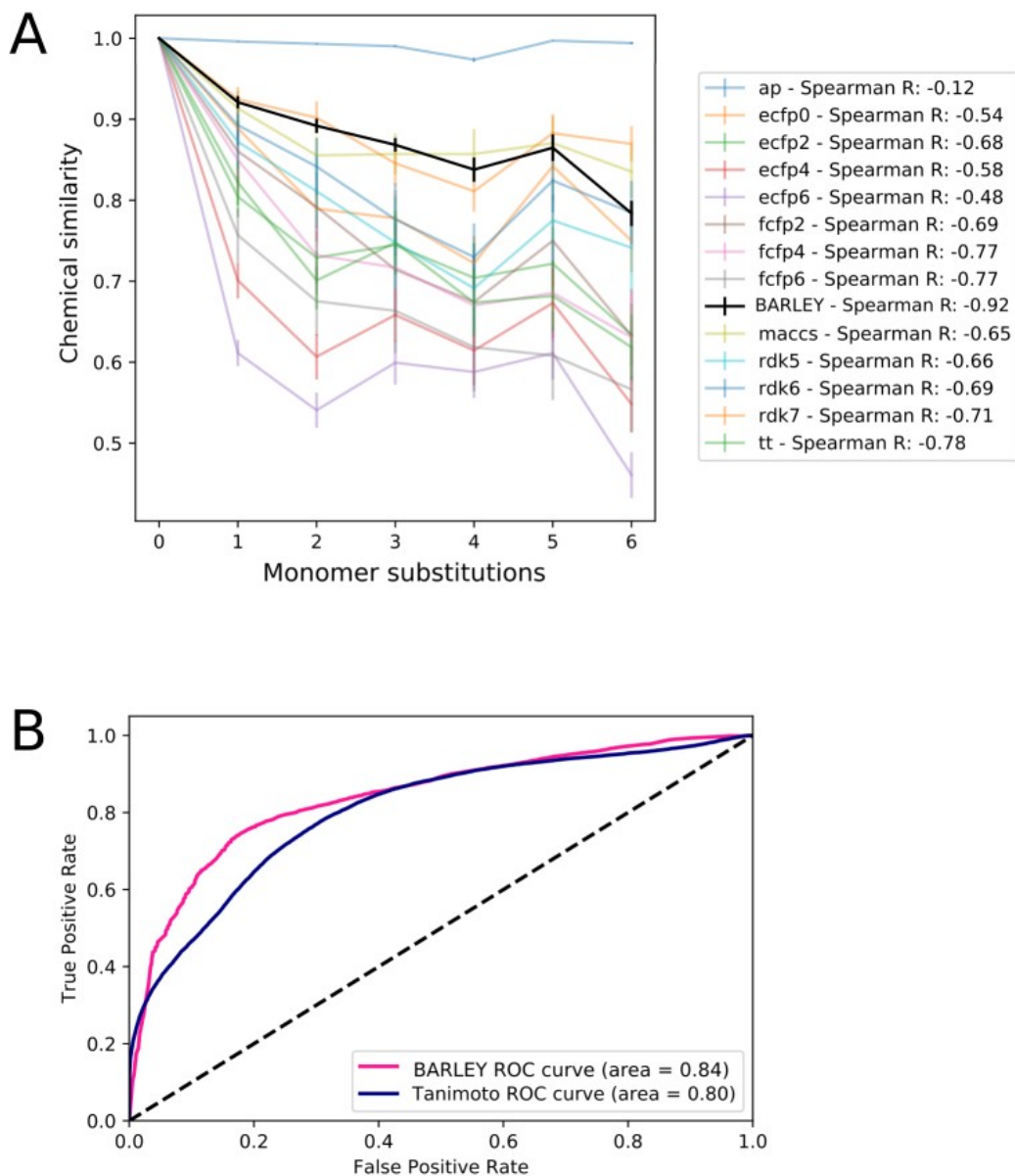


Fig. S7: Comparison of methods for assigning chemical similarity to pairs of RiPPs. (A)

Line chart describing the relationship between increasing chemical divergence (number of monomer substitutions) in an artificially combinatorialized [14] dataset of 600 compounds to chemical similarity scores. BARLEY is highlighted in black, while other metrics are coloured as shown and compared via the Spearman rank correlation coefficient listed here for BARLEY as 0.92 and topological torsion as 0.78, the top scoring metric apart from BARLEY. (ap: atom pair [15], ecfp- n : extended connectivity fingerprint of radius $n/2$ [16], fcfp- n : functional connectivity fingerprint of radius $n/2$ [16], maccs: Molecular ACCESS System structural keys [17], rdk- n : daylight-like fingerprints with path length n [18], tt: topological torsion [19]) (B) Receiver operating characteristic curve comparing BARLEY to the Tanimoto coefficient between ECFP6 fingerprints [16] in classifying chemical scaffold pairs belonging to the same or divergent RiPP families.

The area under the curve (AUC) is used here as a measure of accuracy.

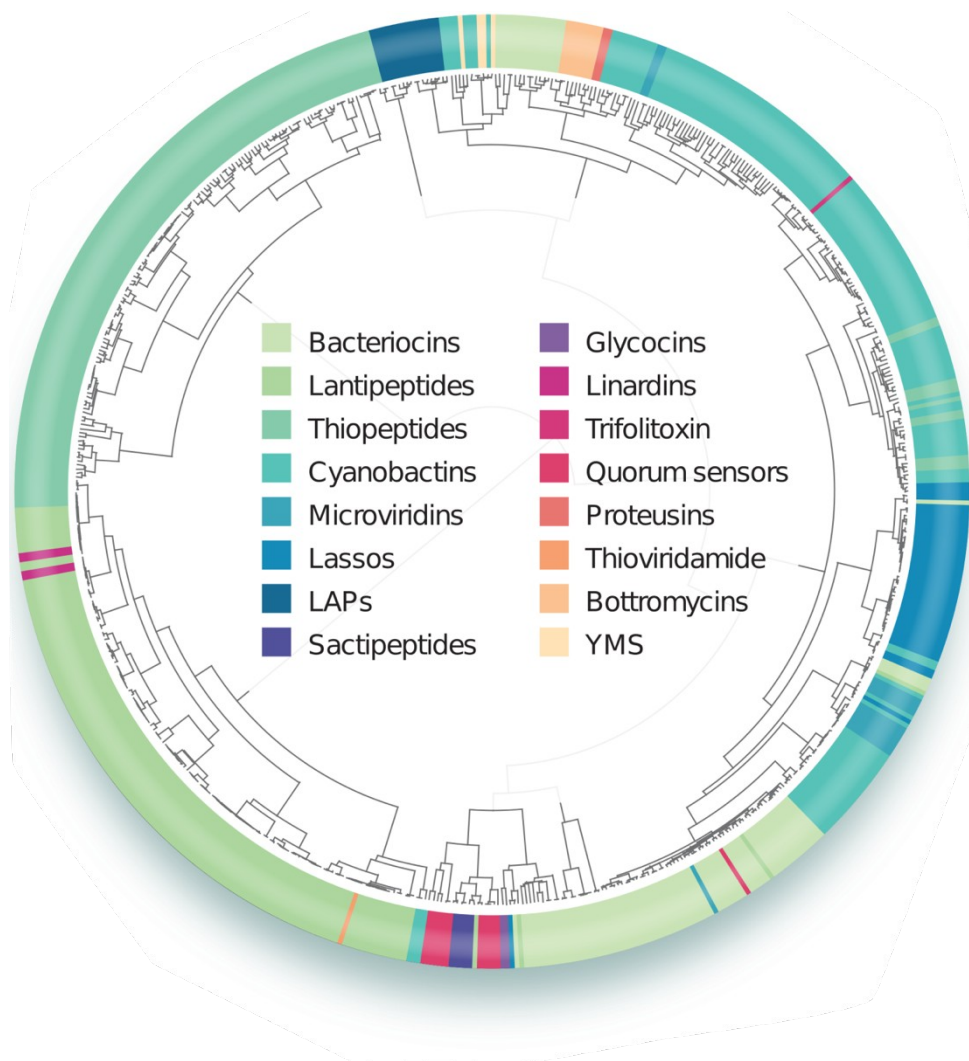
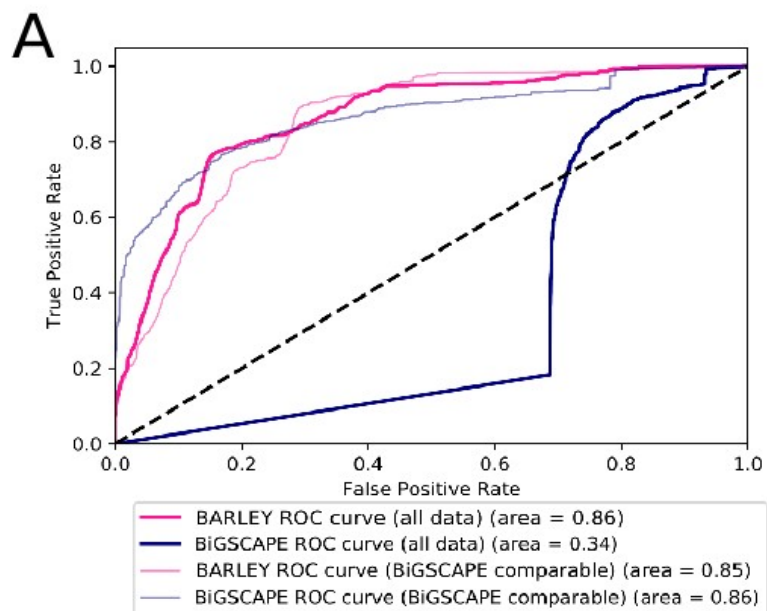
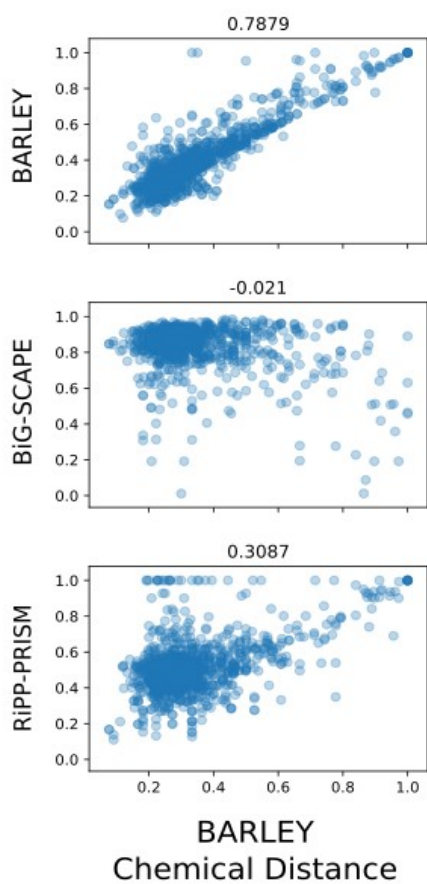


Fig. S8: Grouping of known RiPP chemical scaffolds according to diversity scores assigned by BARLEY. Dendrogram generated through hierarchical clustering using the Ward linkage method on a distance matrix comprising all pairwise scores generated by BARLEY between 638 unique known RiPP scaffolds.



B

Genomic Distances



C

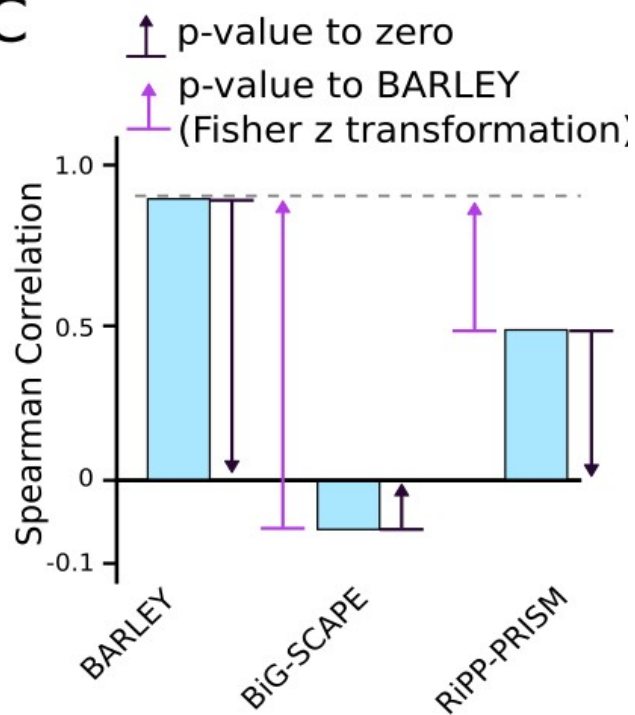


Fig. S9: Evaluating genomic similarities of RiPP genes as predicted by BARLEY.

(A) Receiver operating characteristic curve comparing BARLEY and BiG-SCAPE distance metrics on the task of differentiating RiPPs according to family. Shown in thicker lines are results computed across all gene clusters from Dataset S3, while thinner lines represent only gene cluster comparisons successfully scored by BiG-SCAPE (28% of total data). The area under the curve (AUC) is used here as a measure of accuracy. (B) Scatter plots showing the relationship between BARLEY chemical distances and genomic distances calculated by BARLEY, BIG-SCAPE, and RiPP-PRISM, respectively. The comparison was performed on a dataset of 136 known RiPP clusters which encode 161 small molecules. These genomic distance metrics were compared according to the Spearman correlation coefficient, whose value is shown for each genomic distance, where a greater value represents a more consistent rank correlation. (C) Schematic depiction for the Fisher z transformation used to compare correlation coefficients and test for statistical significance [11,12].

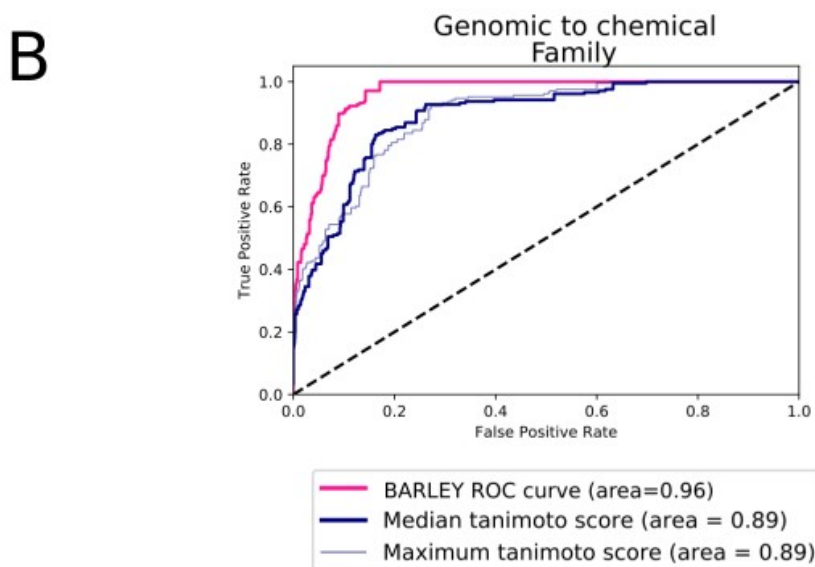
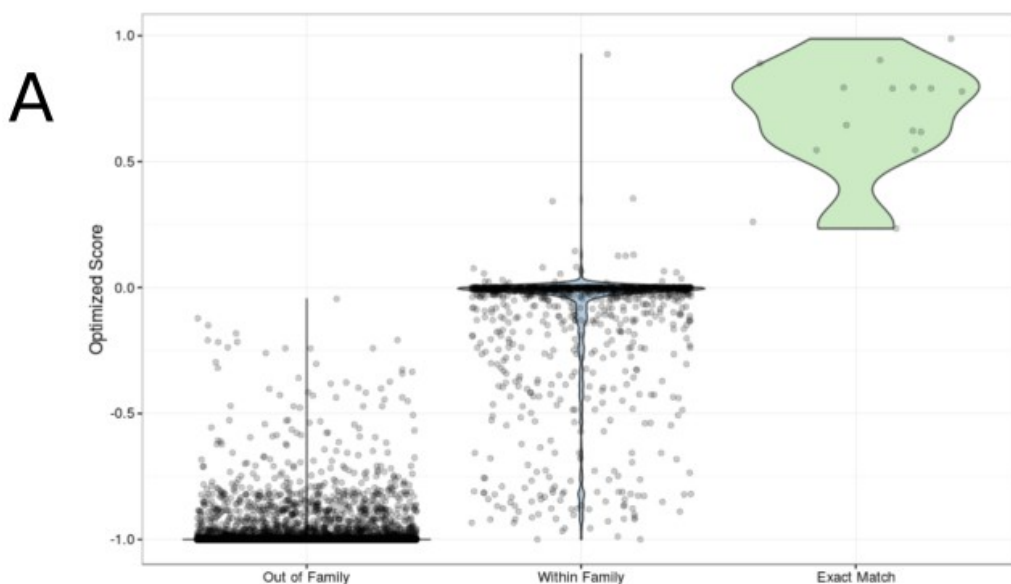


Fig. S10: Accuracy of BARLEY novelty index.

Fig. S10: Accuracy of BARLEY novelty index. (A) Validation of the BARLEY novelty index within a test dataset. A violin plot is shown with BARLEY novelty index (y-axis) and the relationship type (exact match, family match, or out of family) between encoded RiPP and chemical scaffold (x-axis). Using a cutoff of 0.2 in the BARLEY novelty index, there is a 99.7% accuracy in classifying exact matches from other comparison types. (B) Pairs of RiPP chemical scaffolds and clusters are randomly assigned and labelled according to whether they originate from the same biosynthetic family. Among these pairs, the BARLEY novelty index generated alongside the chemical similarity scores of RiPP-PRISM structure predictions (tanimoto coefficient between ECFP6 fingerprints) and the paired RiPP chemical scaffold, considering both the median and maximum of RiPP-PRISM's library of structure predictions. Using these scores, RiPP-PRISM's structure prediction was compared to the BARLEY novelty index using receiver operating characteristic curves, with their area denoted in the legend.

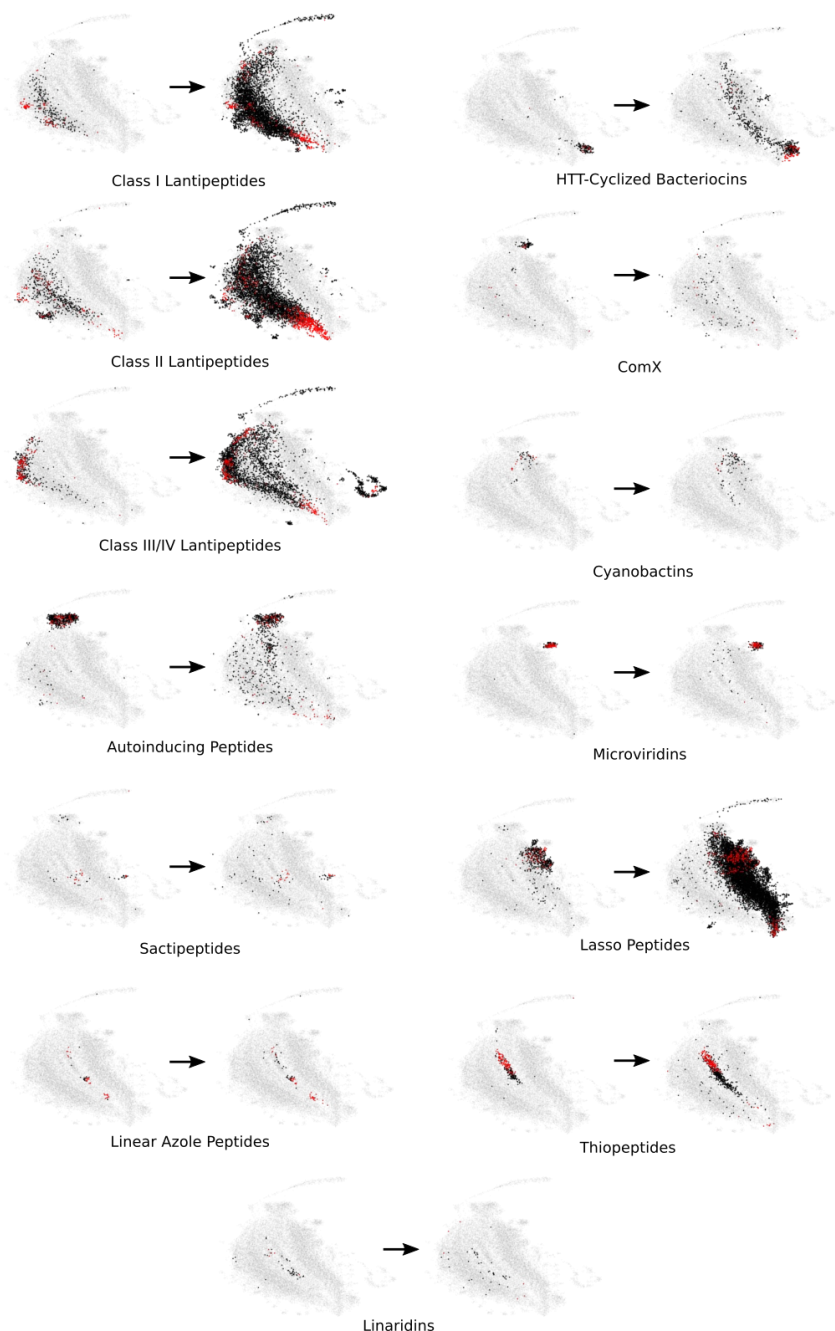


Fig. S11: Chemical diversity and novelty of all RiPPs detected by RiPP-PRISM and NLPPrecursor as measured via BARLEY. Expansion of visualization presented in Fig. 3 across all RiPP families detected by NLPPrecursor.

