Electronic Supplementary Material (ESI) for Digital Discovery.

Electrical Supplementary Information

A deep learning model for type II polyketide natural product prediction without sequence alignment

Jiaquan Huang[1,3], Qiandi Gao[1,3], Ying Tang[2], Yaxin Wu[1], Heqian Zhang[1*], Zhiwei Qin[1*]

[1]Center for Biological Science and Technology, Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, Guangdong, 519087, China.

[2]International Academic Center of Complex Systems, Advanced Institute of Natural Sciences, Beijing Normal University, Zhuhai, Guangdong, 519087, China.

[3]These authors contributed equally to this work

Correspondence

Prof. Zhiwei Qin, Email: z.qin@bnu.edu.cn

Dr Heqian Zhang, Email: zhangheqian@bnu.edu.cn

Materials and methods

Protein sequence data preparation and embedding using protein language models

A dataset comprising 163 KS_β sequences with known chemical structures was retrieved from the NCBI database. These sequences were designated as 'labeled'. In contrast, KS_β sequences without corresponding chemical structures were classified as 'unlabeled'. To further curate the unlabeled KS_β, the 163 labeled KS_α and KS_β sequences were used to obtain 20 kb putative T2PK minimum BGCs following the in-house pipeline described by Chen *et al.*[1]. For the criterion of a reliable T2PK gene cluster, KS_α and KS_β sequences should be identified in the same contig. This process yielded 2566 KS_β sequences without corresponding chemical structures. Additionally, the non-KS_β sequences from actinobacteria were obtained from UniRef50, with a focus on sequences between 300 and 500 amino acids in length. To eliminate irrelevant entries, we excluded any sequence matching the keywords 'beta ketoacyl synthase' or 'chain length factor'. Taken together, our dataset consisted of 2729 (163 + 2566) KS_β sequences and 761,302 non-KS_β sequences, which were subjected to subsequent analysis. All sequences were confirmed to non-redundant using CD-HIT with 100% identity threshold.

As protein sequences should be represented with numerical vectors before being input to the classification algorithm, we employed five general protein language models (PLMs) to embed the amino acids of each sequence, resulting in a representation vector obtained by averaging the embeddings across the entire sequence. We also aimed to investigate which PLMs produced high-quality learned representations for distinguishing functions, as well as whether these representations were associated with the given labels. To achieve this, we used the UMAP dimension-reduction algorithm to visualize the high-dimensional embeddings. Our results demonstrated that the ESM-2 model, with 3 billion parameters, outperformed the other models in terms of learned representation quality. This finding suggests that ESM-2 may be particularly useful for protein sequence classification tasks.

Binary KS_β classifier development

To construct a robust binary classification model, we utilized an 80/10/10 split of the KS_β and non-KS_β datasets to produce a training, validation and test set, respectively. We trained four classifiers, namely, random forest, XGBoost, support vector machine (SVM) using the scikit-learn Python module (version 1.2) and multilayer perceptron (MLP) using the PyTorch module (version 2.0). The selected hyperparameters for each classifier are documented in Table S8. For the MLP classifier, a dropout layer with a value of 0.5 was incorporated into the network, and a focal loss function was utilized to handle the highly imbalanced positive and negative datasets. To evaluate the performance of the

models, a 5-fold cross-validation technique was implemented, and the accuracy value of each fold was averaged. Due to the imbalanced nature of the two datasets, only accuracy, confusion matrix, and F1 score were employed as evaluation metrics for the final model performance. The scikit-learn Python module was used to compute all these metrics.

Relabeling 163 KS_β protein sequences

Initially, 163 KS_β sequences were categorized into 5 classes based on the number of building blocks for the main carbon skeleton. Next, we developed a pipeline for tuning hyperparameters in clustering using HDBSCAN[2] and UMAP[3] to refine and expand the predefined 5 classes labels. Our pipeline was guided by the chatintents python packages (https://github.com/dborrelli/chat-intents), but we replaced unsupervised UMAP with supervised UMAP. This pipeline comprised four major functions: *generate_clusters*, *score_clusters*, *objective* and *bayesian_search*. Briefly, the main function *bayesian_search* takes in a dataset, hyperparameter space and other parameters. It deploys the 'trials' object within the *bayesian_search* function to monitor the results of each evaluation of the *objective* function. The *objective* function, on the other hand, receives hyperparameters from the space, applies the *generate_clusters* function to create a clustering object, and computes the number of clusters and a cost metric via the *score_clusters* function. It also imposes a penalty on the cost if the number of clusters falls outside the desired range. The Bayesian optimization process is conducted using the *fmin* function from the *hyperopt* package, with the *tpe.suggest* algorithm eventually yielding the optimal hyperparameters found, the resulting clustering object, and the trials object for further scrutiny. The hyperparameter space encompasses diverse parameters, including *n_neighbors* and *min_dist* of UMAP and *cluster_selection_epsilon* of HDBSCAN. The detailed range of hyperparameter spaces is supplemented in Table S9. Overall, the pipeline was designed to optimize the creation of new local clusters from global clusters, utilizing supervised UMAP and HDBSCAN with the aid of label penalty and Bayesian optimization techniques. By following this pipeline, we are able to generate new class labels based on the previous class labels while simultaneously improving their consistency with the protein structure information that is embedded in the data.

T2PK classifier architecture and training in different label types

In the present investigation, a total of 163 KS_β sequences, each with known chemical structures, were assigned five manually annotated and nine autogenerated class labels. Owing to the significant imbalance in the labeled dataset, the dataset was only divided into training and test sets following an 80/20 ratio. Notably, random partitioning of data is typically avoided in biological sequence modeling due to its tendency to yield an overly simplistic evaluation of generalization. Thus, sticking to the

80/20 ratio, we conducted manual partitioning of the dataset. Detailed information is provided in Table S4. To assess the quality of the assigned labels and to determine the most suitable multiclass classification algorithm, four multiclass classifiers were trained using the scikit-learn and Pytorch modules. Hyperparameter optimization was performed using grid searches, and model performance was assessed by the metrics mentioned above.

To enhance the robustness of the initial MLP classifier, we employed a consistency regularization-based semisupervised framework, as proposed by Laine et al[4]. The loss function comprised two distinct components: the standard cross-entropy loss, which exclusively evaluated the labeled inputs, and a second component that evaluated all inputs (both labeled and unlabeled). This second component penalized divergent predictions for the same training input, measured by the mean square difference between the prediction vectors and perturbed vectors. To augment the unlabeled data, Gaussian noise was added to the embeddings by specifying a standard deviation for the Gaussian distribution. The optimal amount of noise was determined based on its effect on the training accuracy. The resulting neural network classifier was then trained using a weighted total loss function, consisting of both the consistency loss and the standard cross-entropy loss. The detailed hyperparameter spaces mentioned above have been summarized in Table S8.

Detection of T2PKs with potentially new skeletons

Softmax-based classifiers are known to generate overconfident posterior distributions when presented with out-of-distribution detection (ODD) data. To address this issue, a generative classifier, specifically Gaussian discriminant analysis, and a Mahalanobis distance-based framework were utilized to obtain confidence scores, as described by Lee et al[5]. To evaluate the performance of this approach, 163 KS_{α} sequences were designated ODD data, and 163 KS_{β} sequences were designated in-distribution (ID) data. The feature vector for each layer was extracted from the neural network, and a generative classifier was applied to obtain the mean and covariance matrix for each class. The Mahalanobis distance-based scores were then calculated between a test sample and the closest class Gaussian to obtain confidence scores for both datasets. To identify the feature layer that was most suitable for distinguishing between ID and ODD data, all layers were evaluated using a one-class SVM on both datasets. The layer that exhibited the best performance on both datasets was selected. The isolation forest, a general abnormal data detection algorithm, was then used to detect novelty data from an unlabeled dataset. The protein structure of the detected KS_{β} novelty data was predicted using ESMFold[6] *in silico*, and the difference in protein structures was determined using the root-mean-square deviation.

General microbiology and chemistry experiments

Total DNA of five *Streptomyces* strains was used for experimental confirmation, and their genomes were extracted using a custom genomic extraction protocol. The genomes were sequenced using a one-dimensional MinION flow cell with an r9.4.1 flow cell from Oxford Nanopore Technologies, UK, and base-calling was performed with Guppy v6.2.1. Additionally, some DNA samples were sequenced using MiSeq sequencing from Illumina, CA, USA. Quality control of the long reads (LRs) and short reads (SRs) was conducted using fastqc v0.11.6[7], with quality control performed using different software suites: Porechop v0.2.4[8] and Filtlong v0.2.0 (https://github.com/rrwick/Filtlong) for LRs and FASTX-Toolkit for SRs. A hybrid assembly using Unicycler v0.5.0[9] was then performed by combining LRs and SRs. For samples with LRs only, the assembly was carried out using canu 2.1[10] and polished with racon 1.0[11]. The protein sequences of all genomes were predicted using prokka v1.14.6[12]. The BGCs of the six genomes were predicted using DeepBGC[13] and antiSMASH[14]. The performance of DeepT2, DeepBGC and antiSMASH in predicting T2PKs was compared.