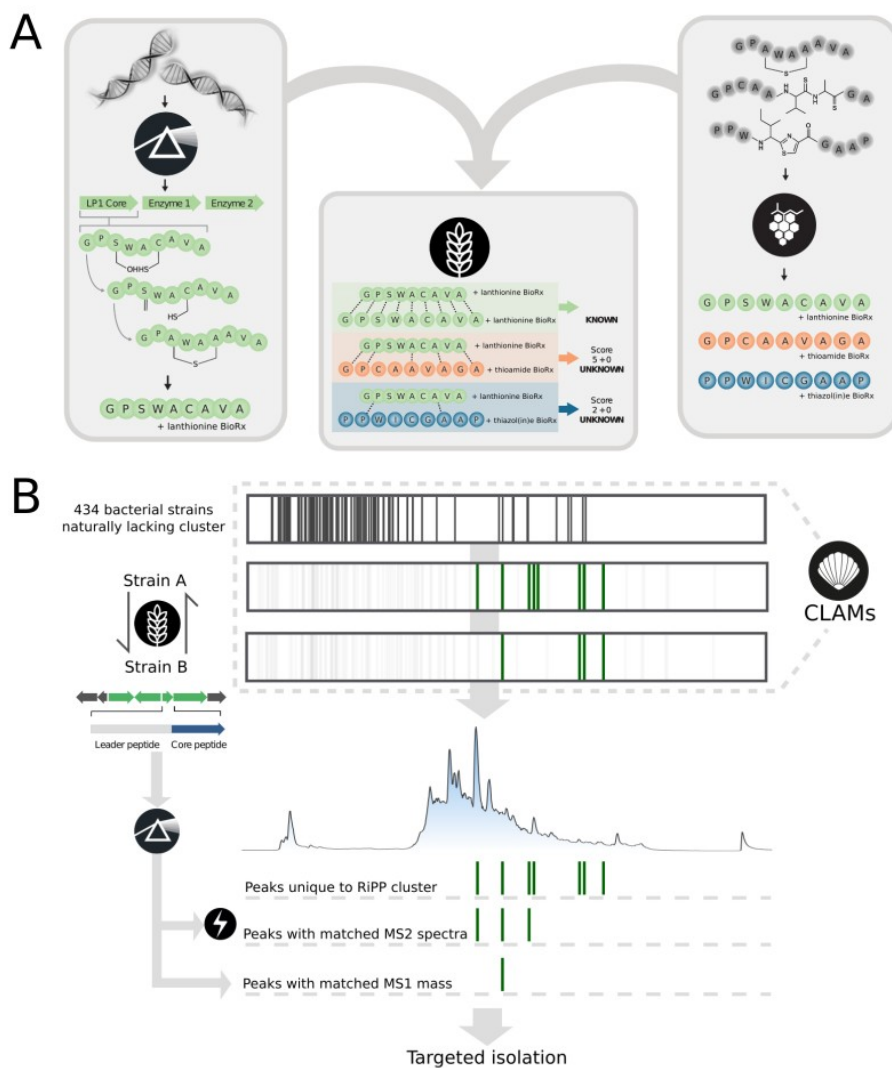


Fig. S12: Workflow for unknown RiPP discovery from paired genomic and untargeted metabolomics data using CLAMS. (A) RiPP-PRISM is used to identify putative RiPP biosynthetic gene clusters and predict the chemical structures of their products, while GRAPE is used to perform retrobiosynthesis of a large library of known RiPP structures, stored in SMILES format. BARLEY performs local alignment between genomic (RiPP-PRISM) and chemical (GRAPE) information, incorporating both the core peptide sequence and any tailoring reactions that decorate the peptide (GRAPE) or whose catalytic enzymes are found within the candidate RiPP biosynthetic gene cluster (RiPP-PRISM). (B) BARLEY can subsequently be used to identify all prokaryotic genomes that encode a biosynthetic gene cluster for a specific putative RiPP of interest, as well as the numerous strains that naturally lack the capacity to produce that product. CLAMS enables the identification of metabolites present uniquely in strains carrying the specific encoded RiPP from untargeted metabolomics data. Candidate peaks are selected by matching the exact mass of the structures predicted by RiPP-PRISM, or by *in silico* fragmentation of these predicted chemical structures and retention of tandem mass spectra with significant overlaps.

(1) deepstreptin

Top Matches	Alignment
Caulosegnin I Score: -0.03	(1) Leu Ala Gly Thr Ala Lys Asp Leu Gly Gly His Gln Leu Ile Phe Val Gly Gln Pro Glu Gly Ala Ala Val Asn Pro Leu Gly Arg Glu Ile Gln Gly
Caulonodin IV Score: -0.03	(1) Leu Ala Gly Gly Thr Ala Lys Asp Leu Leu Gly Gly His Gln Leu Ile Ile Lys Glu Ser Phe Asp Val Gly Thr Val Ser Gln Tyr Tyr Phe Ala Gly Leu
Rhodanodin Score: -0.03	(1) Leu Ala Gly Thr Ala Lys Asp Leu Leu Gly Gly His Gln Leu Ile Ile Gly Asn Glu Gly Val Leu Pro Met Gly His Ala Ala Thr Pro Gly Phe

(2) deepflavo

Top Matches	Alignment
FlvA2.b Score: -0.12	(2) Thr Ile Pro Pro Arg Thr Phe Trp Cys Asp Thr Thr Val Ile Cys Thr Gly Leu Ile Val Ala Ala Thr Thr Gly Phe Asp Trp Cys Pro Thr Gly Ala Cys Thr Tyr Ser Cys Arg Val Gly Ser Pro Leu Thr Val Thr Ile
Pinensin Score: -0.15	(2) Thr Ile Pro Pro Arg Thr Phe Trp Cys Asp Thr Thr Val Ile Cys Pro Thr His Thr Val Ala Thr Asp Asp Glu Gly His Leu Cys Thr Thr Ile Ile Cys His
FlvA2.d Score: -0.03	(2) Thr Ile Pro Pro Arg Thr Phe Trp Cys Asp Thr Thr Val Ile Cys Thr Thr Ile Val Asn Thr Thr Ile Gly Thr Leu Val Val Phe Asp Cys Pro Thr Ser Ala Cys Thr Pro Pro Cys Arg Phe Gly

(3) deepginsen

Top Matches	Alignment
Salivaricin B Score: -0.03	(3) Thr Leu Pro Thr Gln Phe Ile Ser Cys Gly Glu Thr Phe Trp Val Cys Gly Gly Val Ile Gln Thr Ile Ser His Glu Cys Arg Met Asn Trp Gln Phe Leu Phe Thr Ser Ser Ser Gly
Prochlorosin 3.3 Score: -0.17	(3) Thr Leu Pro Thr Thr Gln Phe Ile Ser Cys Gly Glu Thr Phe Trp Val Cys Asp Thr Gly Ile Gln Ala Val Leu His Thr Ala Gly Cys Tyr Gly Gly Thr Lys Met Cys Arg Ala Gly
Ruminococcin A Score: -0.17	(3) Thr Leu Pro Thr Thr Gln Phe Ile Ser Cys Gly Glu Thr Phe Trp Val Cys Asn Gly Val Leu Lys Thr Ile Ser His Glu Cys Asn Met Asn Thr Trp Gln Phe Leu Phe Thr Ser Ser Cys Gly

Fig. S13: Alignments of deepstreptin, deepflavo, and deepginsen to best matching RiPPs, as determined by BARLEY. Each alignment represents a query genomically encoded RiPP to a subject known chemical scaffold stored in SMILES format and

subject to retrobiosynthesis by GRAPE. Matched residues are shown in red, while unmatched residues are depicted in grey. Amino acids are represented by their three-letter codes where possible, but in cases where definitive retrobiosynthesis is not possible, a combination is depicted (e.g. SerCys for amino acids linked by lanthionine linkages).

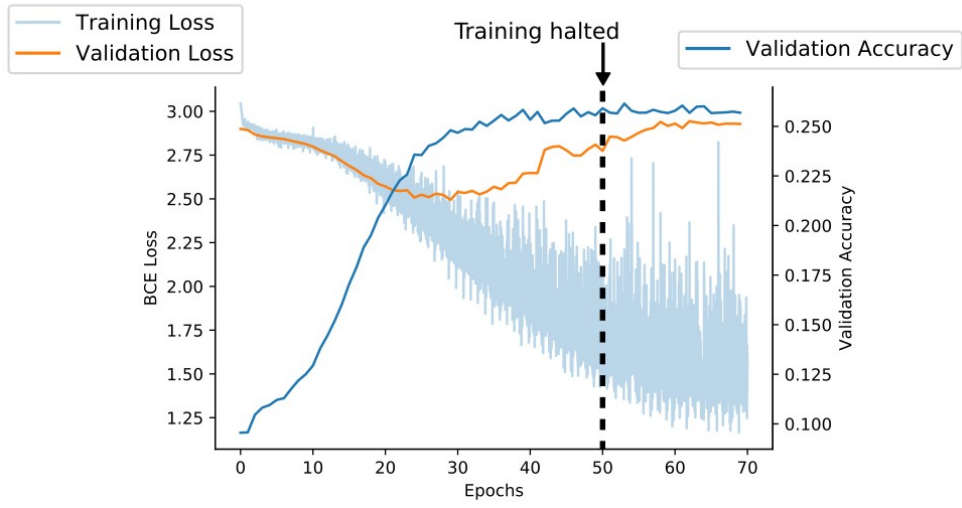
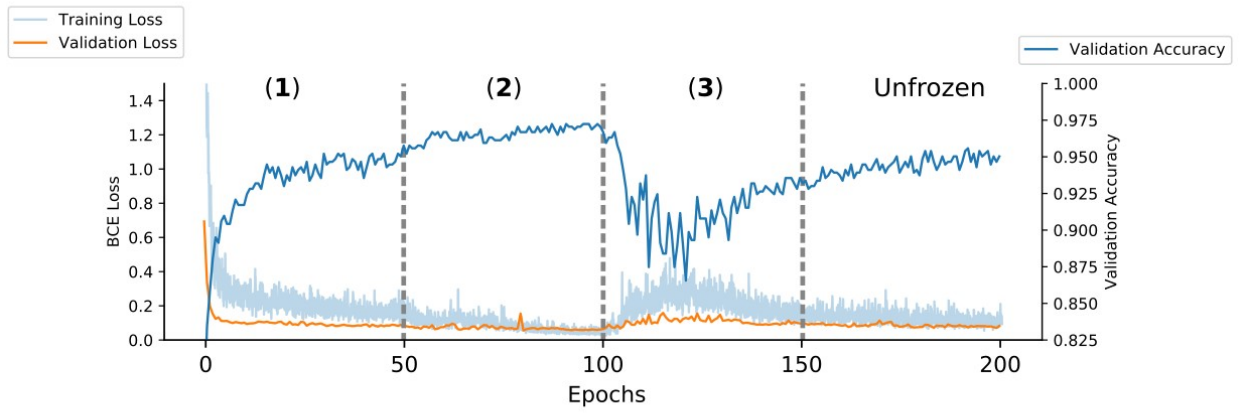
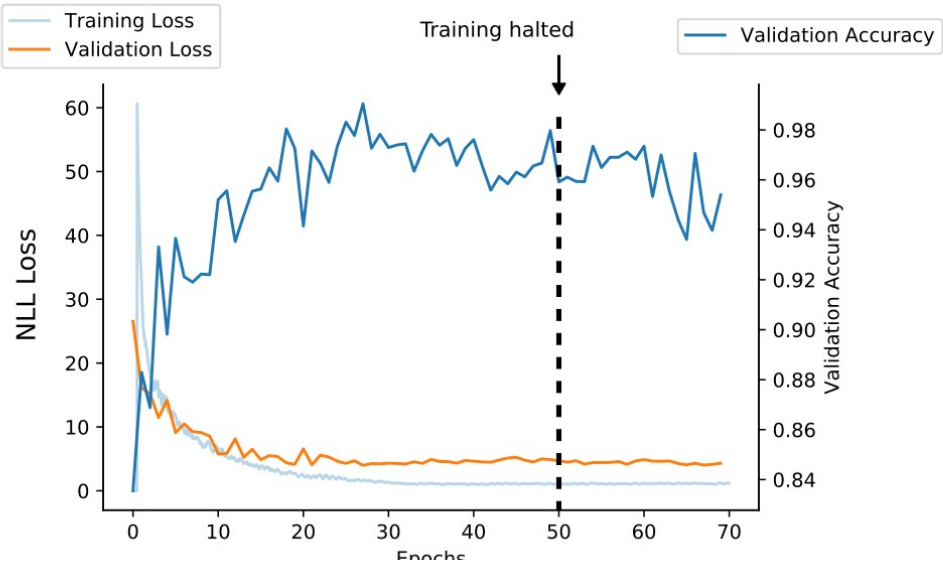
A**B****C**

Fig. S14: Monitoring NLPPrecursor validation loss across training iterations.

NLPPrecursor consists of two modules, a classification (A-B) and annotation (C) module. The classification module is trained in two stages where first (A) an unsupervised language model is trained to predict the next AA given a leading portion of the RiPP precursor peptide. In the next stage (B), the same language model is trained in a supervised fashion to predict the family of a given ORF using gradual unfreezing where the first (1), second (2), third (3) and all layers are unfrozen after training at each stage for 50 epochs. In both stages, the binary cross entropy (BCE) loss is tracked within the training and validation datasets, while the accuracy is tracked in the validation dataset. (C) The cleavage module of NLPPrecursor is trained over the course of 50 epochs, where negative log likelihood (NLL) loss is tracked within the training and validation datasets while the accuracy is tracked in the validation dataset.

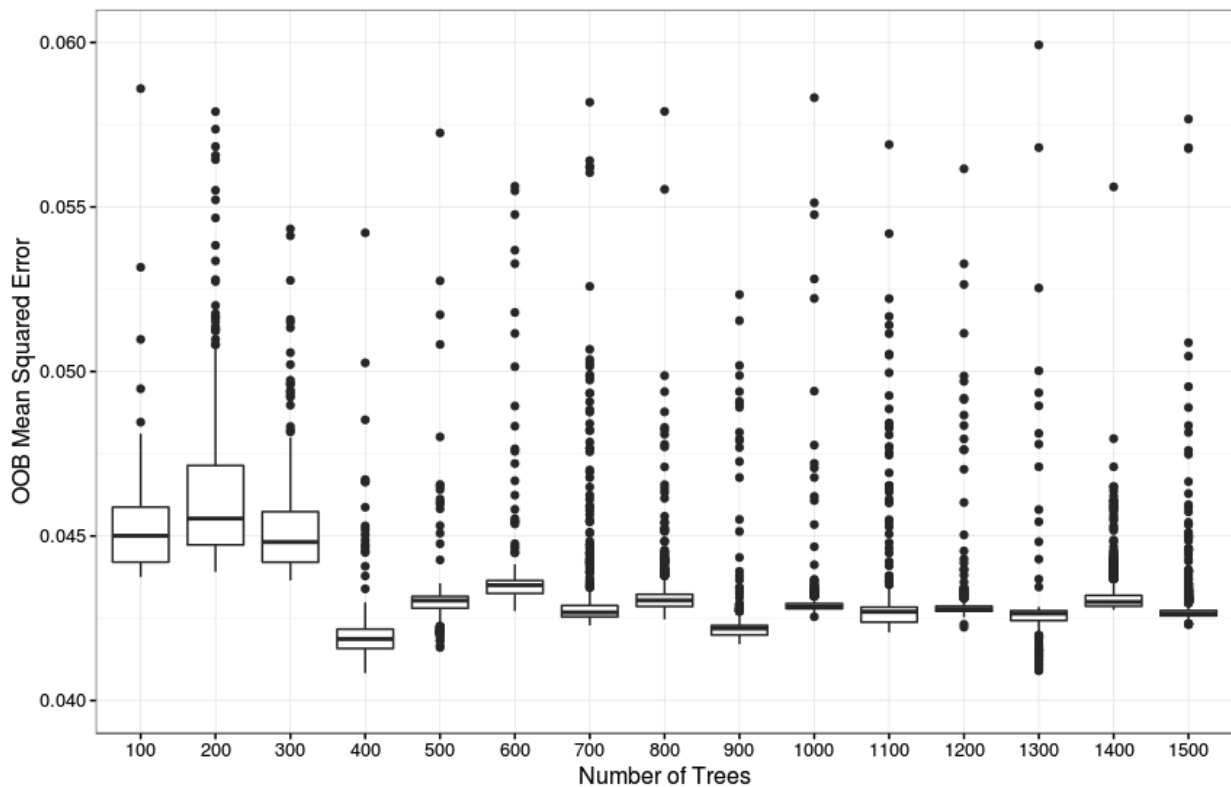


Fig. S15: Hyperparameter optimization of random forest for BARLEY novelty index: number of trees. Random forests are ensemble methods which use many less accurate

estimators to generate a final decision. Through iterating through the number of base estimators, it was found that median error reaches an approximate minimum around 400 trees.

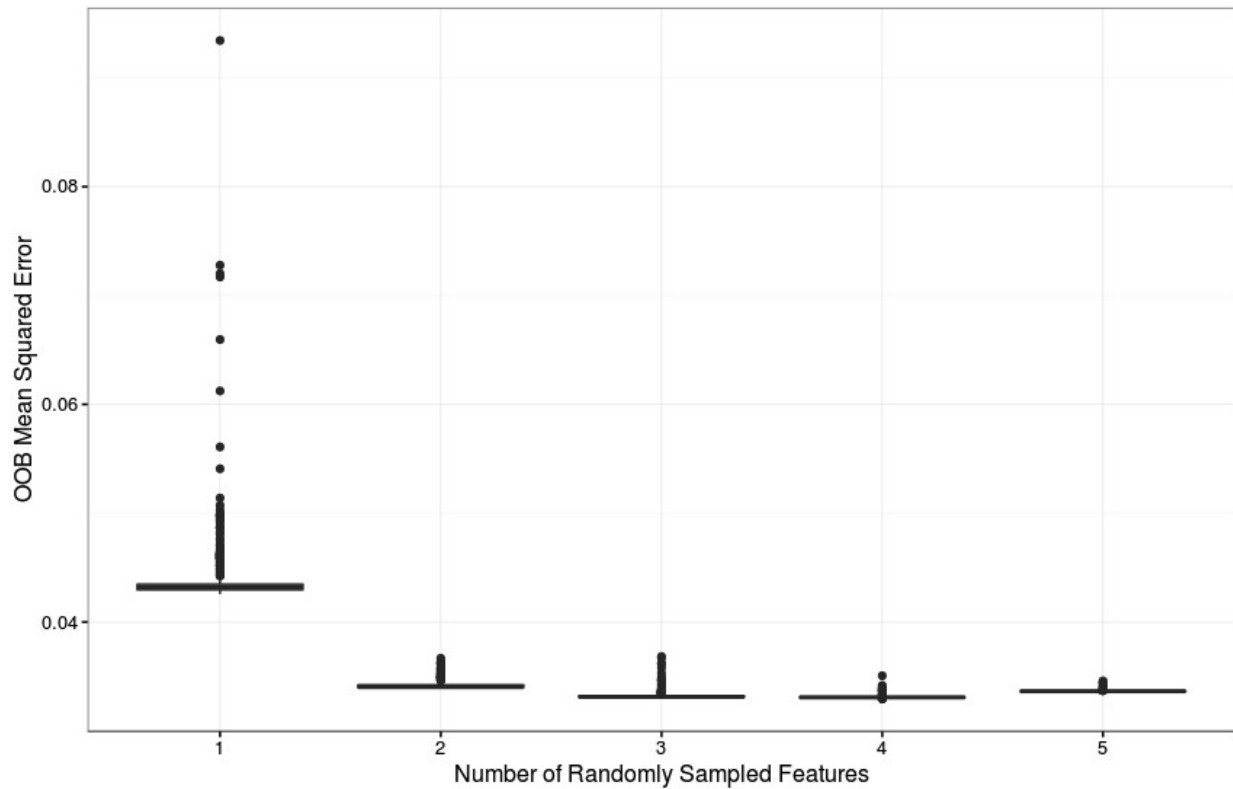


Fig. S16: Hyperparameter optimization of random forest for BARLEY novelty index: number of randomly sampled features. In a random forest, each decision tree can be randomly assigned a set of features to predict upon, a method known as feature bagging. Of a total of five features, this model performed optimally when each decision tree was able to access all five features.

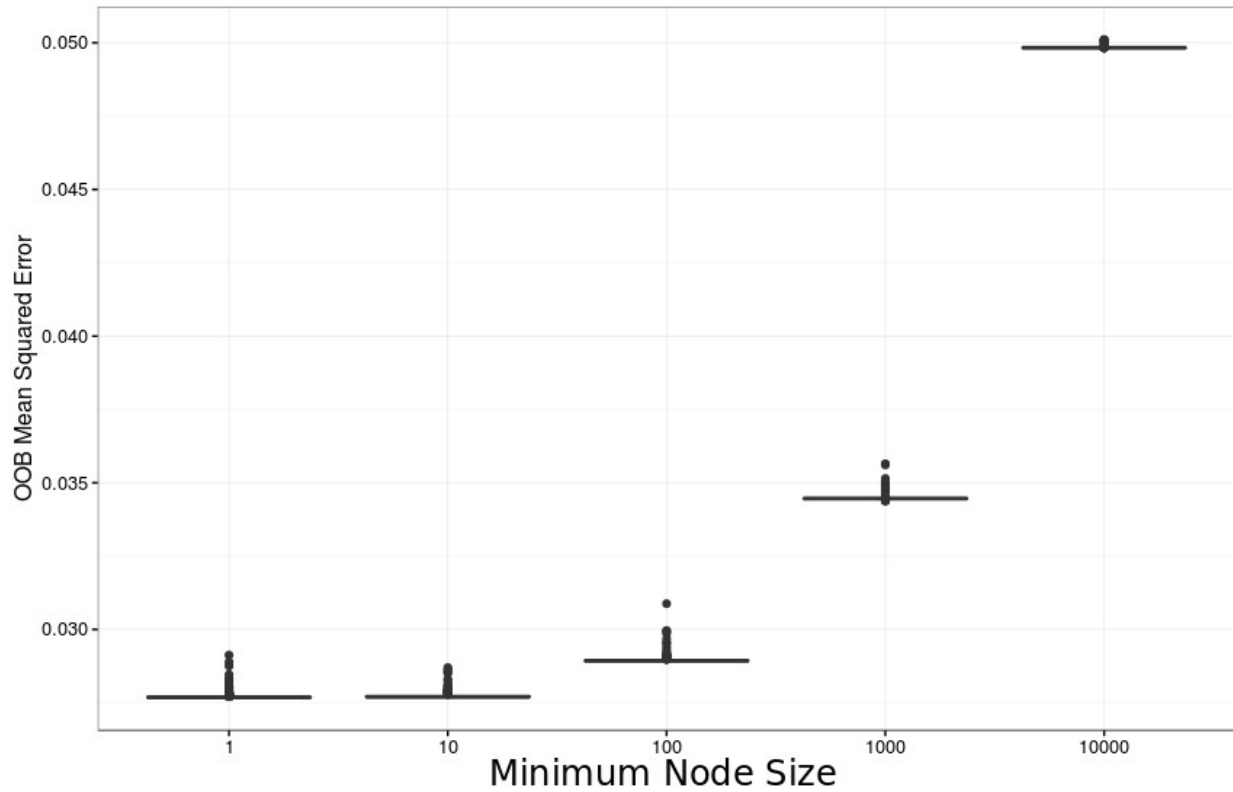


Fig. S17: Hyperparameter optimization of random forest for BARLEY novelty index: minimum node size. Minimum node size represents the minimum size of the terminal node in a regression tree where smaller sizes develop deeper and more complex trees. The most accurate models were built with a minimum node size of 1.

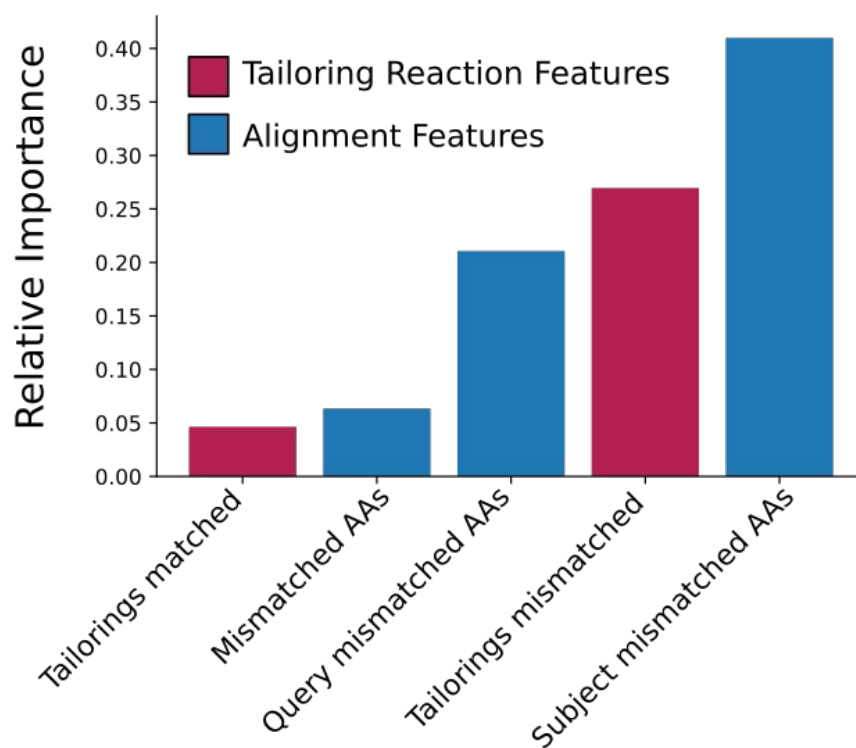


Fig. S18: Relative feature importance for BARLEY genome to chemical comparisons.

Features based on tailoring reactions are coloured in red while core peptide alignment features are coloured in blue. Relative importance is calculated according to mean decrease in impurity as implemented in scikit-learn (v3.2.4)[20] such that the sum of feature importance values is equal to one.

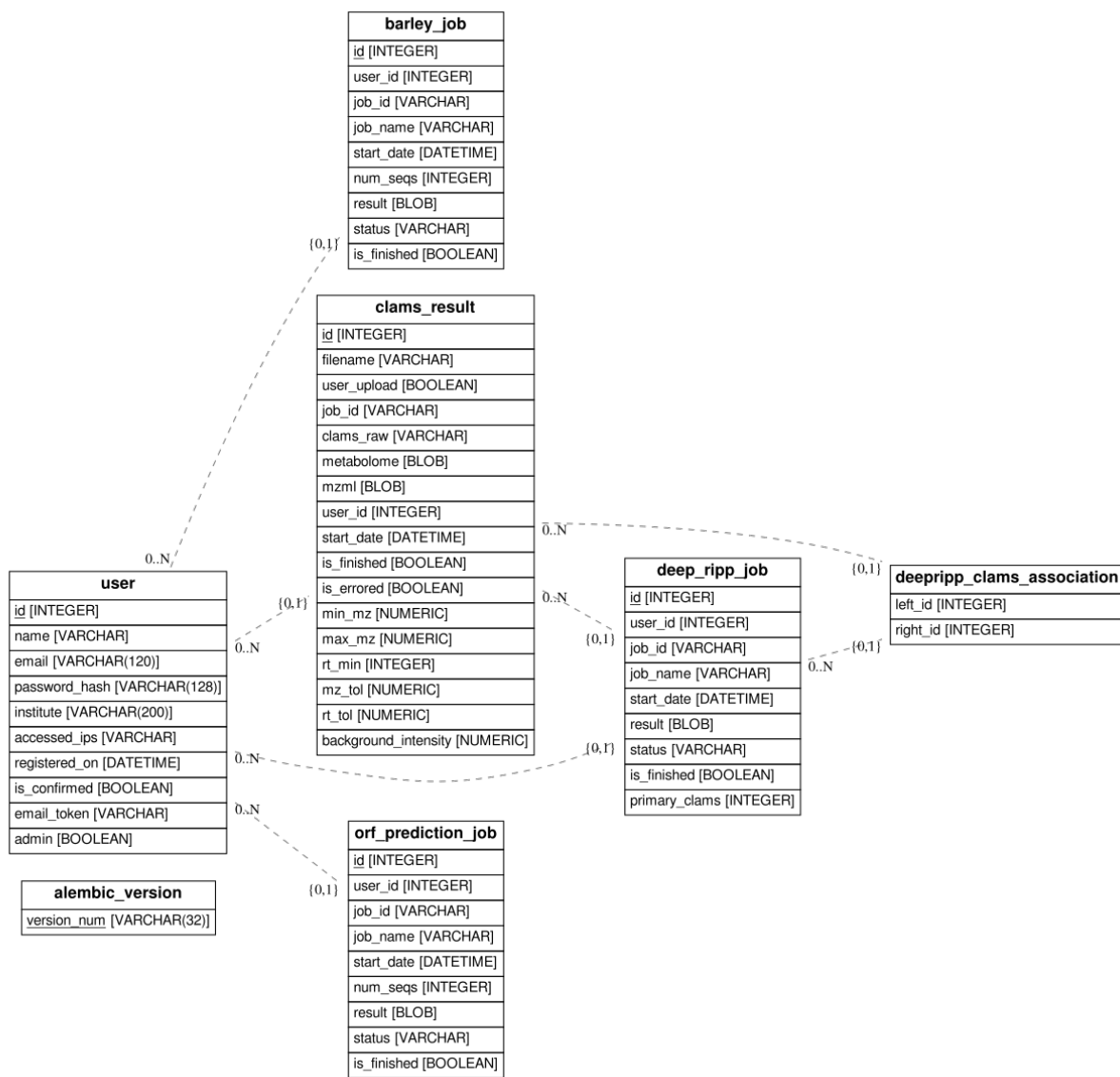


Fig. S19: Entity relationship diagram of DeepRiPP SQLite database backend used to manage user registration, login and job management. Each table is represented by its name, and column names (with data types shown in parentheses). Relationships between tables are shown as dotted lines annotated according to their one to many (0..N), and one to zero or one (0, 1). This database schema was generated through SQLAlchemy (v2.4) with migrations managed via alembic (v1.0.10).

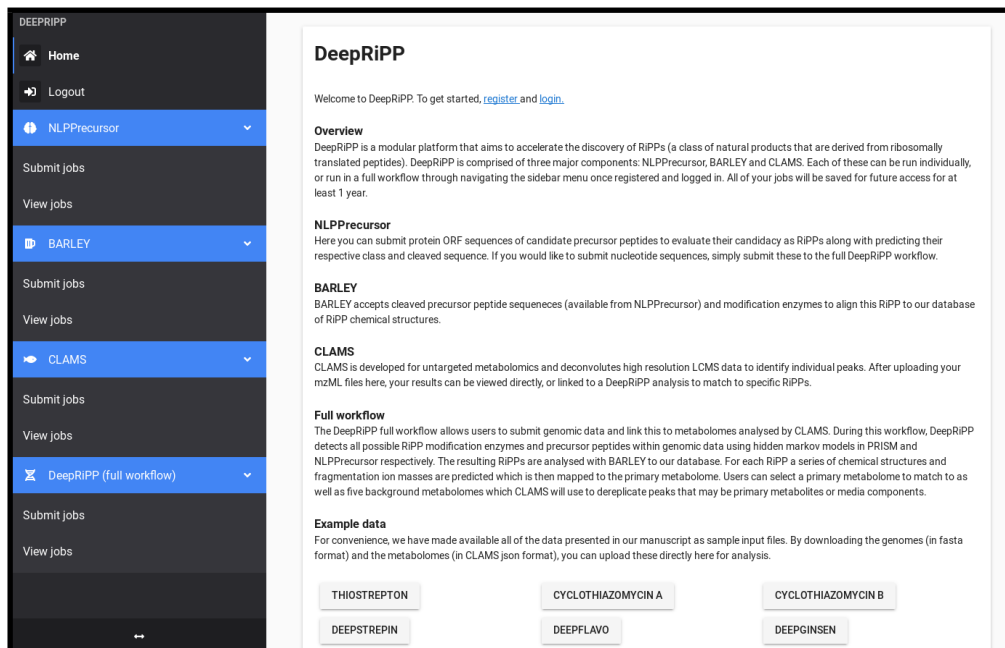


Fig. S20: DeepRiPP Homepage depicting various job tasks, instruction manual and example datasets.

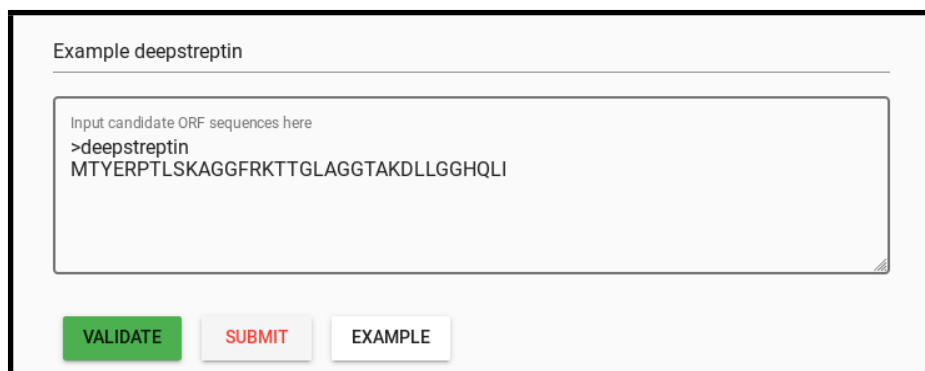


Fig. S21: Submitting a protein sequence for NLPrecursor analysis.

NLPPrecursor Jobs

search

Job Name	Status	Start Date ↓
Example deepstreptin	Complete	Tue, 04 Jun 2019 18:26:23 GMT

deepstreptin

Submitted sequence:
 MTYERPTLSKAGGFRKTTG **LAGGTAKDLLGGHQLI**

Cleaved sequence:
 LAGGTAKDLLGGHQLI

Predicted class: **LASSO_PEPTIDE** Class prediction probability: **100 %**

Fig. S22: NLPPrecursor results and analysis.

BARLEY

BARLEY is a machine learning software that compares RiPPs to each other to determine their relative similarity. In doing so, we can identify cryptic RiPPs within genomes that are yet uncharacterized by comparing them to a database of known RiPP scaffolds. To use this tool, enter a cleaved precursor peptide sequence along with any modification enzymes. BARLEY automatically computes to predicted modifications and compares this predicted RiPP to our database.

Example nisin

Cleaved ORF sequence
 ITSISLCTPGCKTGALMGCNMKTATCHCSIHVSK

ADD MODIFICATION ENZYMES ↓ LanB ✕ LanC ✕


SUBMIT EXAMPLE You're all set to submit!

Fig. S23: Submitting a cleaved ORF sequence, and accompanying tailoring enzymes, for BARLEY analysis.

CLAMS


CLAMS (Computational Library for the Analysis of Mass Spectra) is a tool that identifies specific peaks in mass spectrum data through aggregating isotopic distributions across retention times and m/z windows. CLAMS is an essential tool in mapping chemical matter in untargeted LCMS experiments, and typically takes ~10 minutes for most analysis. Data submitted here can be viewed as the peaks alone, or can be combined with the DeepRIPP workflow to map genomic predictions to peak targets for downstream isolation and elucidation.

Upload MZML file:

UPLOAD 

SUBMIT

RM_17-05-19_13425_Mediablank_KE_HP-20bags_C12.mzml

CLAMS Settings (advanced) 

Minimum m/z value
150

Maximum m/z value
2000

Minimum retention time to scan
60

M/z tolerance
0.005

Retention time tolerance
30

Background noise intensity
1000

Fig. S26: Submitting a metabolomic dataset for CLAMS analysis.

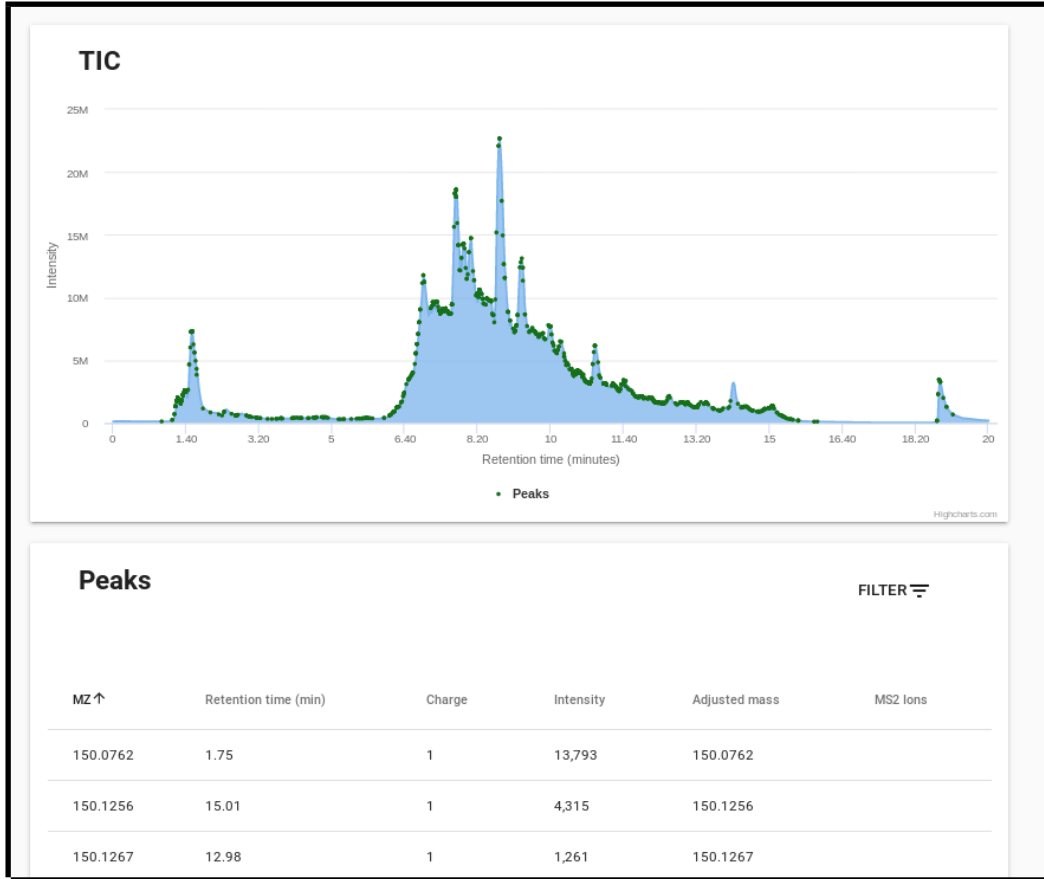


Fig. S27: CLAMS visualization without accompanying genomic data.