Unless stated otherwise, all chemicals were supplied by Macklin. All solvents were of HPLC grade or equivalent. Actinomycetes were cultivated at 30 °C on ISP2 agar. Crude metabolites were extracted by ethyl acetate. Samples were analyzed by LCMS/MS on an Agilent G6500 UHPLC system attached to a quadrupole time-of-flight (Q-ToF) mass spectrometer. The spray chamber conditions were as follows: nebulizer, 5 L/min; drying gas, 200; sheath gas temperature, 350 °C; sheath gas flow, 11 L/min; and drying gas on, 5 L/min. The instrument was calibrated using an API-TOF Reference Mass Solution Kit according to the manufacturer's instructions. The following analytical LCMS method was used throughout this study: Phenomenex Kinetex C18 column (100 × 2.1 mm, 100 Å); mobile phase A: water + 0.1% formic acid; mobile phase B: acetonitrile + 0.1% formic acid. Elution gradient: 0-1 min, 20% B; 1-12 min, 20%-100% B; 12-14 min, 100% B; 14-14.1 min, 100%-20% B; 14.1-17 min, 20% B; flow rate: 0.3 mL/min; injection volume: 10 µL. HPLC was performed on an Agilent 1290 system, and the following method was used throughout this study: Phenomenex Kinetex C18 column (100 × 2.1 mm, 100 Å); mobile phase A: water + 0.1% formic acid; mobile phase B: acetonitrile + 0.1% formic acid. Elution gradient: 0-1 min, 20% B; 1-12 min, 20%-100% B; 12-14 min, 100% B; 14-14.1 min, 100%-20% B; 14.1-17 min, 20% B; flow rate: 0.3 mL/min; injection volume: 10 µL. UV: 250 and 415 nm.

Molecular networking description

A molecular network was created using the online workflow on the GNPS website (http://gnps.ucsd.edu)[15]. The data were filtered by removing all MS/MS fragment ions within +/- 17

Da of the precursor m/z. MS/MS spectra were window filtered by choosing only the top 6 fragment ions in the +/- 50 Da window throughout the spectrum. The precursor ion mass tolerance was set to 1 Da with an MS/MS fragment ion tolerance of 0.5 Da. A network was then created where edges were filtered to have a cosine score above 0.2 and more than 2 matched peaks. Furthermore, edges between two nodes were kept in the network if and only if each of the nodes appeared in each other's respective top 10 most similar nodes. Finally, the maximum size of a molecular family was set to 100, and the lowest scoring edges were removed from molecular families until the molecular family size was below this threshold.

Supplementary Figures

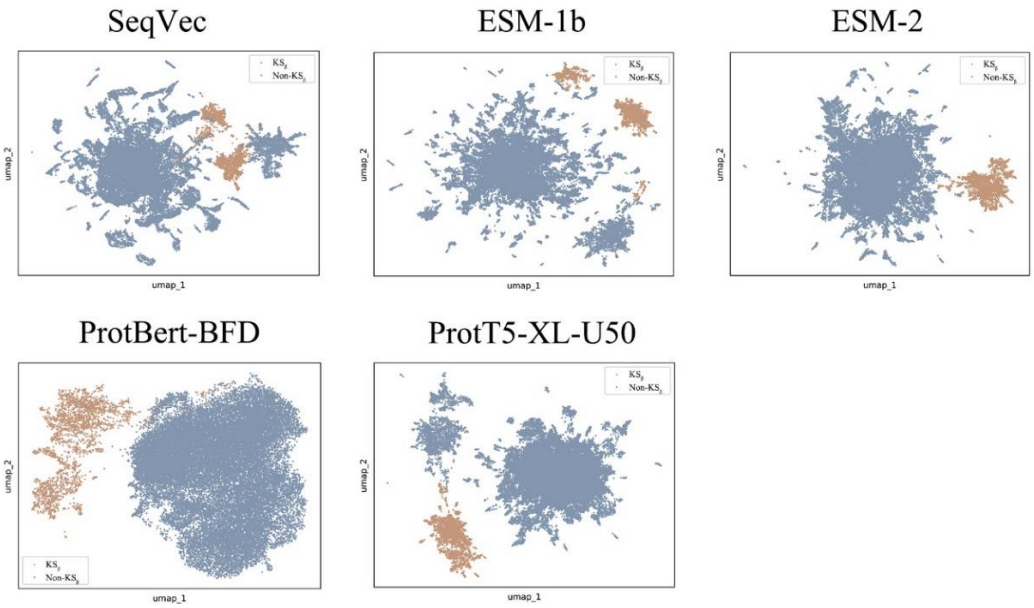

Figure S1. Dimensional-reduction representations of KS_β and non- KS_β embeddings are encoded in five general protein language models. KS_β: 2,729; non- KS_β: Random selection of 36,089 from 761,302.

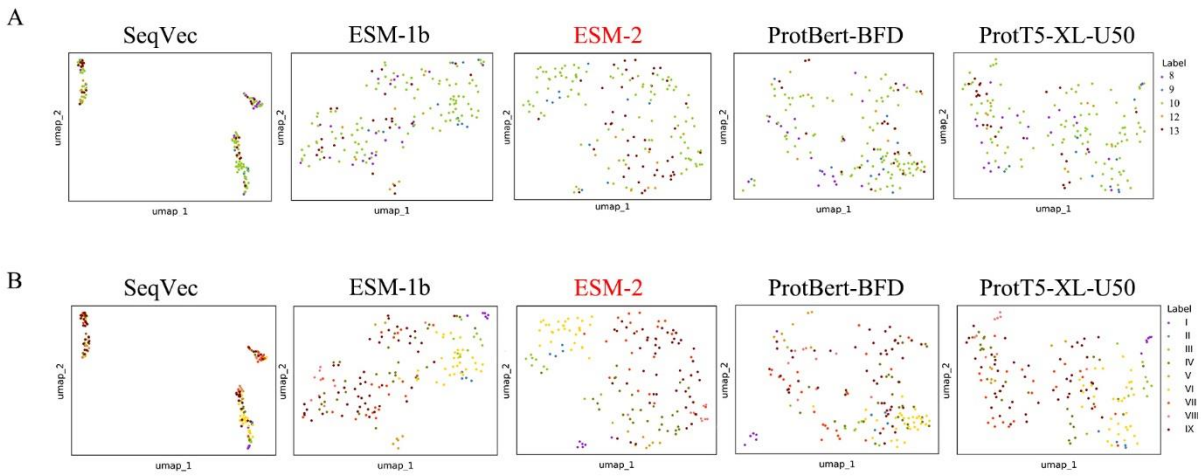

Figure S2. Dimensional-reduction representations of 163 KS_{β} embeddings with two types of class label are encoded in five general protein language models. Panel A: Five class labels according to the building block number of their corresponding to T2PK main skeleton, namely, 8, 9, 10, 12 and 13; Panel B: Nine class labels derived from five class labels using constrained optimization approach. These dimensional-reduction representations were generated through an unsupervised UMAP algorithm.