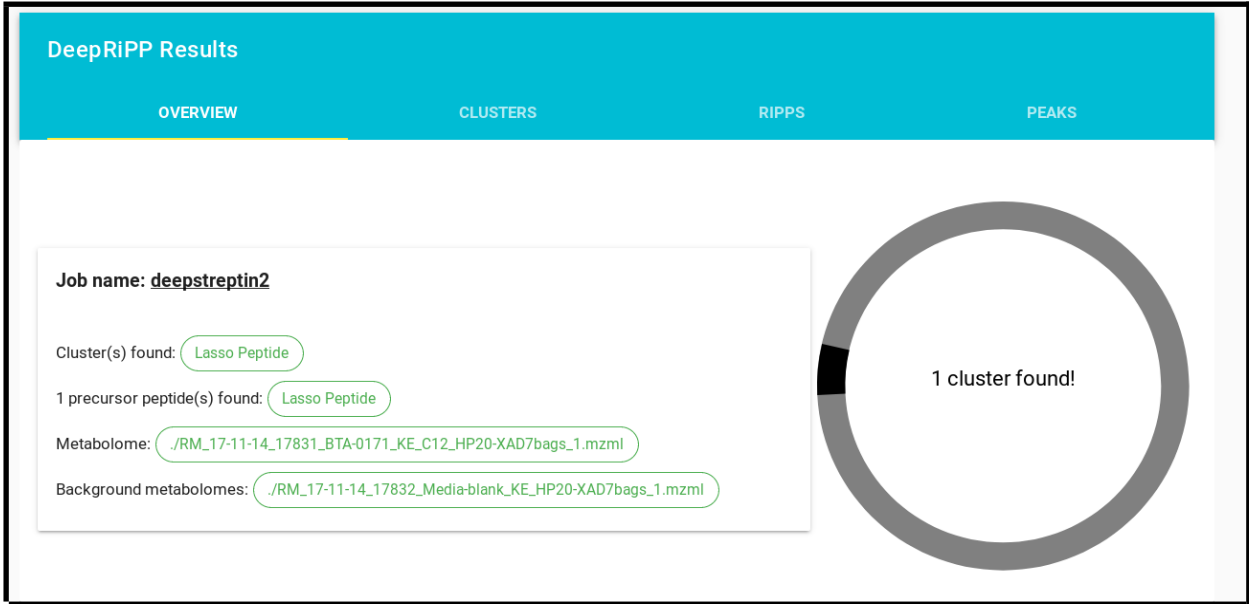


Fig. S28: Overview of DeepRiPP results page.

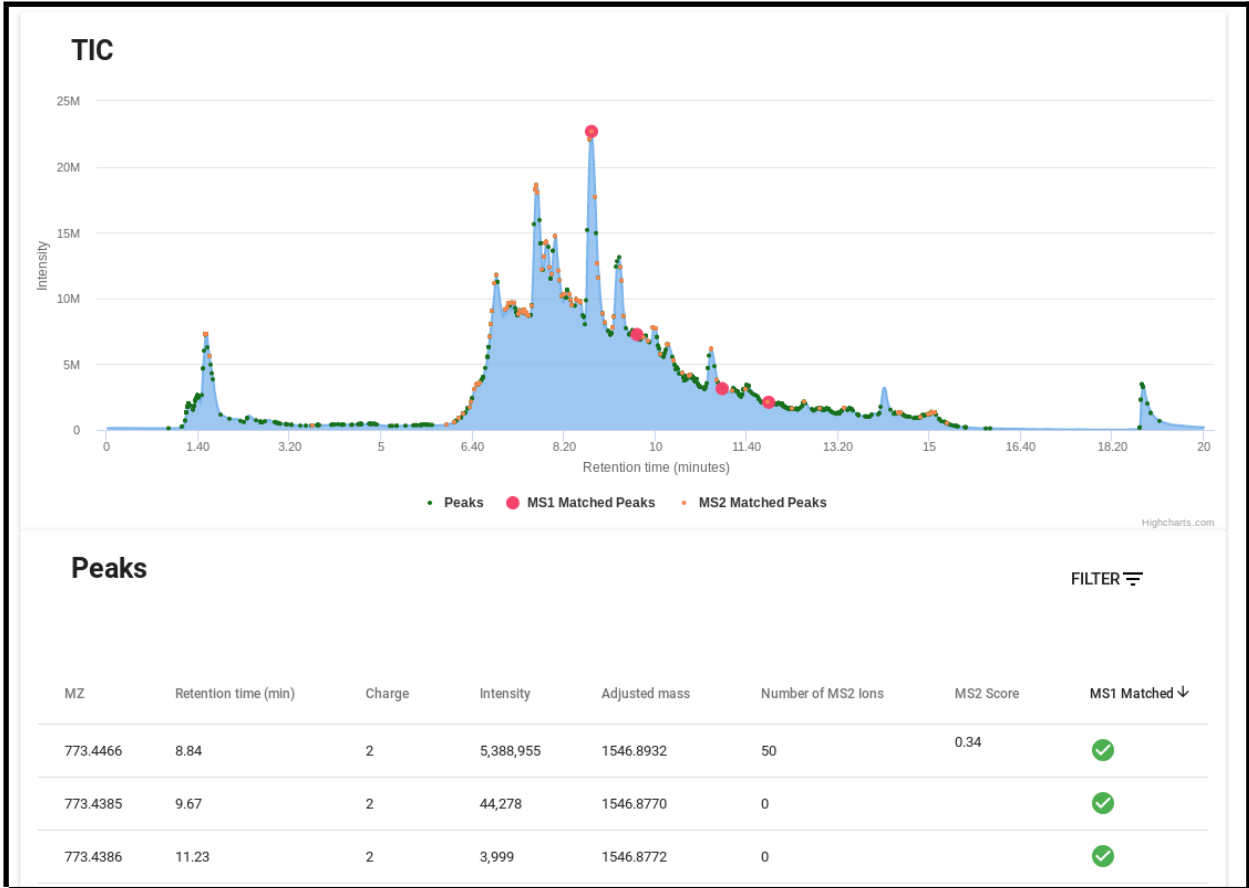


Fig. S29: Metabolomic view of DeepRiPP results for paired genomic and metabolomic data from *Streptomyces* sp. BTA0171, highlighting peaks to isolate corresponding to deepstreptin.



Fig. S30: DeepRiPP identifies cyclothiazomycin A within the genome and metabolome of *Streptomyces mobaraensis* DSM 40847.



Fig. S31: DeepRiPP identifies cyclothiazomycin B within the genome and metabolome of *Streptomyces corchorusii* DSM 40340 and its corresponding extract.

A

RiPP 1 Thiopeptide Known

MS1 Matched Peaks: 16
MS2 Matched Peaks: 493

Modification enzymes:

LazB

P450

TsrI

LazC

TsrE

LazE

OMT

TsrD

CapC

GATase

TsrU

LazF

LazC

LazC

TsrT

TsrA

LazB

Cleaved sequence
SASCTTCICTCSCSS

Predicted structure
[VIEW STRUCTURE](#)

BARLEY Matches

Thiostrepton A

Score: 0.67 [VIEW ALIGNMENT](#) [VIEW STRUCTURE](#)

Thiostrepton

Score: 0.67 [VIEW ALIGNMENT](#) [VIEW STRUCTURE](#)

SIOMYCIN-C

Score: 0.67 [VIEW ALIGNMENT](#) [VIEW STRUCTURE](#)

B



Fig. S32: DeepRiPP identifies thiostrepton within the genome and metabolome of *Streptomyces sp.* BTA-0473.

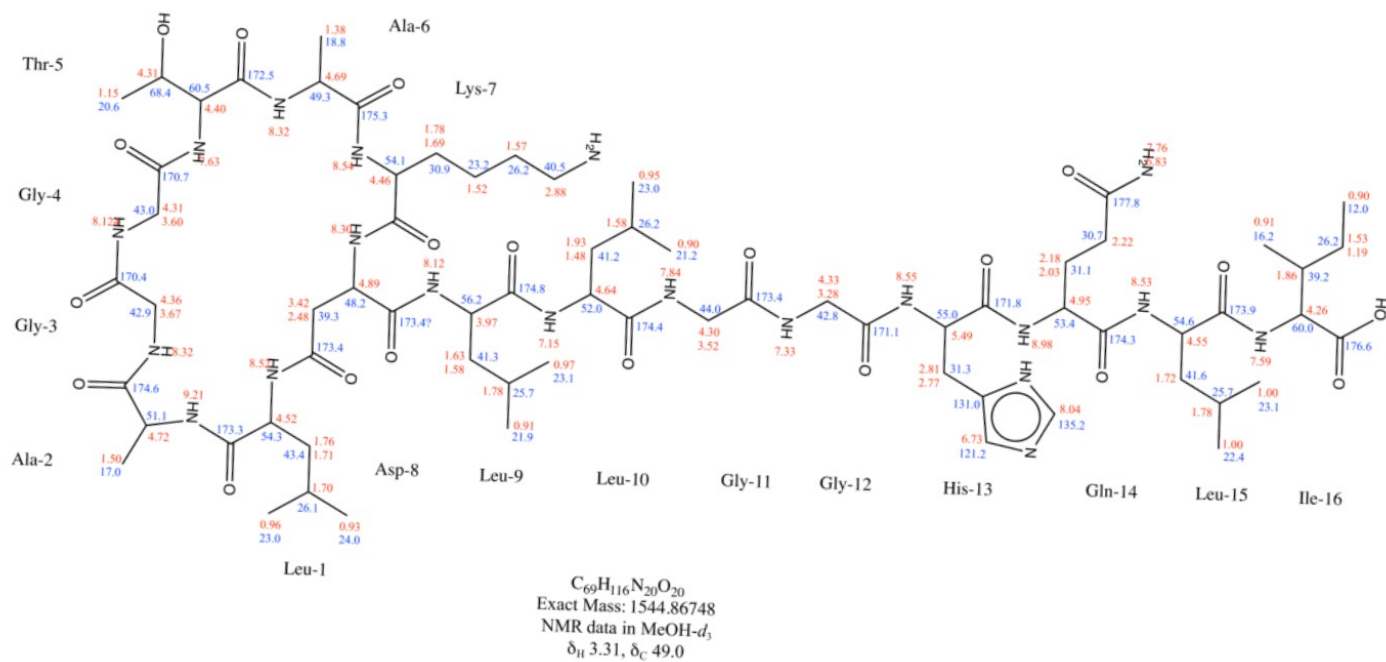


Fig. S33: Structure of deepstreptin with assigned chemical shifts. Proton chemical shifts are shown in red while carbon chemical shifts are shown in blue.

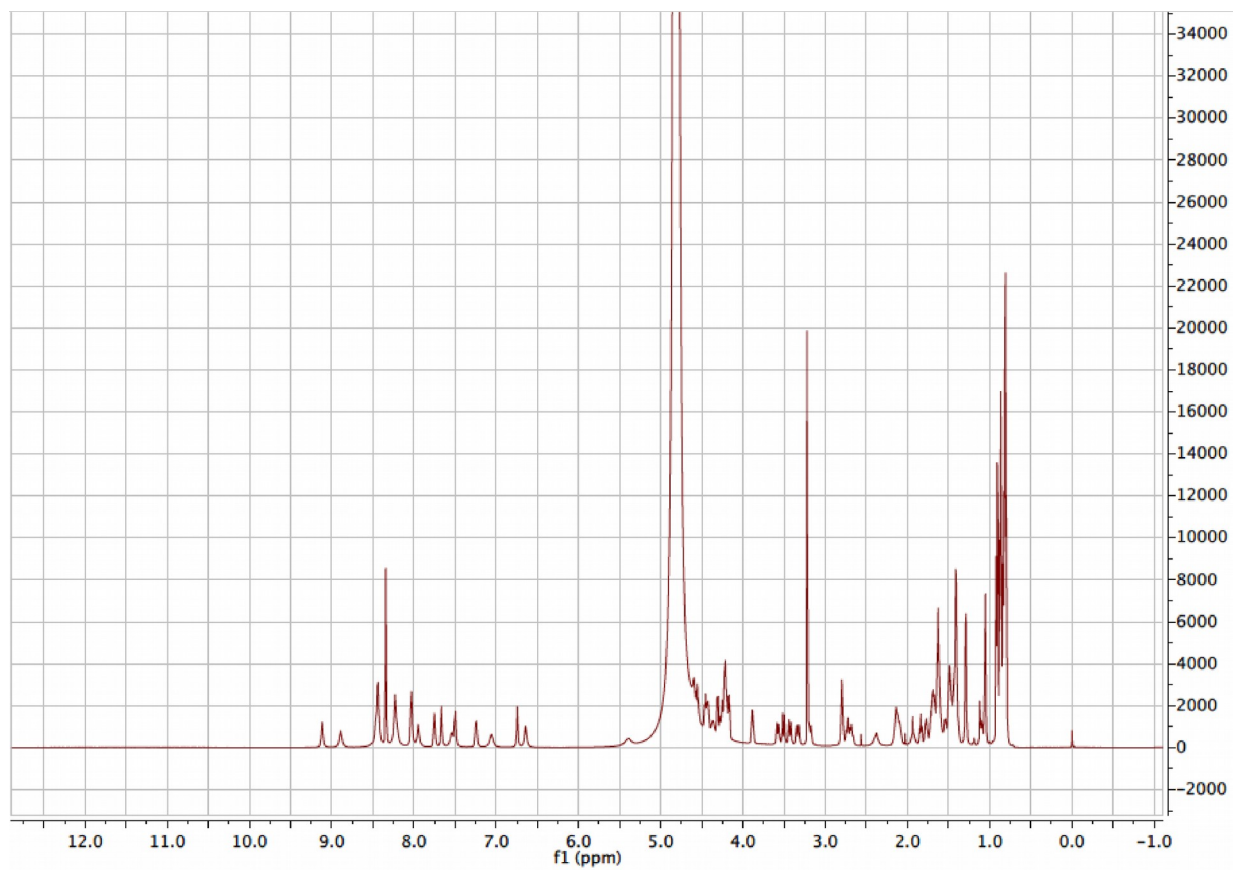


Fig. S34: $^1\text{H-NMR}$ spectrum of deepstreptin in methanol- d_3 .

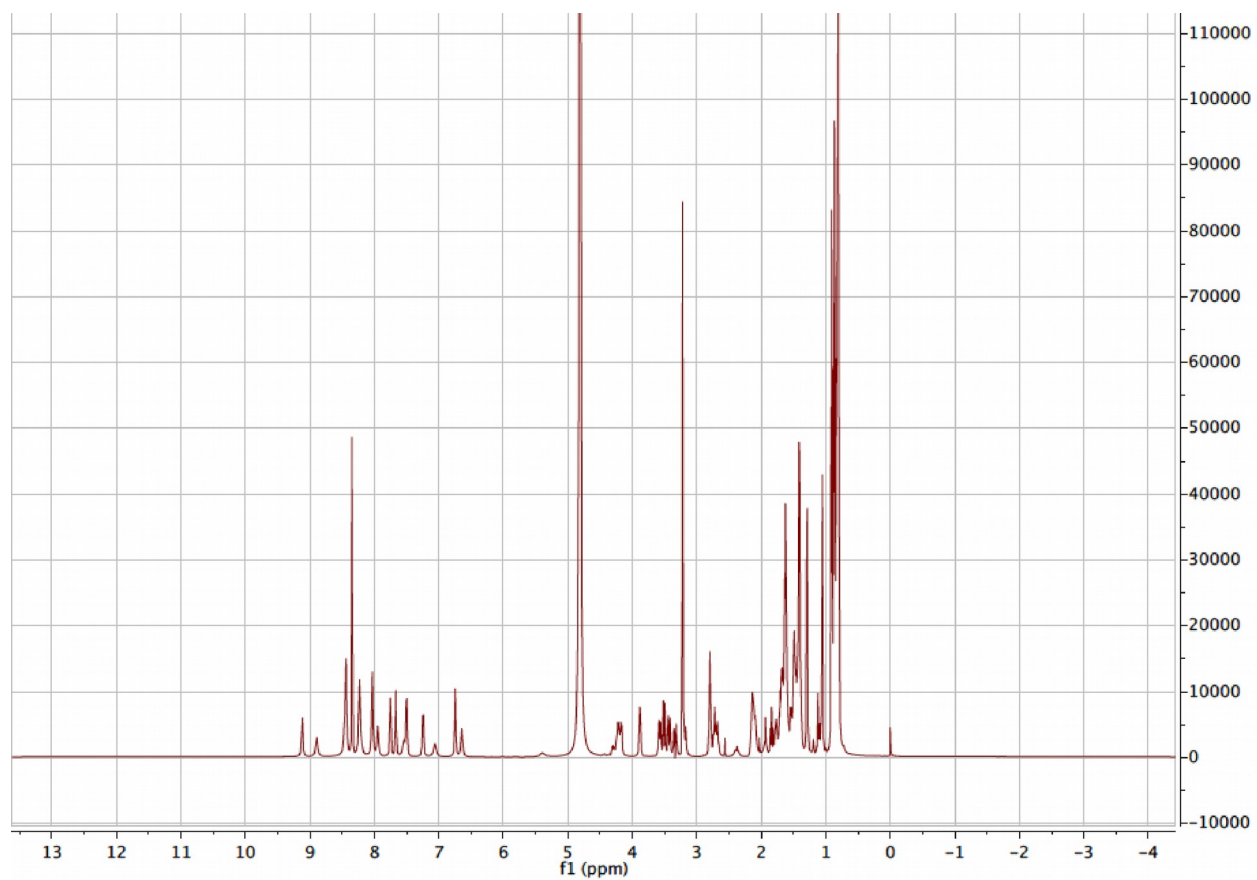


Fig. S35: $^1\text{H-NMR}$ (water suppression) spectrum of deepstreptin in methanol- d_3

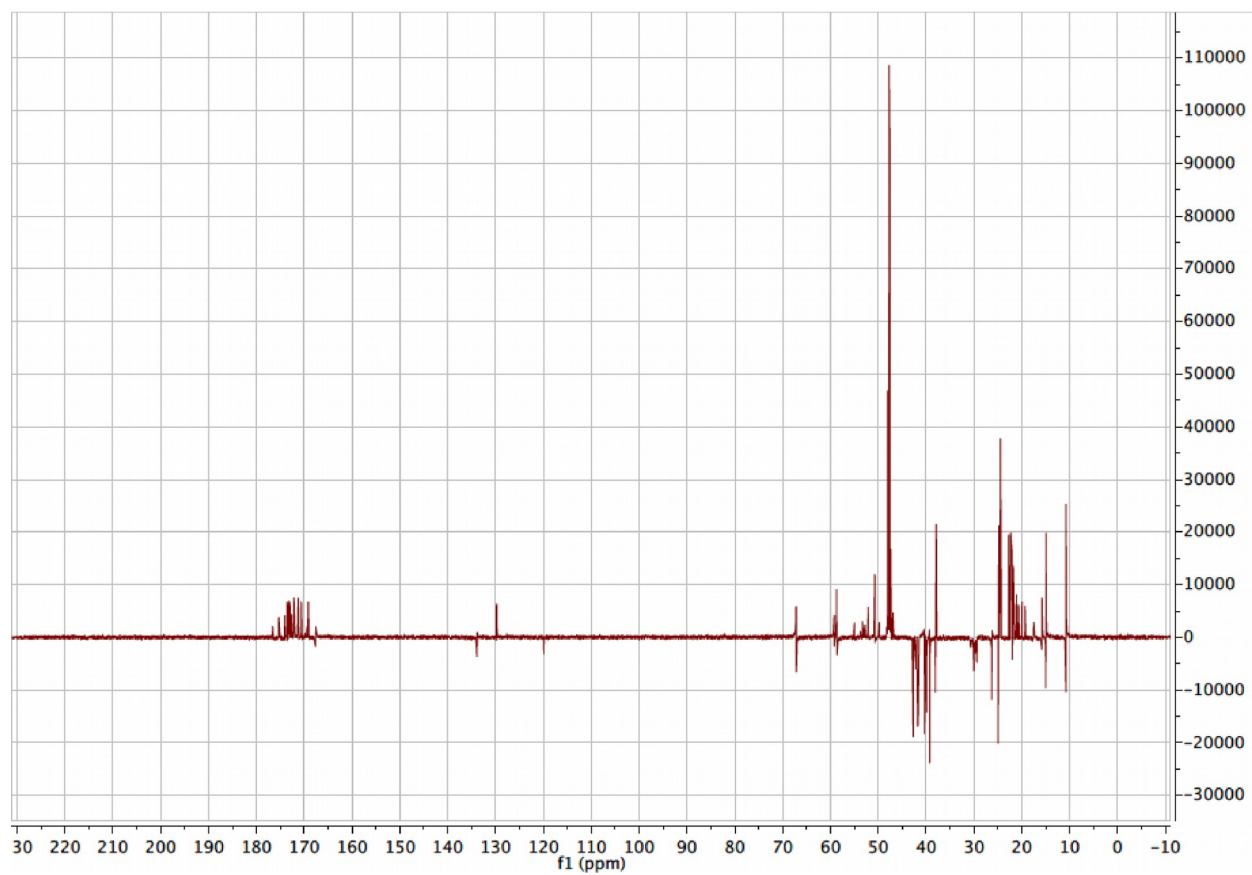


Fig. S36: DEPTq spectrum of deepstreptin in methanol-d3.