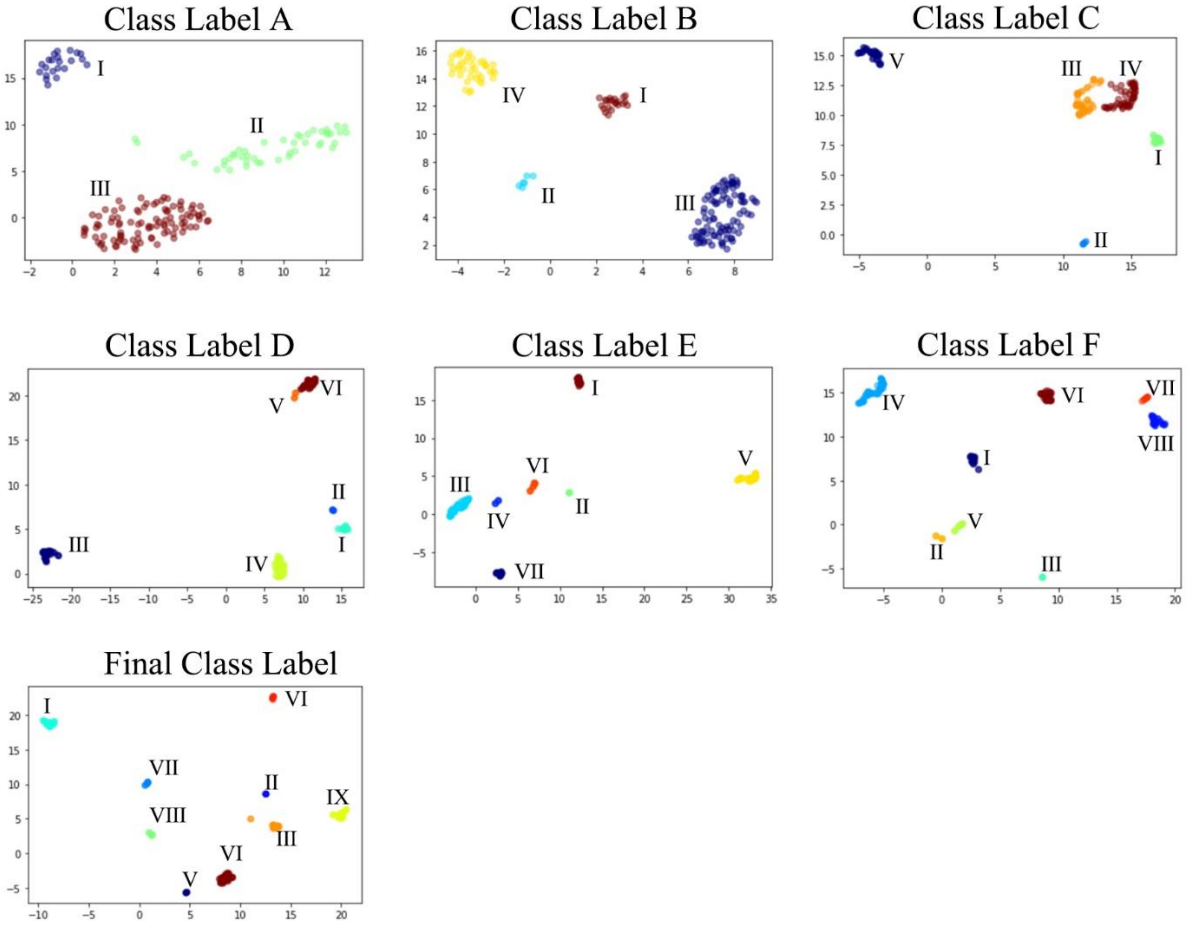

Figure S3. Two-dimensional representations showing the process of class labeling for T2PKs. Details of the hyperparameters used for cluster generation are referred to Table S8.

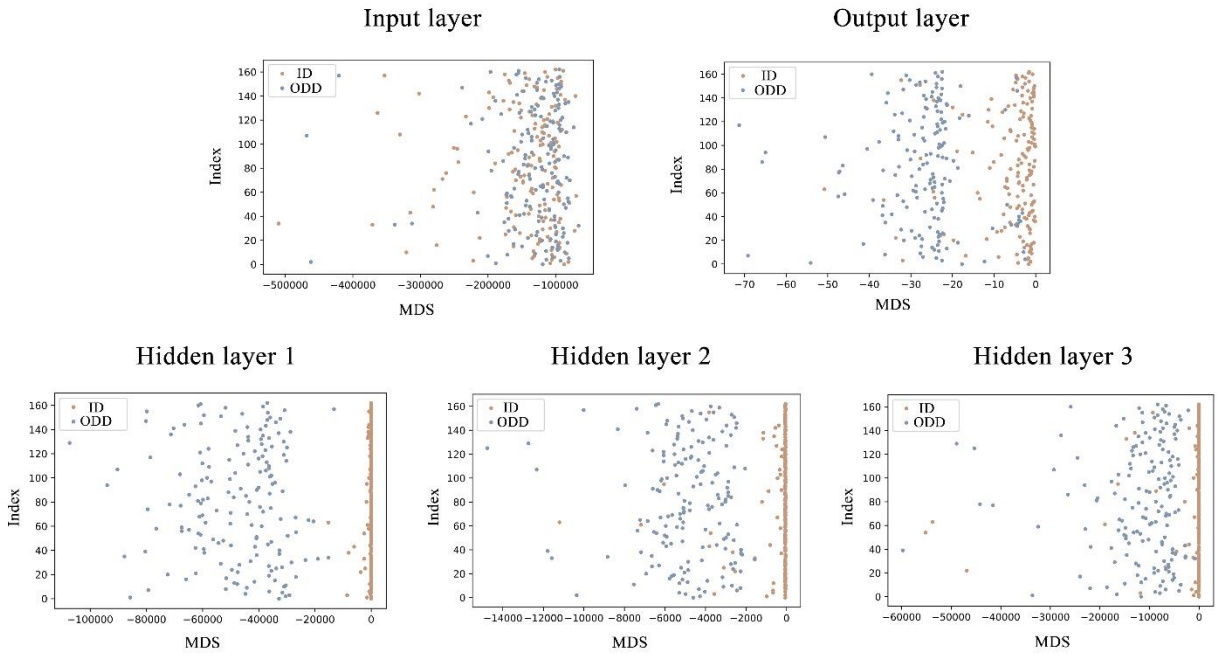

Figure S4. Two-dimensional representations of the features extracted from each layer of enhanced T2PK classifier model. The in-distribution data comprised 163 labeled KS_{β}, while the out-of-distribution data comprised 163 labeled KS_{α}. The Mahalanobis distance-based scores calculated by Gaussian discriminant analysis are plotted on the X-axis, while the Y-axis denotes the index of each datapoint in their dataset.

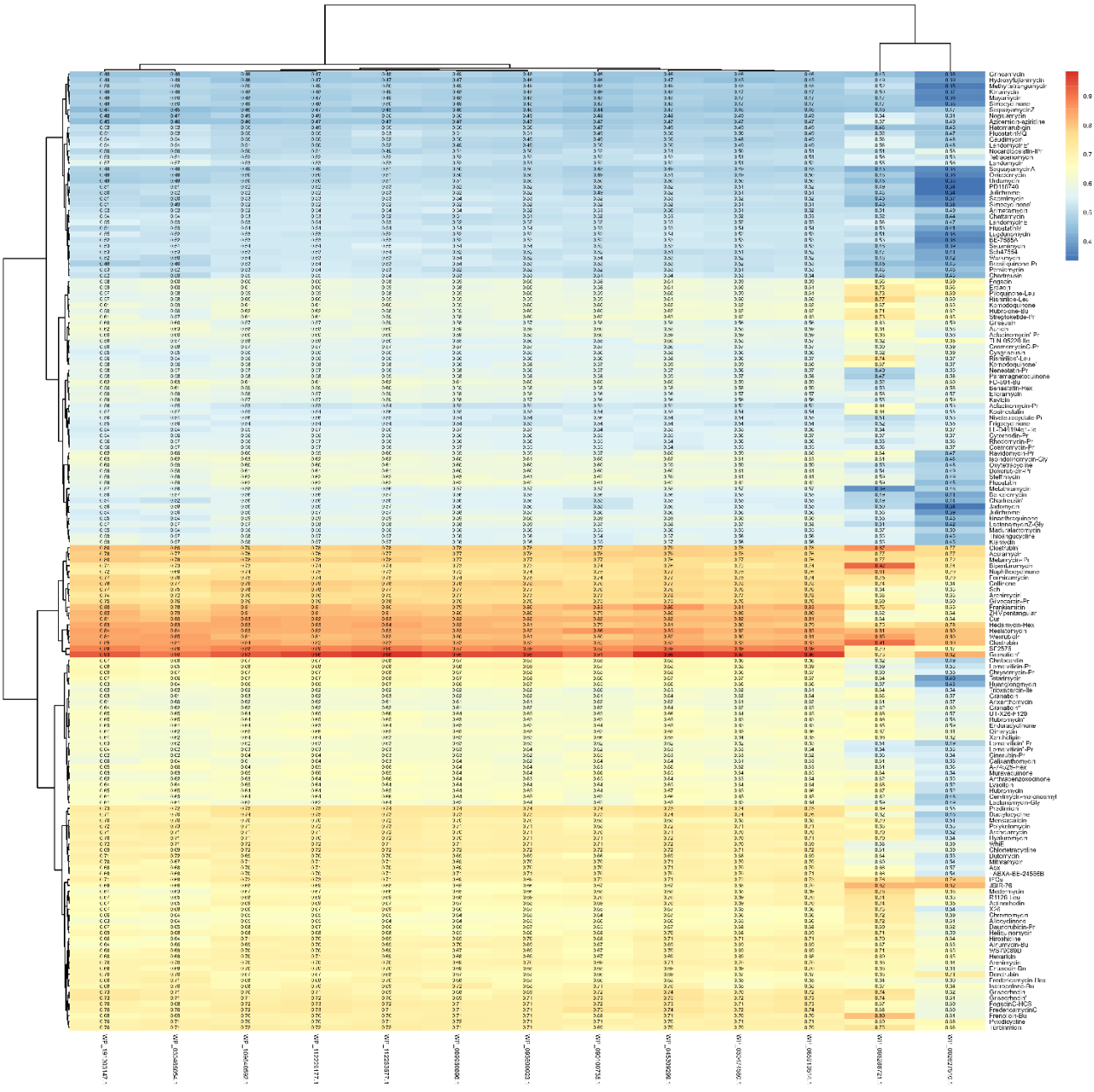

Figure S5. Heatmap representation showing the root mean square deviation (RMSD) of the predicted protein structure between 13 novel KS_{β} in cluster 1 and 163 labeled KS_{β}. Detailed values are referred to Table S5.