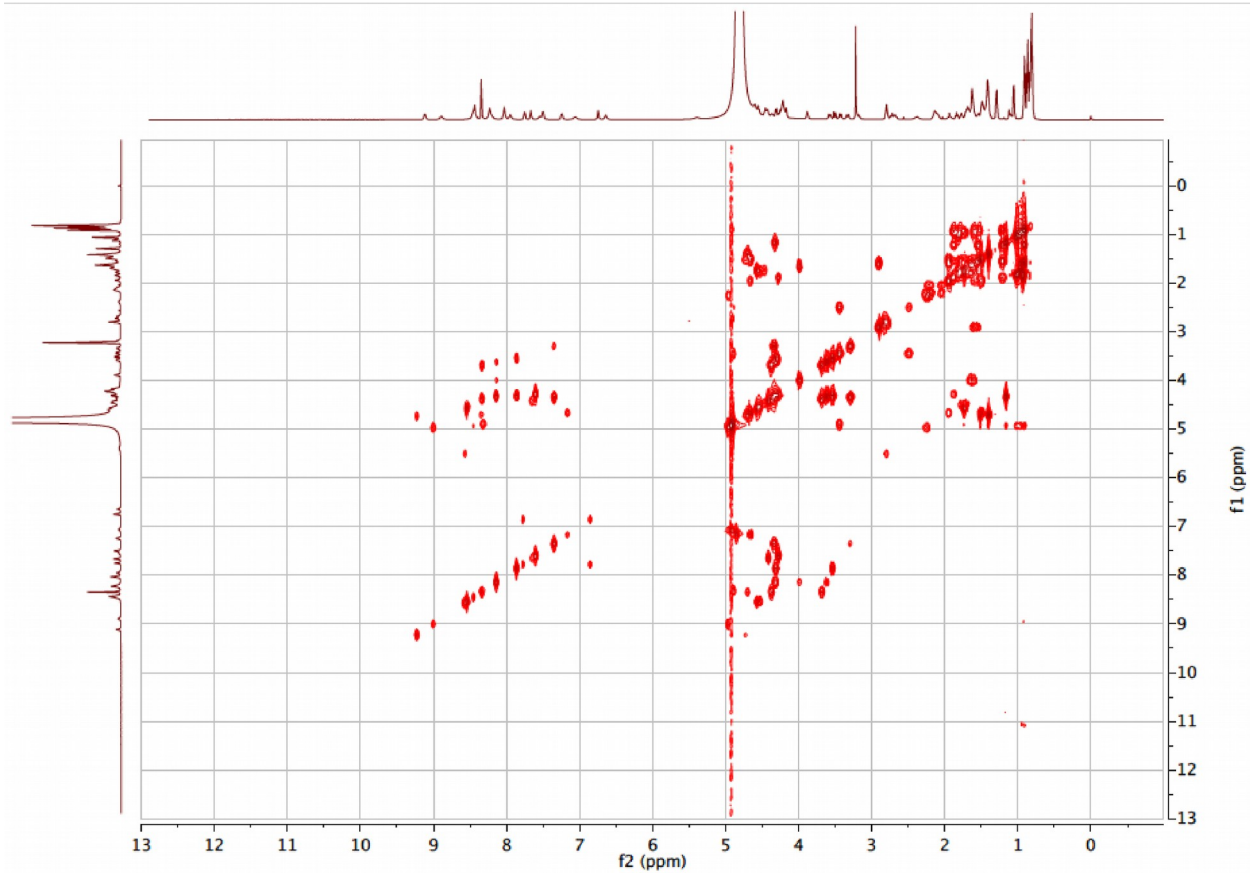


Fig. S35: ^1H - ^1H COSY spectrum of deepstreptin in methanol- d_3 .

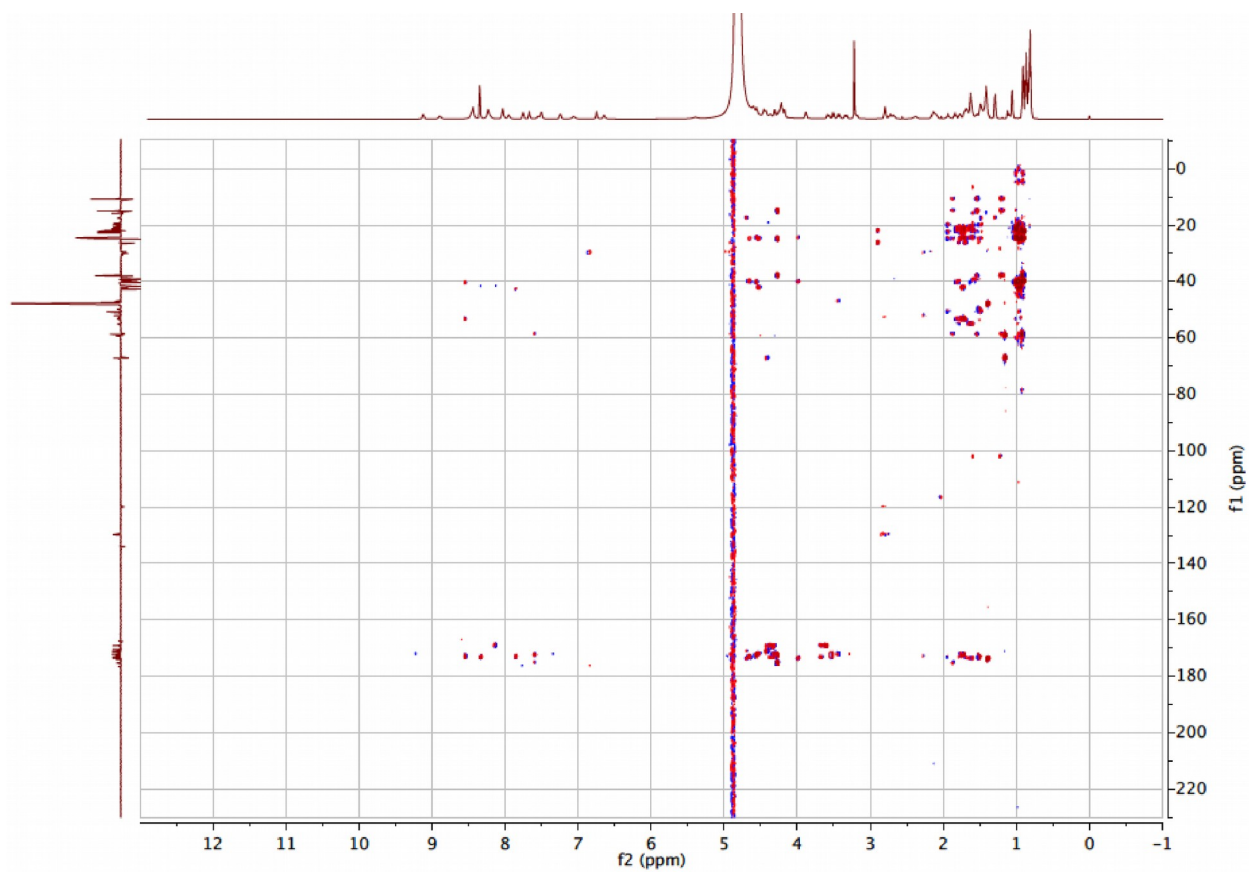


Fig. S36: ^1H - ^{13}C HMBC spectrum of deepstreptin in methanol- d_3 .

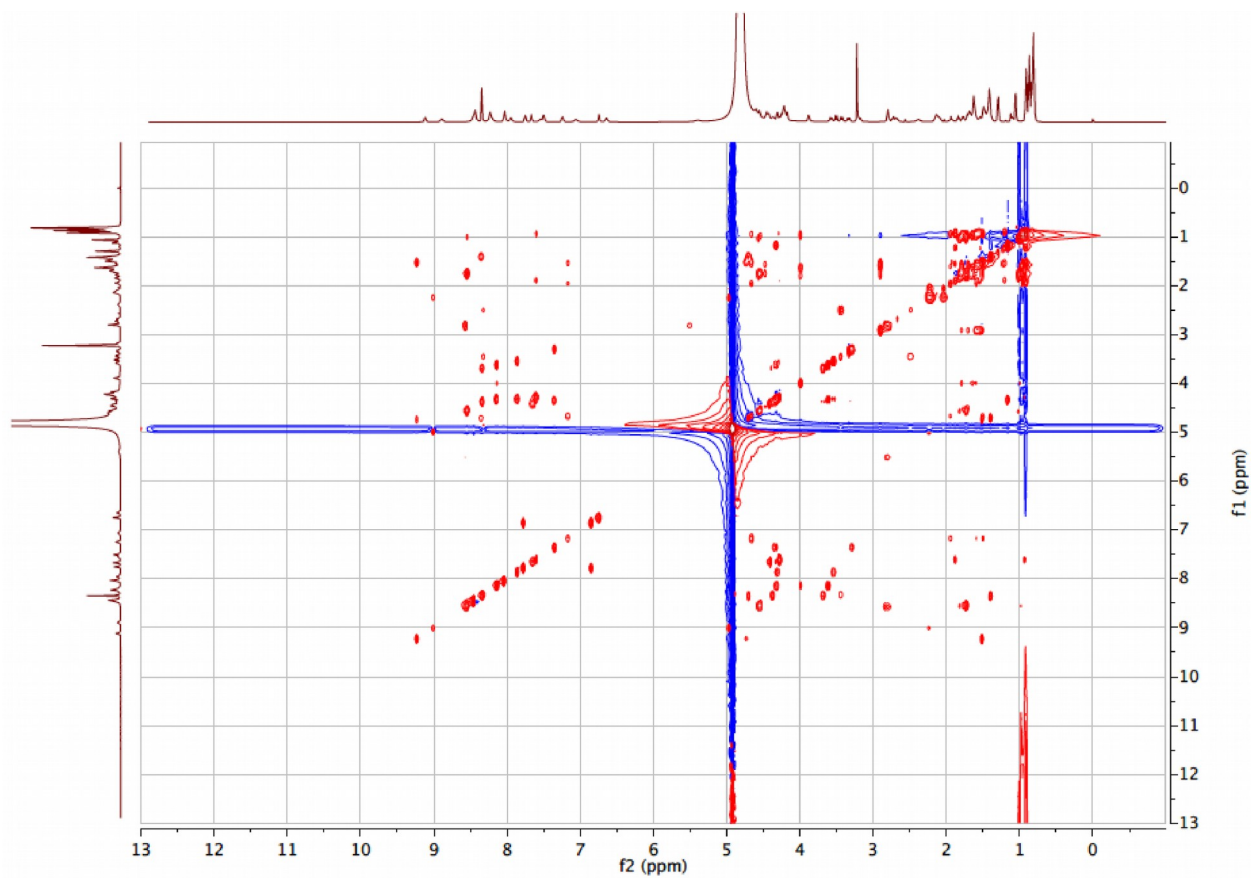


Fig. S37: ^1H - ^1H TOCSY spectrum of deepstreptin in methanol- d_3 .

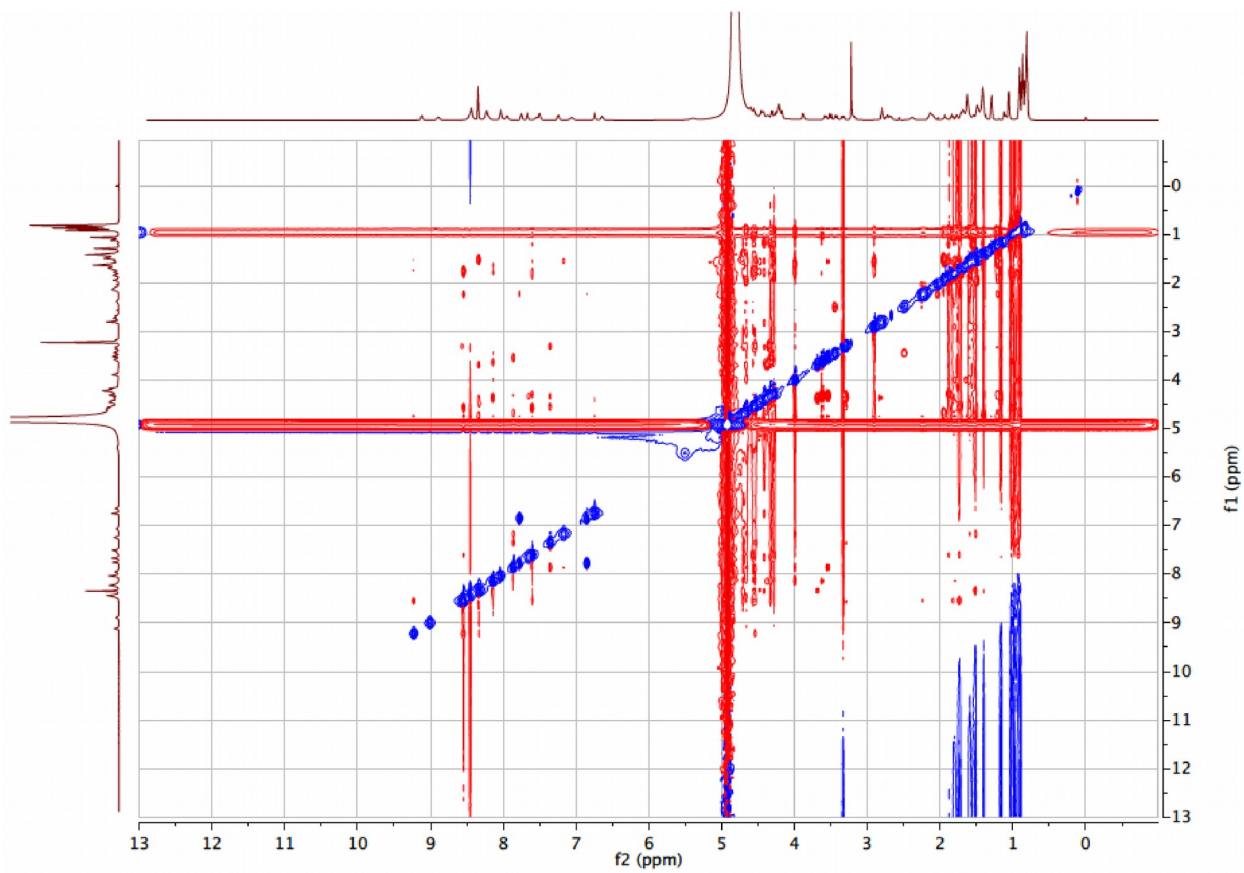


Fig. S38: ^1H - ^1H ROESY spectrum of deepstreptin in methanol- d_3 .

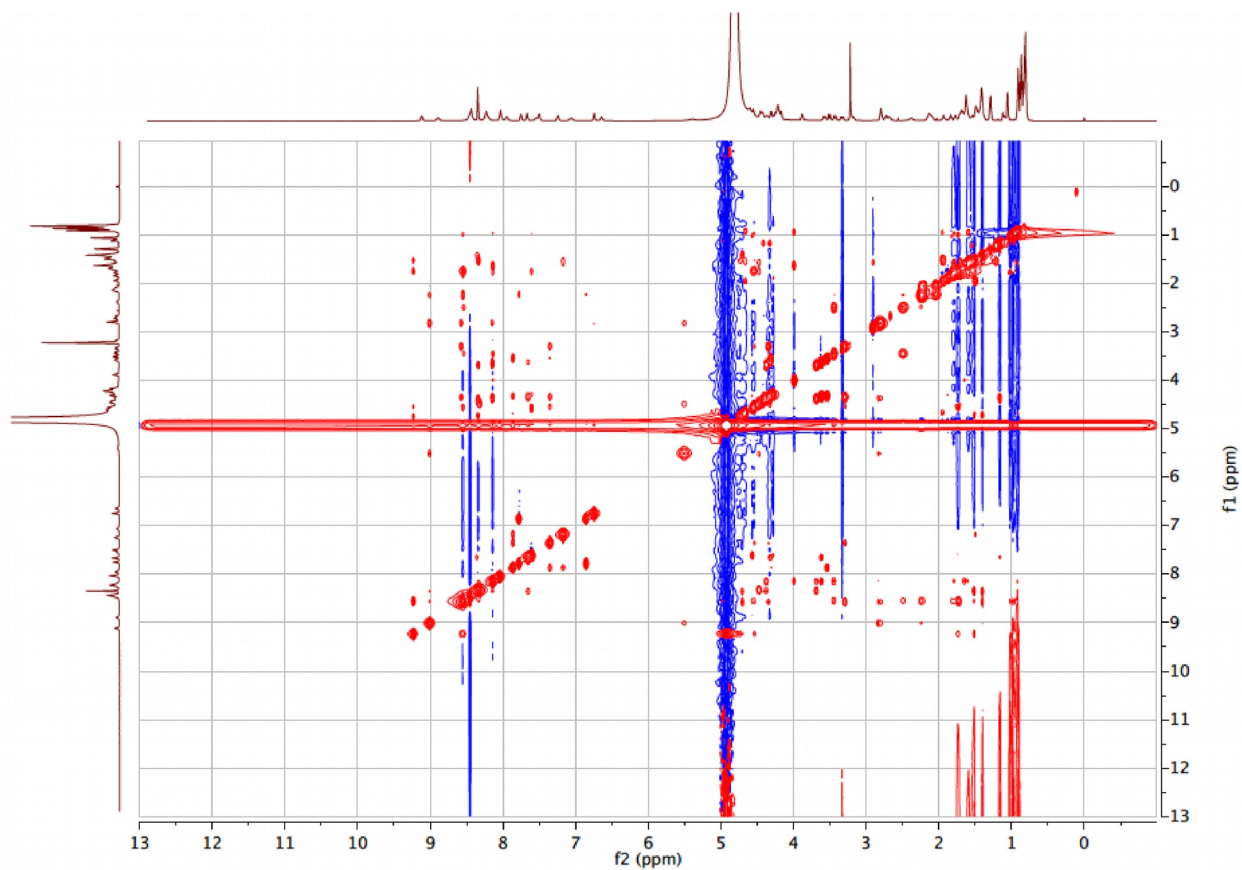


Fig. S39: ^1H - ^1H NOESY spectrum of deepstreptin in methanol- d_3 .