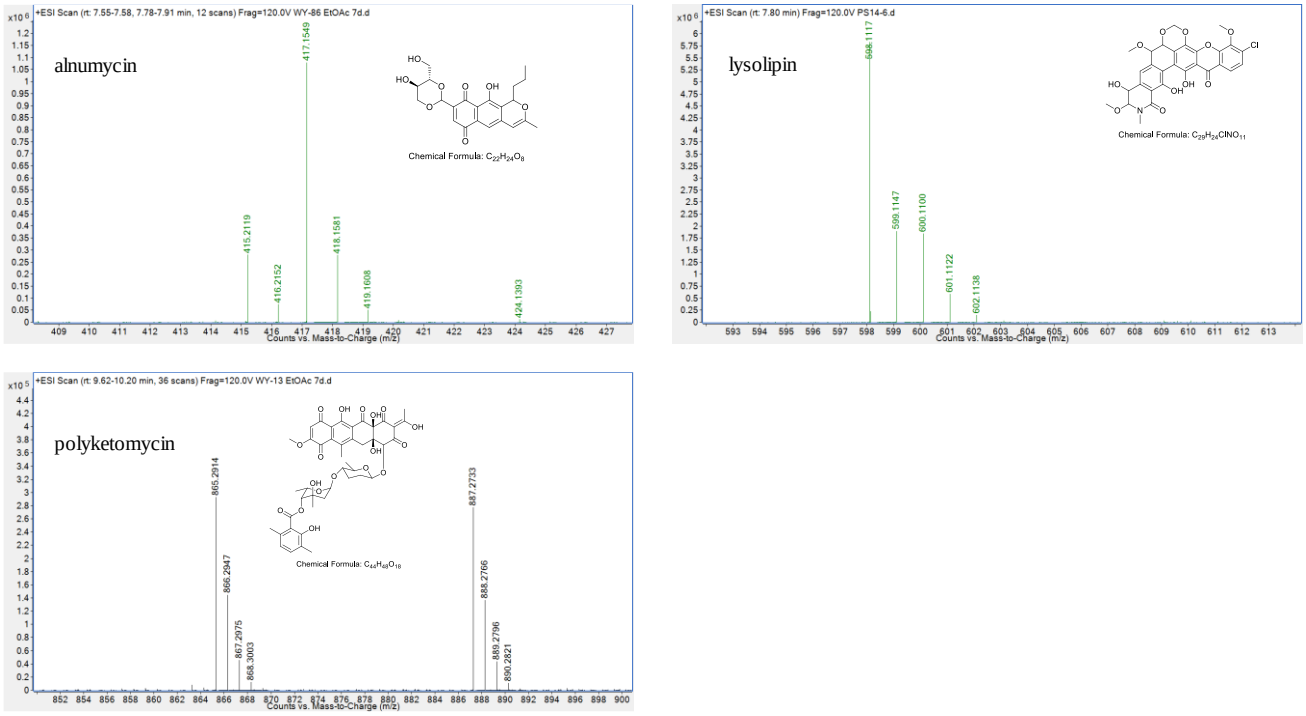

Figure S6. Identification of T2PKs detected in this work. Liquid chromatography high-resolution mass spectrometry (LCHRMS) showing alnumycin (molecular formula $C_{22}H_{24}O_8$, calculated $[M + H]^+ = 417.1544$, observed $[M + H]^+ = 417.1549$, $\Delta = 1.2$ ppm), polyketomycin (molecular formula $C_{44}H_{48}O_{18}$, calculated $[M + H]^+ = 865.2913$, observed $[M + H]^+ = 865.2914$, $\Delta = 1$ ppm), , and lysolipin (molecular formula $C_{29}H_{24}NO_{11}Cl$, calculated $[M + H]^+ = 598.1111$, observed $[M + H]^+ = 598.1117$, $\Delta = 0.12$ ppm), respectively.