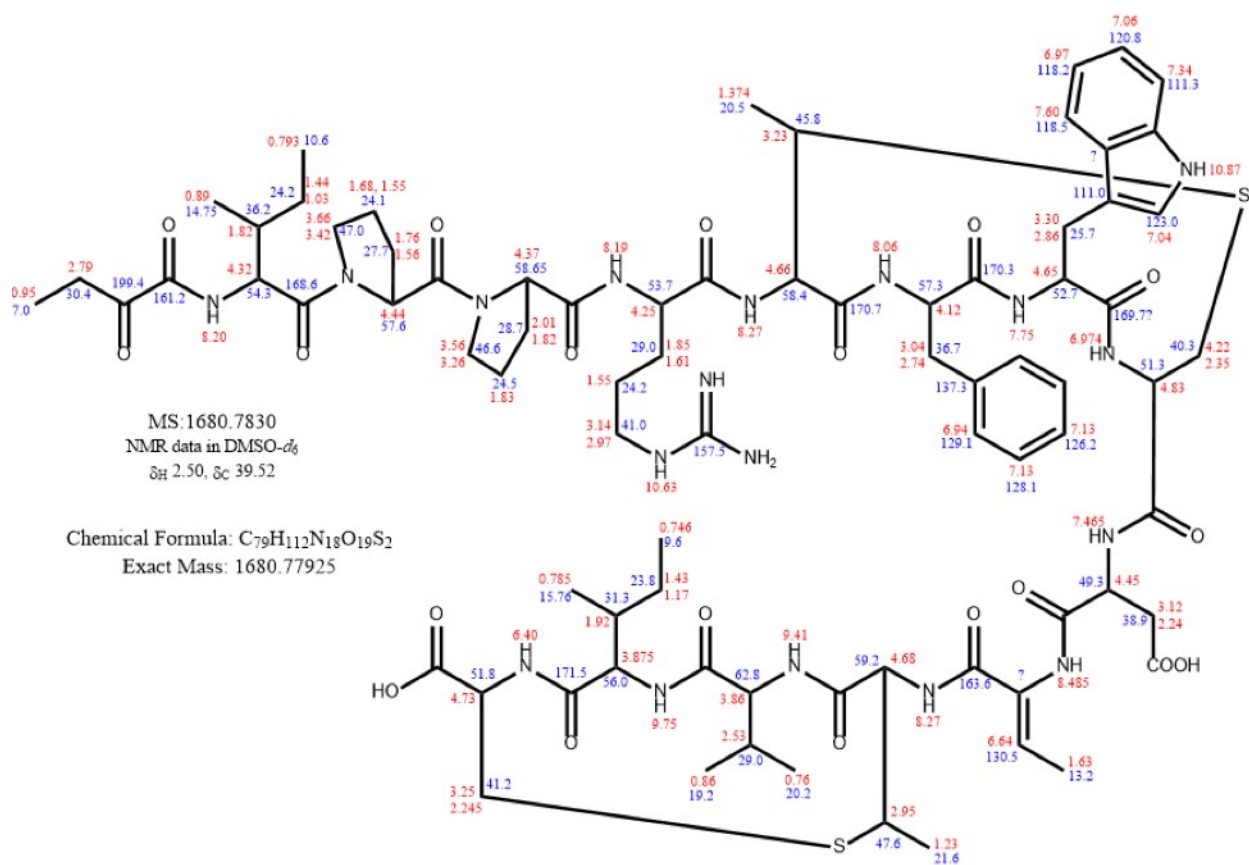


Fig. S40: Structure of deepflavo with assigned chemical shifts. Proton chemical shifts are shown in red while carbon chemical shifts are shown in blue.

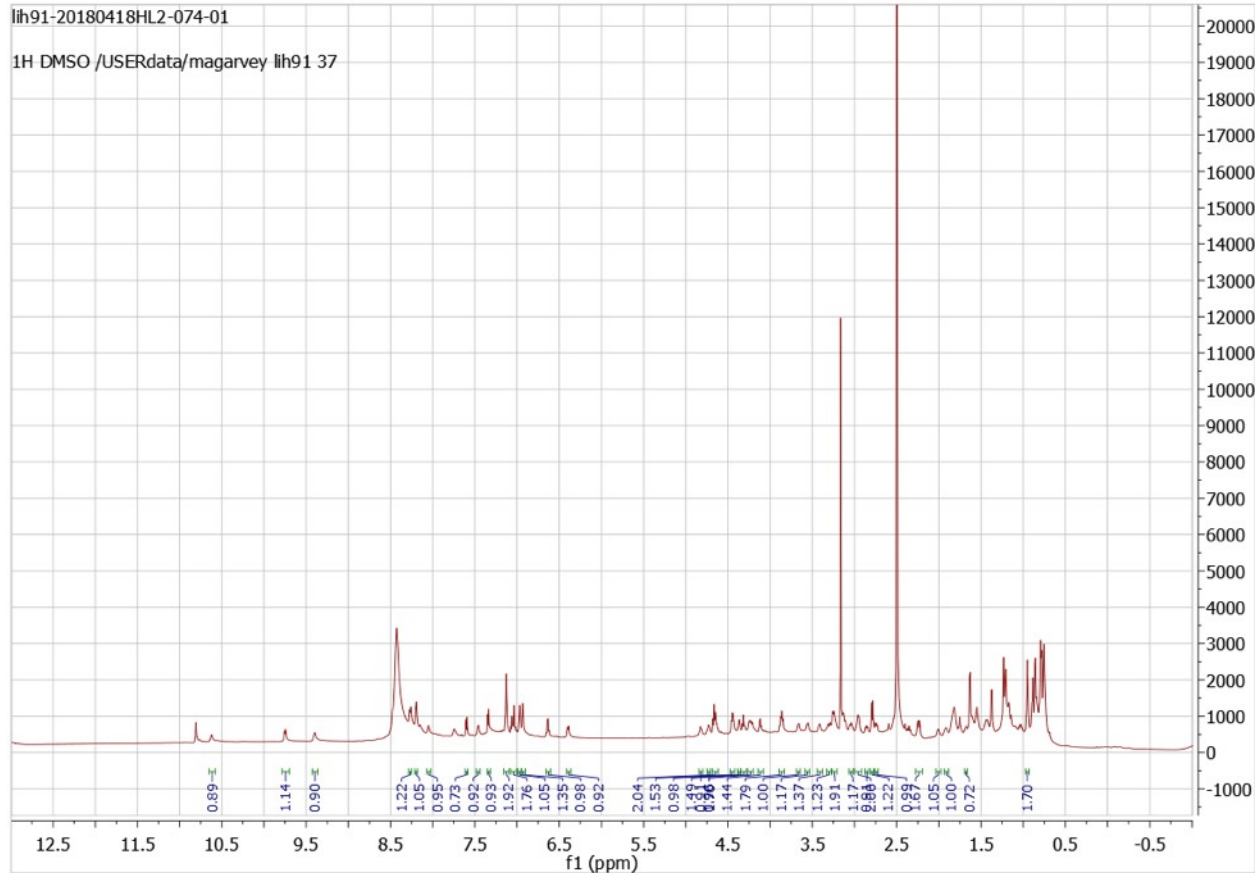


Fig. S41: ^1H -NMR spectrum of deepflavo in DMSO-d₆.

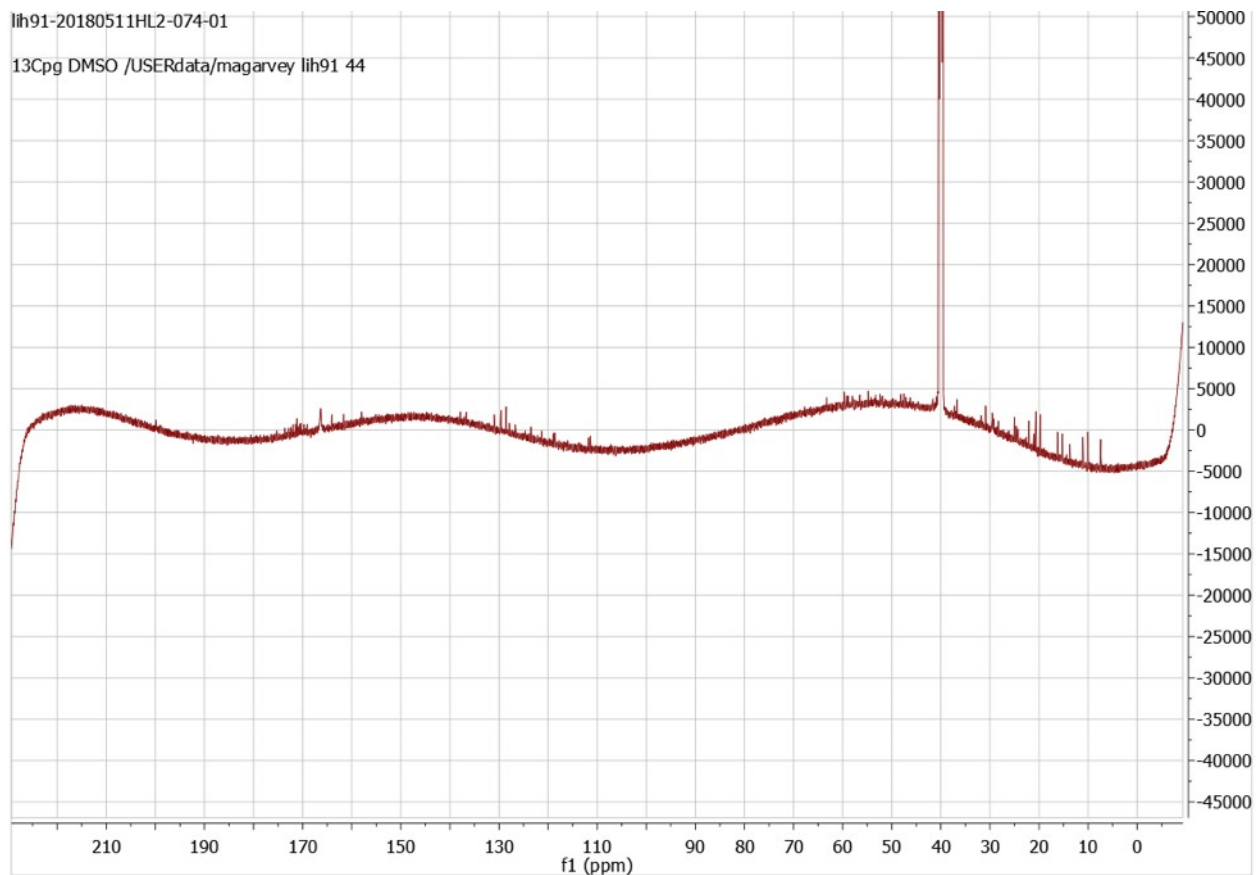


Fig. S42: ^{13}C -NMR spectrum of deepflavo in DMSO-d_6 .

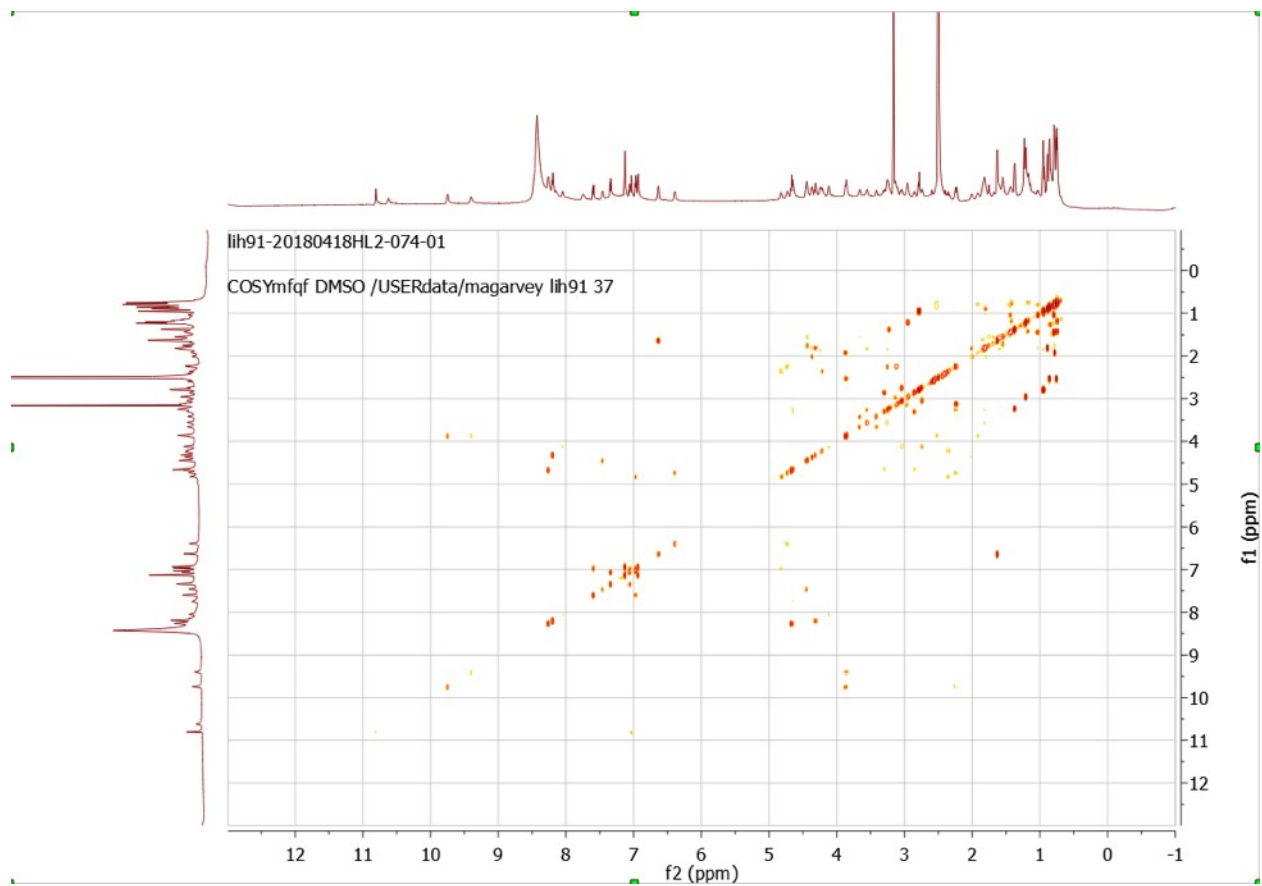


Fig. S43: ^1H COSY spectrum of deepflavo in DMSO-d₆.

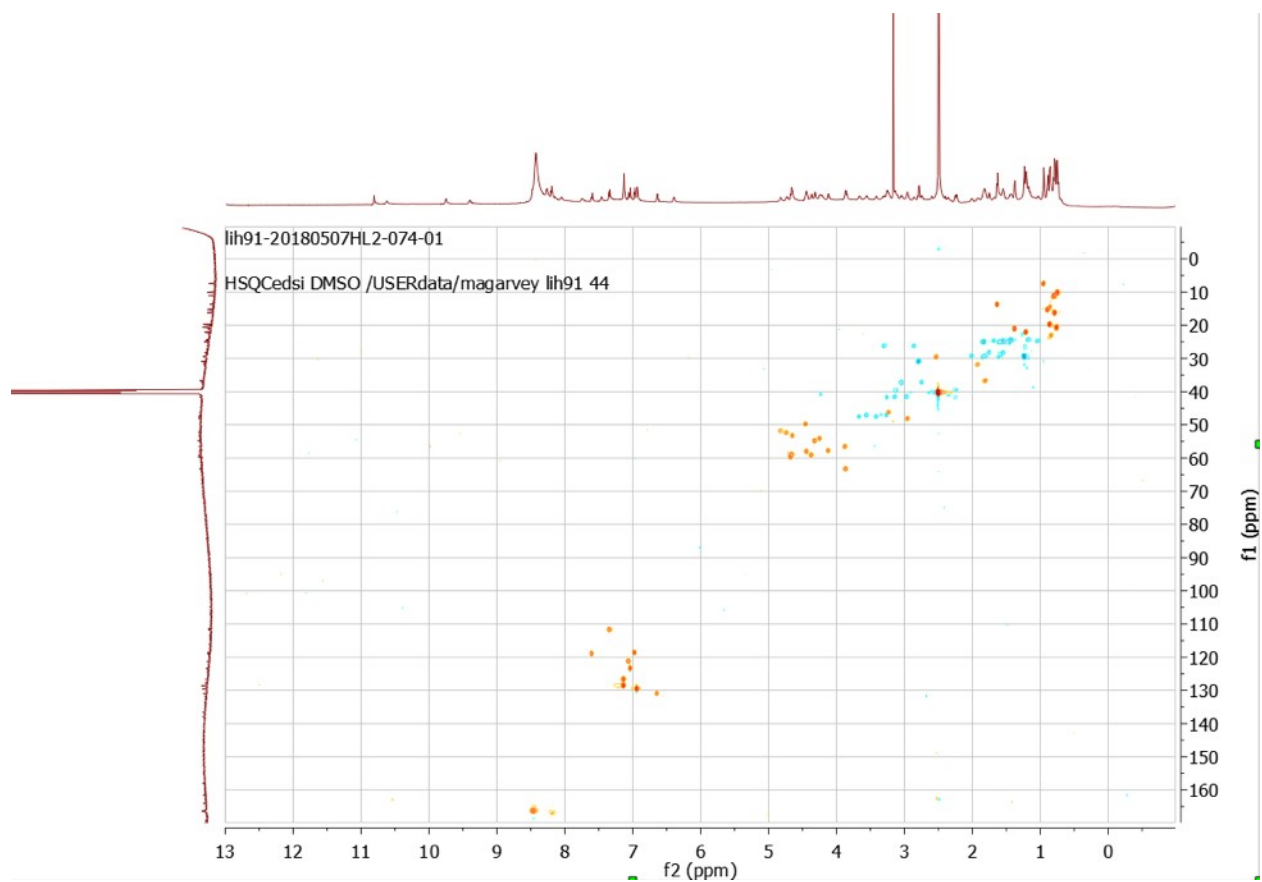


Fig. S44: ^1H - ^{13}C HSQC spectrum of deepflavo in DMSO- d_6 .

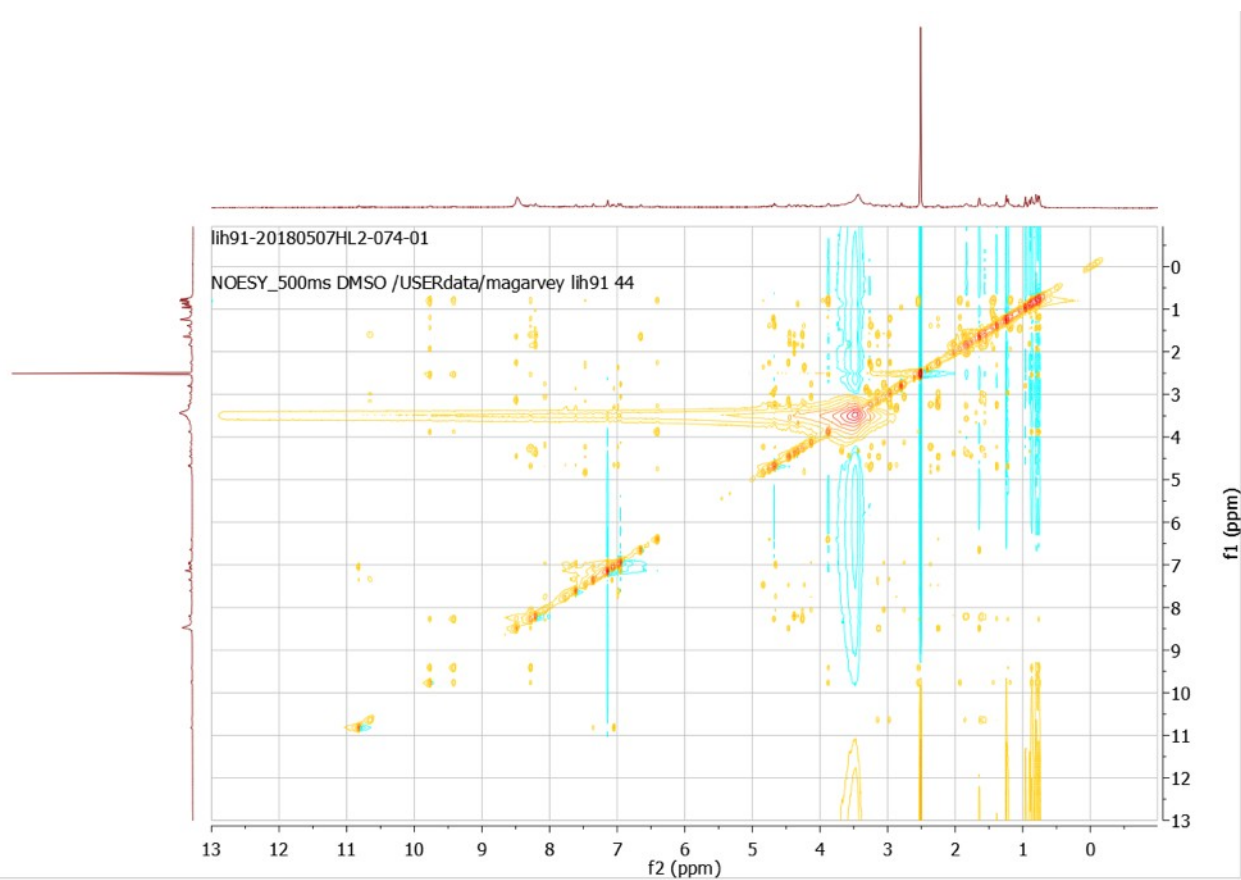


Fig. S45: ¹H-¹H NOESY spectrum of deepflavo in DMSO-d₆.

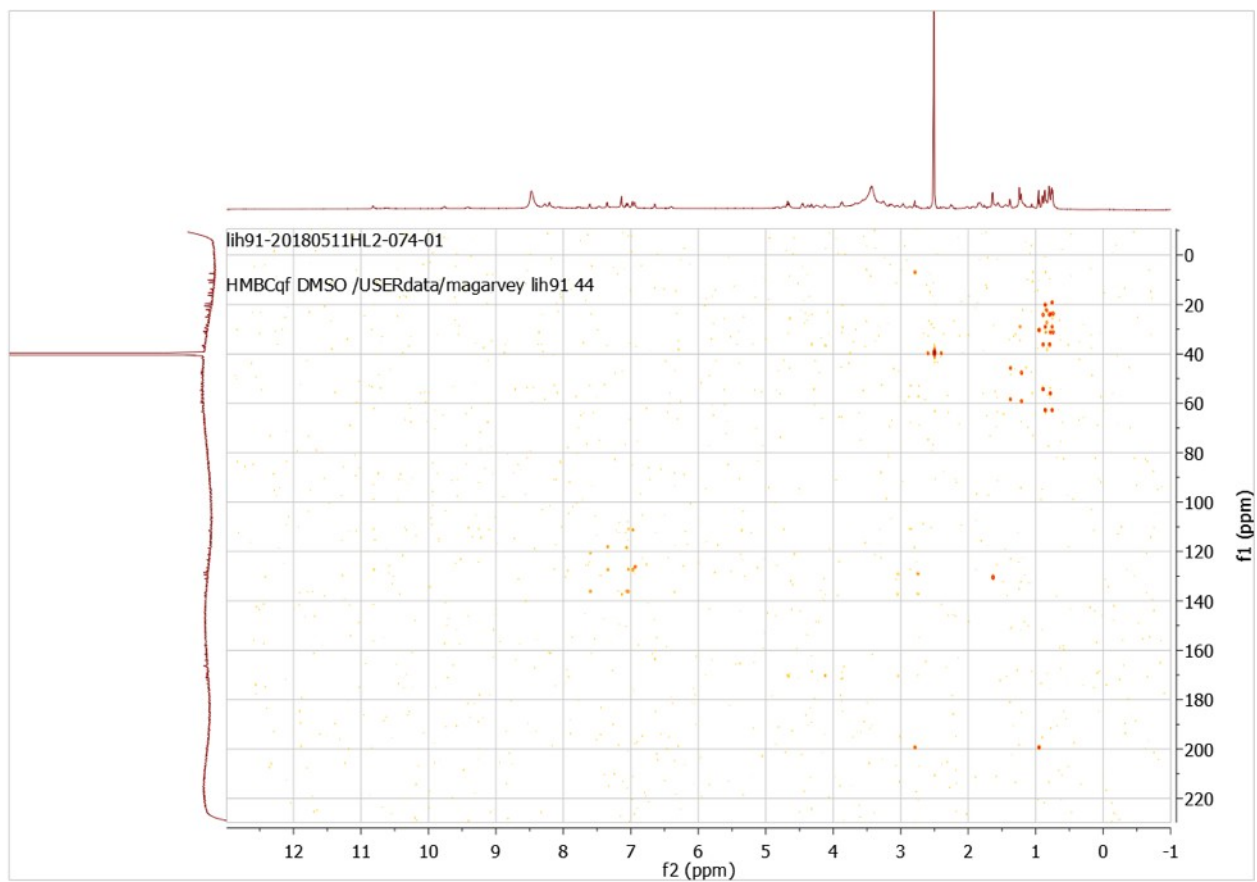


Fig. S46: ^1H - ^{13}C HMBC spectrum of deepflavo in DMSO- d_6 .

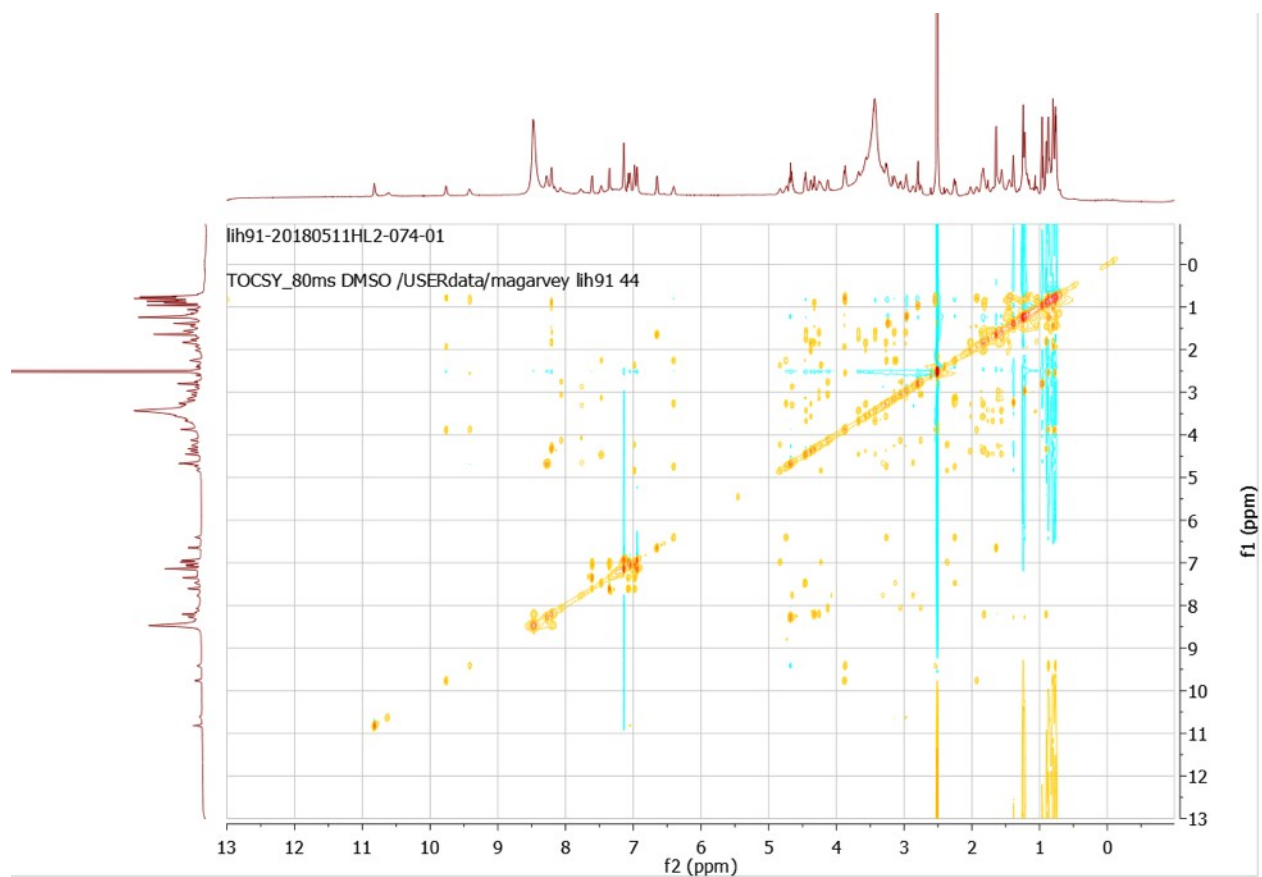


Fig. S47: ^1H - ^1H TOCSY spectrum of deepflavo in DMSO- d_6 .

MS: 1842.8162
 NMR data in DMSO- d_6
 d_4 2.50, d_2 39.52
 $C_{65}H_{113}N_{13}O_{22}S_2$
 Exact Mass: 1842.81095

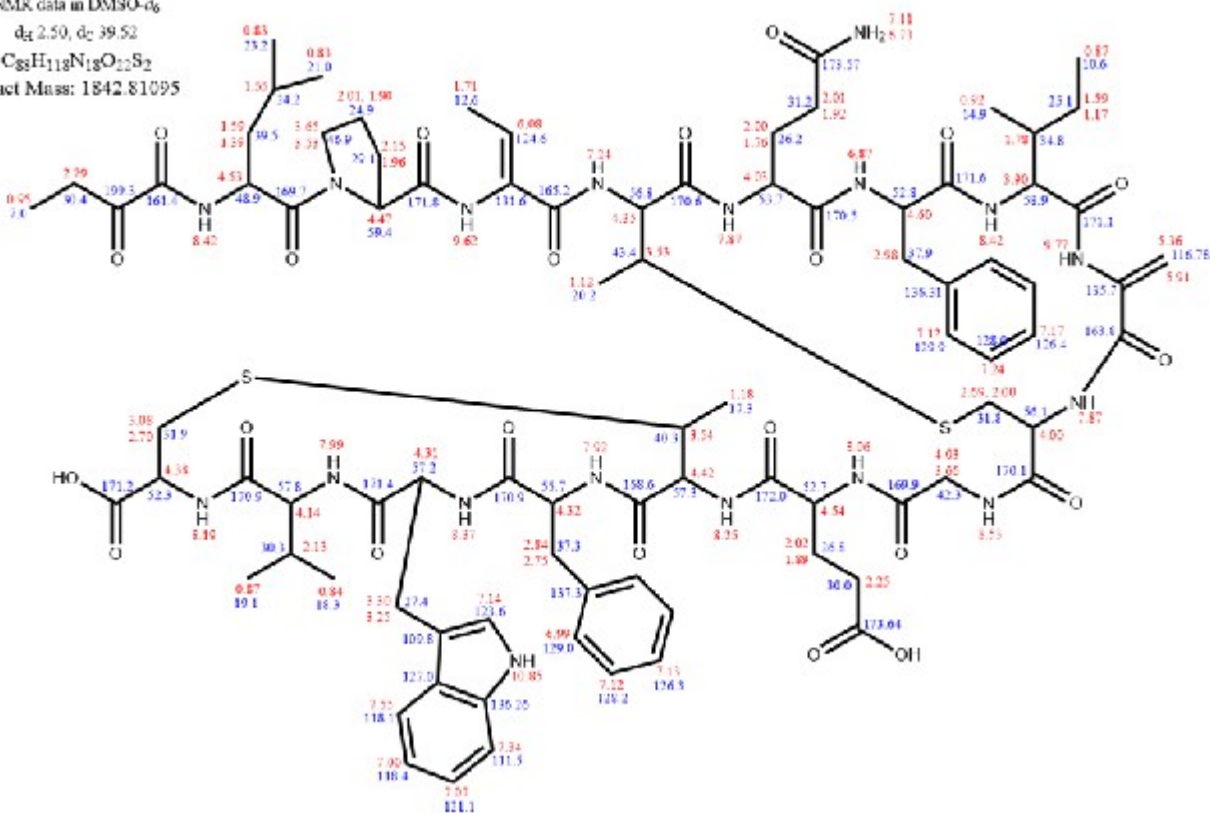


Fig. S48: Structure of deepginsen with assigned chemical shifts. Proton chemical shifts are shown in red while carbon chemical shifts are shown in blue.

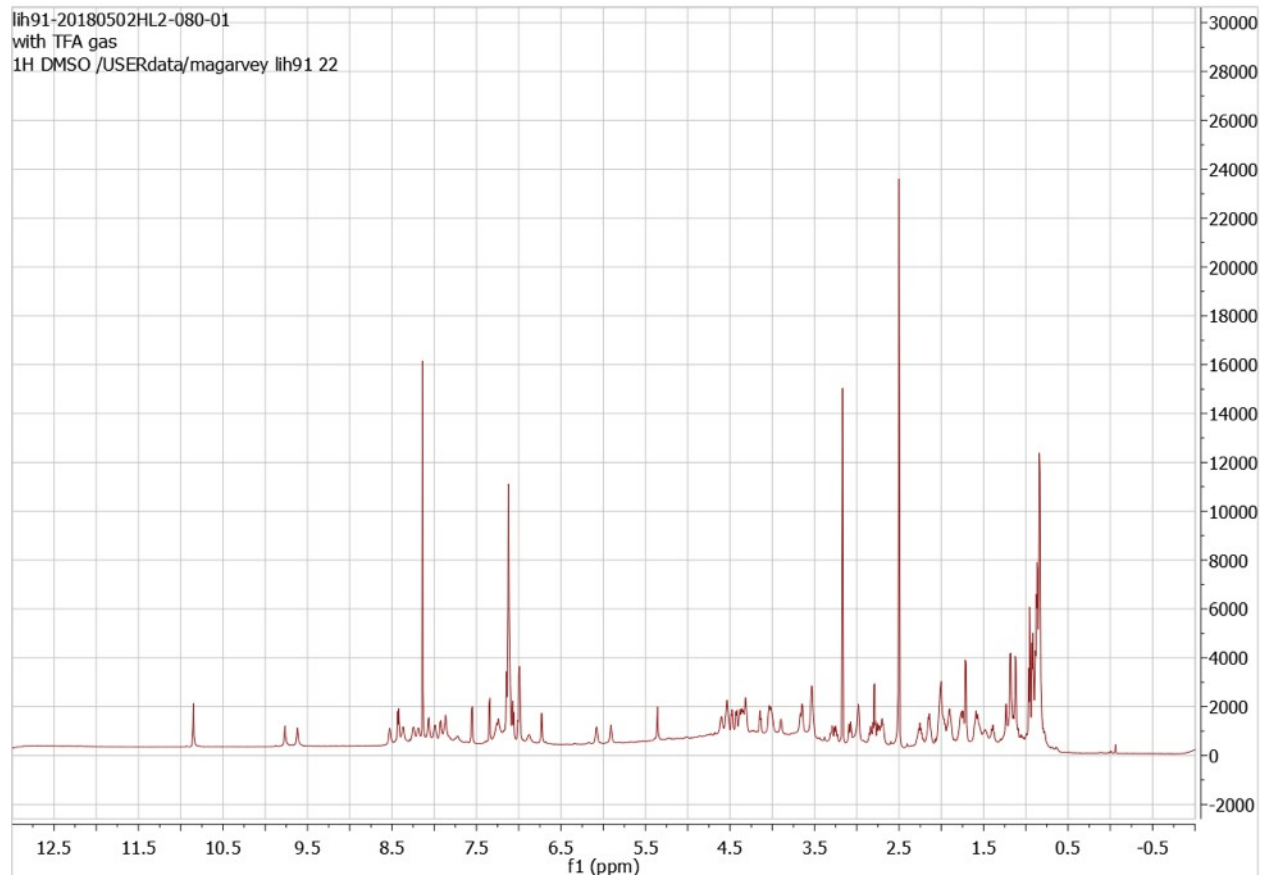


Fig. S49: ^1H -NMR spectrum of deepginsen in DMSO-d_6 .

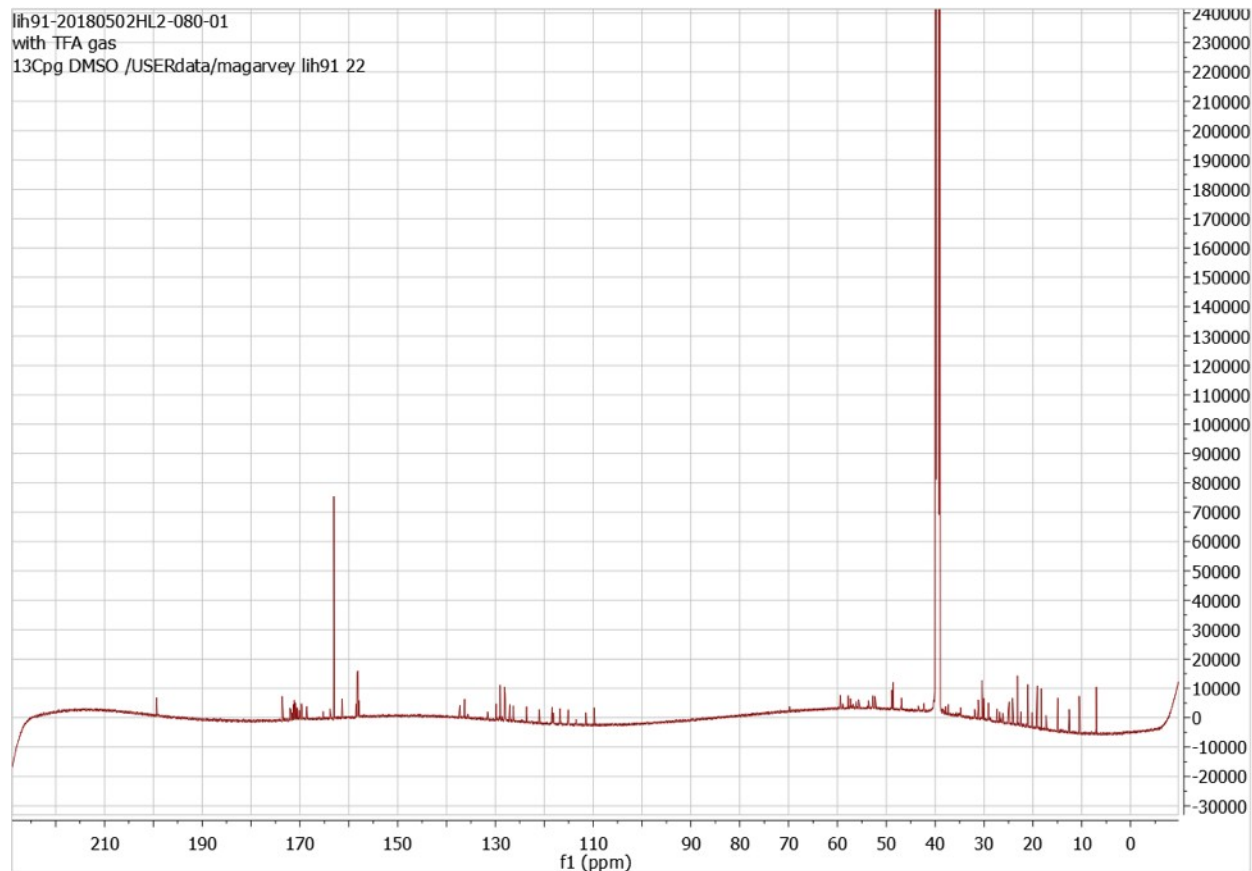


Fig S50: ^{13}C -NMR spectrum of deepginsen in DMSO-d₆.

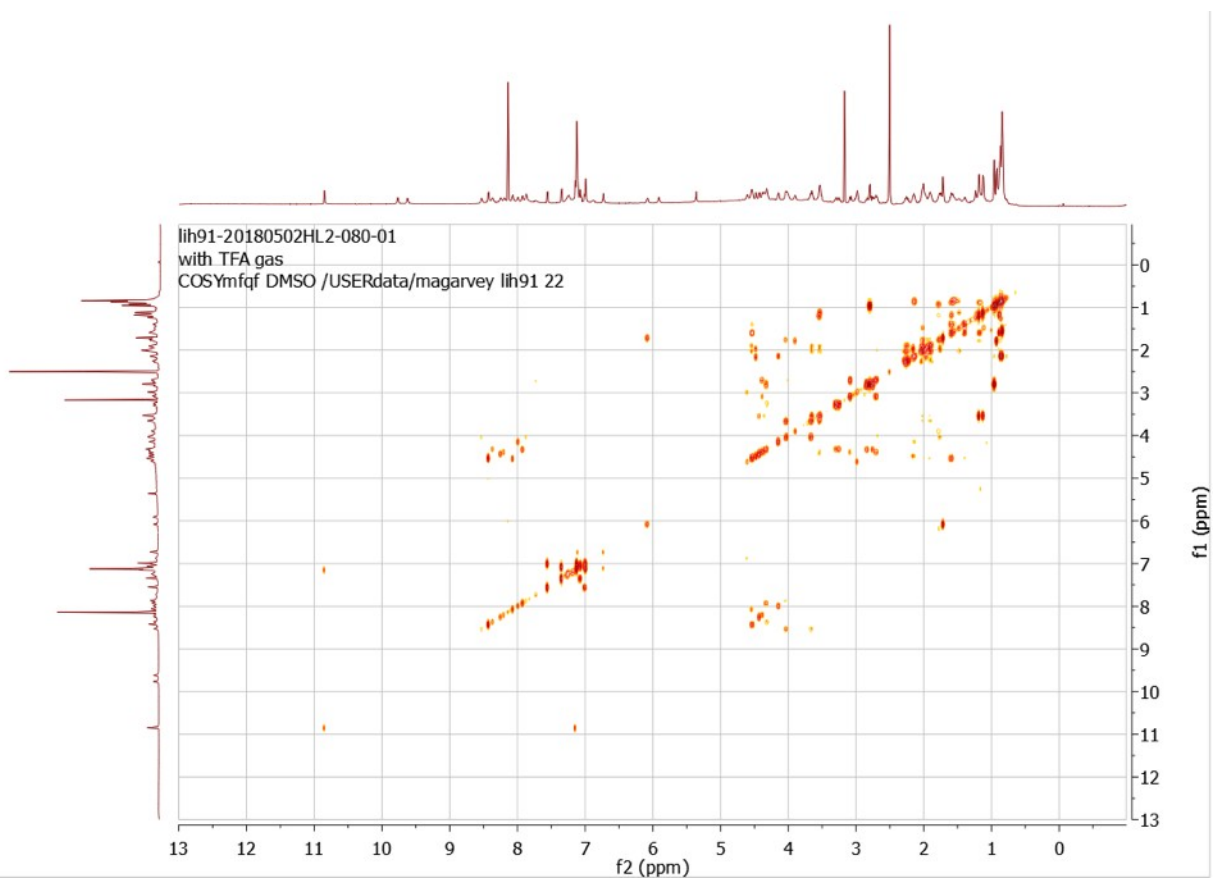


Fig. S51: ^1H - ^1H COSY spectrum of deepginsen in DMSO-d₆.

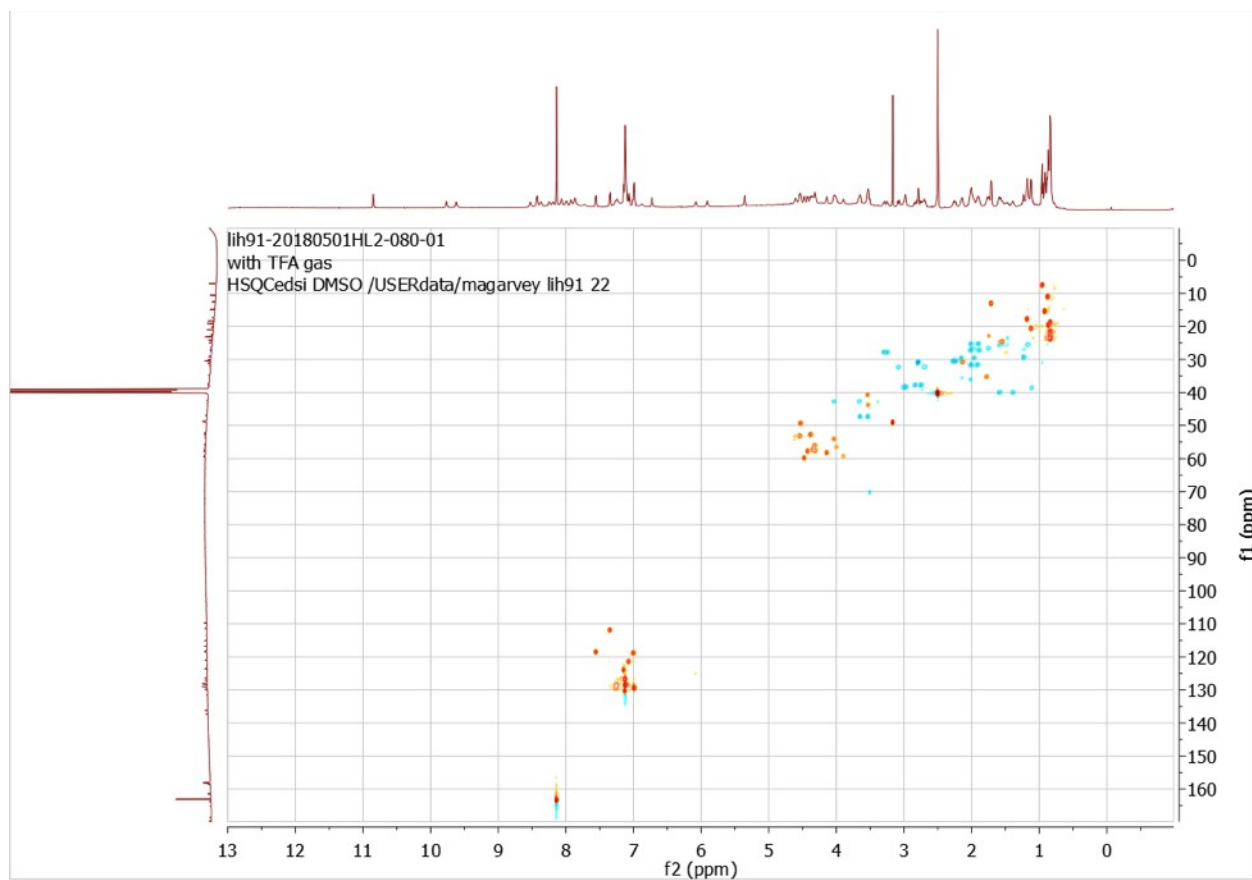


Fig. S52: ^1H - ^{13}C HSQC spectrum of deepginsen in DMSO- d_6 .

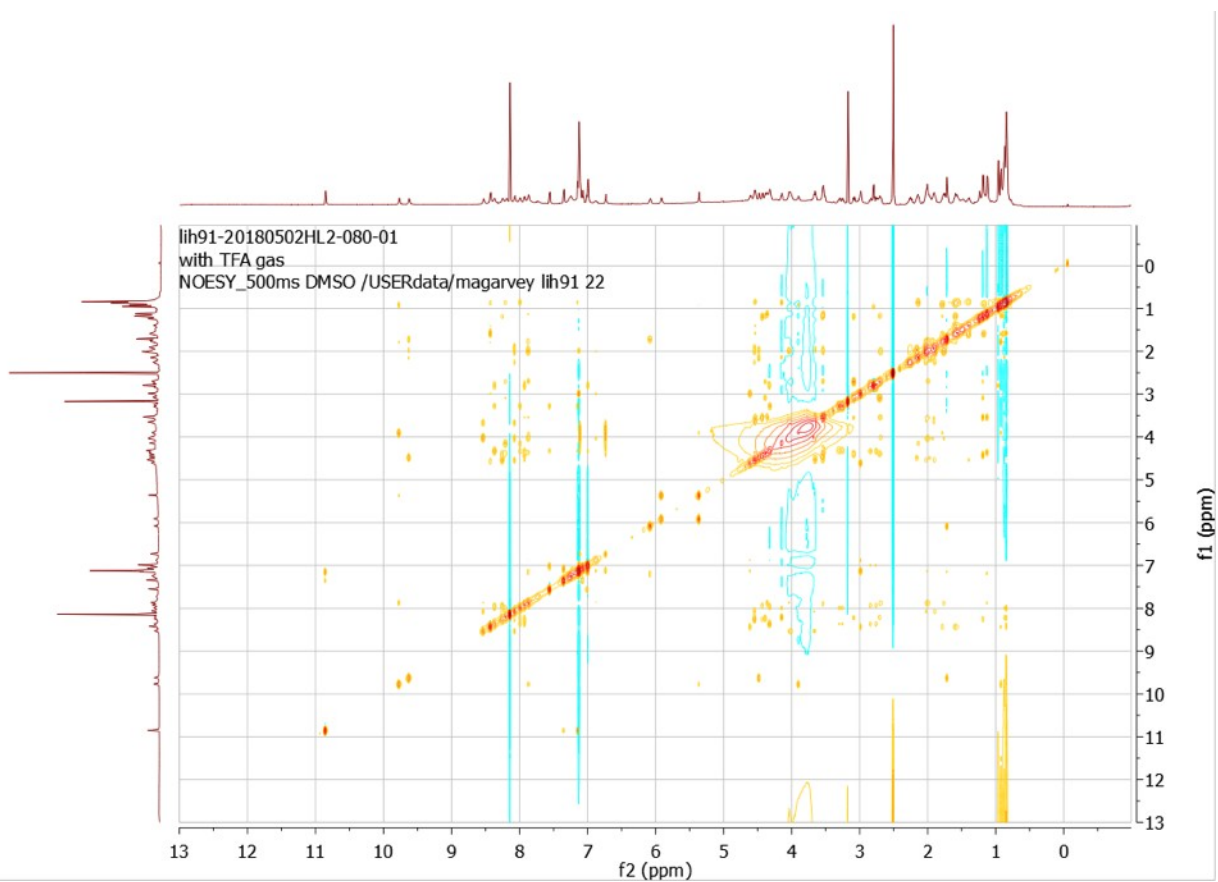


Fig. S53: ^1H - ^1H NOESY spectrum of deepginen in DMSO- d_6 .

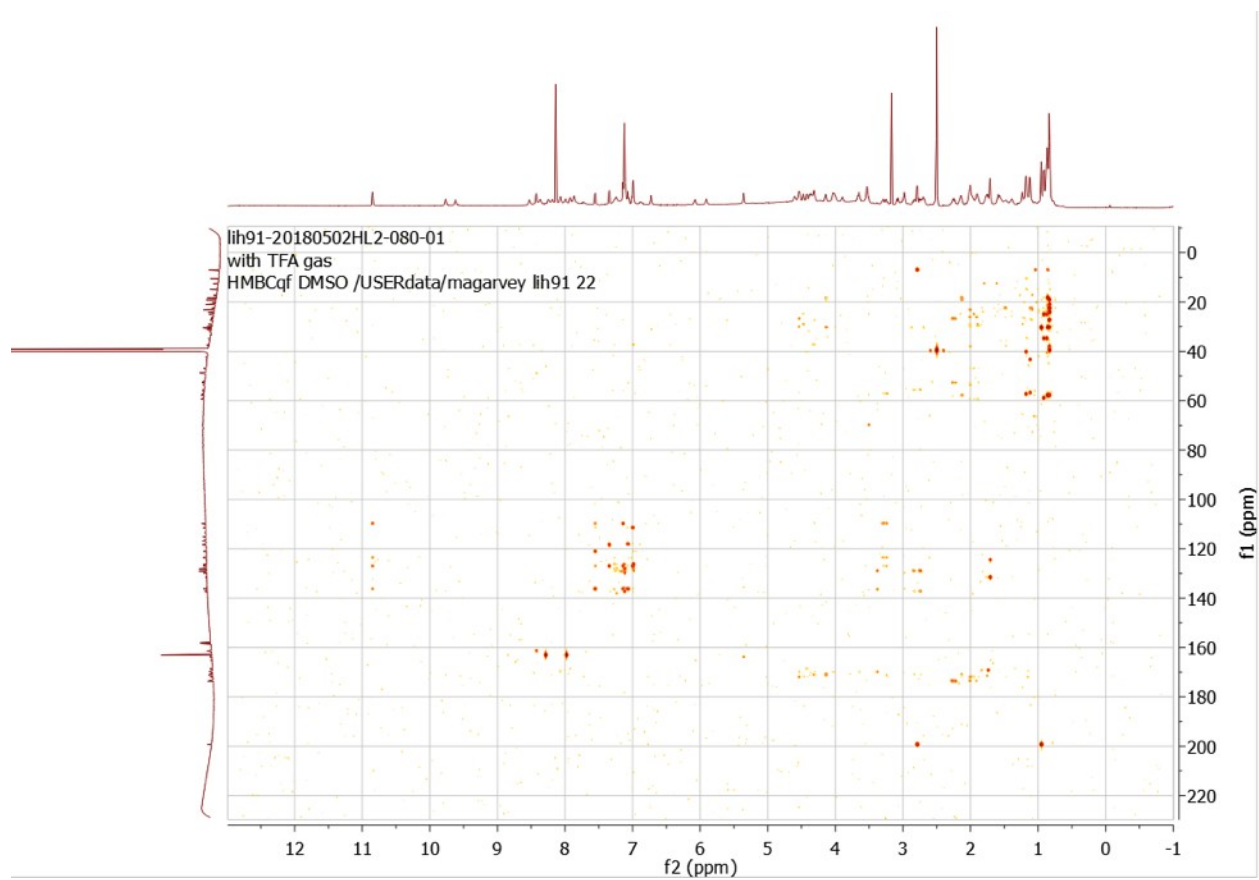


Fig. S54: ^1H - ^{13}C HMBC spectrum of deepginsen in DMSO- d_6 .

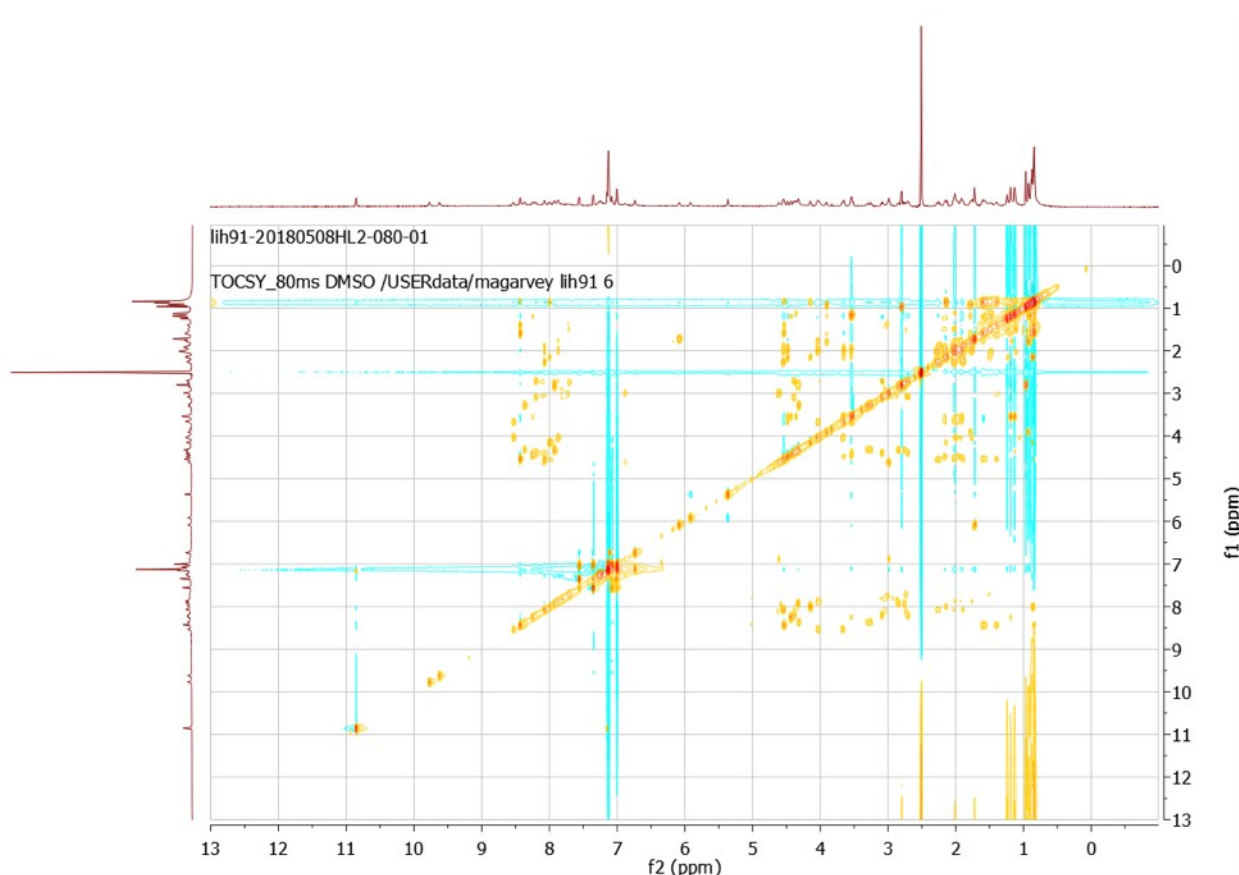


Fig. S55: ^1H - ^1H TOCSY spectrum of deepginsen in DMSO- d_6 .

References

1. Tietz JI, Schwalen CJ, Patel PS, Maxson T, Blair PM, Tai H-C, et al. A new genome-mining tool redefines the lasso peptide biosynthetic landscape. *Nat Chem Biol.* 2017;13: 470–478.
2. Agrawal P, Khater S, Gupta M, Sain N, Mohanty D. RiPPMiner: a bioinformatics resource for deciphering chemical structures of RiPPs based on prediction of cleavage and cross-links. *Nucleic Acids Res.* 2017;45: W80–W88.
3. Howard J, Ruder S. Universal Language Model Fine-tuning for Text Classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Stroudsburg, PA, USA: Association for Computational Linguistics; 2018. pp. 328–339.
4. Skinnider MA, Johnston CW, Edgar RE, Dejong CA, Merwin NJ, Rees PN, et al. Genomic charting of ribosomally synthesized natural product chemical space facilitates targeted mining. *Proc Natl Acad Sci USA.* 2016; 201609014.

5. Kingma DP, Ba J. Adam: A Method for Stochastic Optimization [Internet]. arXiv [cs.LG]. 2014. Available: <http://arxiv.org/abs/1412.6980>
6. McCallum A, Li W. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons. Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. Stroudsburg, PA, USA: Association for Computational Linguistics; 2003. pp. 188–191.
7. Dejong CA, Chen GM, Li H, Johnston CW, Edwards MR, Rees PN, et al. Polyketide and nonribosomal peptide retro-biosynthesis and global gene cluster matching. *Nat Chem Biol*. 2016;12: 1007–1014.
8. Ibrahim A, Yang L, Johnston C, Liu X, Ma B, Magarvey NA. Dereplicating nonribosomal peptides using an informatic search algorithm for natural products (iSNAP) discovery. *Proc Natl Acad Sci U S A*. 2012;109: 19196–19201.
9. Skinnider MA, Johnston CW, Zvanych R, Magarvey NA. Automated identification of depsipeptide natural products by an informatic search algorithm. *Chembiochem*. 2015;16: 223–227.
10. Navarro-Muñoz J, Selem-Mojica N, Mallowney M, Kautsar S, Tryon J, Parkinson E, et al. A computational framework for systematic exploration of biosynthetic diversity from large-scale genomic data [Internet]. bioRxiv. 2018. p. 445270. doi:10.1101/445270
11. Myers L, Sirois MJ. Spearman Correlation Coefficients, Differences between [Internet]. Encyclopedia of Statistical Sciences. 2004. doi:10.1002/0471667196.ess5050
12. Zar JH. Biostatistical Analysis: Pearson New International Edition. Pearson Higher Ed; 2013.
13. Fieller EC, Pearson ES. Tests for Rank Correlation Coefficients: II [Internet]. *Biometrika*. 1961. p. 29. doi:10.2307/2333127
14. Skinnider MA, Dejong CA, Franczak BC, McNicholas PD, Magarvey NA. Comparative analysis of chemical similarity methods for modular natural products with a hypothetical structure enumeration algorithm. *J Cheminform*. 2017;9: 46.
15. Kuroda M. A novel descriptor based on atom-pair properties. *J Cheminform*. 2017;9: 1.
16. Rogers D, Hahn M. Extended-Connectivity Fingerprints [Internet]. *Journal of Chemical Information and Modeling*. 2010. pp. 742–754. doi:10.1021/ci100050t
17. Durant JL, Leland BA, Henry DR, Nourse JG. Reoptimization of MDL keys for use in drug discovery. *J Chem Inf Comput Sci*. 2002;42: 1273–1280.
18. Landrum G. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling [Internet]. rdkit.org; 2013. Available: http://www.rdkit.org/RDKit_Overview.pdf
19. Nilakantan R, Bauman N, Dixon JS, Venkataraghavan R. Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *J Chem Inf Comput Sci*. 1987;27: 82–85.

20. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. *J Mach Learn Res.* 2011;12: 2825–2830.