Supplementary Tables

Note that deo to the scales Table S1-2, S4-5 are deposited as independent Excel spreadsheets.

Table S3. Performance metrics of KS_{β} classifier trained by Random Forest, XGBoost, support vector machine (SVM), and multilayer perceptron (MLP). TPR: True positive rate; FPR: False positive rate.

Model	TPR (%)	FPR (%)	Accuracy	Precision	Recall	F1-score
Random Forest	98.89	1.11	1.00	1.00	0.99	0.99
XGBoost	99.63	0.37	1.00	1.00	1.00	1.00
SVM	100.00	0.00	1.00	1.00	1.00	1.00
MLP	100.00	0.00	1.00	1.00	1.00	1.00

Table S6. Average root-mean-square deviation (RMSD) calculation between ODD cluster 1 and IV, V, VI, VII classes.

	IV	V	VI	VII	ODD cluster 1
IV	0.24				
V	0.53	0.39			
VI	0.48	0.57	0.29		
VII	0.45	0.57	0.58	0.30	
ODD cluster 1	0.51	0.59	0.62	0.58	0.20

Table S7. Performance comparison on predicting T2PKs using DeepT2, DeepBGC and antiSMASH.

Strain name	Genome Size (bp)	DeepBGC	Antismash		Our model (DeepT2)			
			Counts	T2PK name	Counts	Class	T2PK name	Euclidean distance
S. sp. PS14	6848633	polyketide	2	Spore pigment Lysolipin	2	VIII IX	Cur Sch Formicamycin Lysolipin Anthrabenzoxocinone +ABXA-BE-24566B	0.87 0.88 0.92 0.45 0.61 0.62
S. sp. WY13	10713003	polyketide	3	Polyketomycin Gaudimycin Spore pigment	3	VI VII VIII	Dutomycin LL-D49194α1-Ile Polyketomycin Gaudimycin LandomycinE Landomycin Collinone Cur Sch	0.33 0.42 0.45 0.68 0.69 0.71 0.69 0.82 1.07
S. sp. WY86	11617619	polyketide	1	Alnumycin	1	I	Frenolicin-Bu Alnumycin-Bu Granaticin	0.41 0.58 0.60
S. kanamyceticus 4.1441	10265719	polyketide	3	Aurachin cinerubin B Formicamycin	3	I VI VIII	Isoindolinomycin-Gly R1128-Leu Fogacin Keyicin Chrysomycin-Pr Gilvocarcin-Pr Accramycin (fasamycin) Formicamycin Sch	0.34 0.56 0.57 0.25 0.42 0.44 0.61 0.75 1.40
S. sp. WY170	8864212	polyketide	1	Spore pigment	1	VIII	ZHMpentangular Cur WhiE	0.80 0.82 0.88

Table S8. General information on selected hyperparameters to all classifier.

Classifier type	Model type	Hyperparameters

Binary KS_{β} classifier	Random forest	n_estimators = 150;
		max_features='sqrt';
		min_samples_split=4;
		min_samples_leaf=1;
		max_depth=6

	XGBoost	learning_rate=0.1;
		max_depth=5;
		min_child_weight=1;
		subsample=0.7

	Multilayer perceptron	num_epochs=20;
		learning_rate=0.001;
		activation='relu';
		solver='adam';
		neuron = 50;
		hidden_layer = 3

	Support vector machine	C=10; gamma=1; kernel=' rbf'

Initial T2 PK classifier	Random forest	n_estimators = 150;
		max_features='log2';
		min_samples_split=5;
		min_samples_leaf=1;
		max_depth=6

	XGBoost	learning_rate=0.2;
		max_depth=3;
		min_child_weight=1;
		subsample=0.6

	Multilayer perceptron	num_epochs=120;
		learning_rate=0.001;
		activation='tanh';
		solver='adam';
		neuron=200;
		hidden_layer=3;
		dropout=0.5
		train_loader_batch_size=12
		test_loader_batch_size=3

	Support vector machine	C=10; gamma=1; kernel=' rbf';

Enhanced T2 PK classifier	Multilayer perceptron	num_epochs=500;
		learning_rate=0.001;
		activation='tanh';
		solver='adam';
		neuron=200;
		hidden_layer=3;
		Gaussian_noise_mean=0
		Gaussian_noise_stddev =0.07
		train_loader_batch_size=12
		test_loader_batch_size=3
		merge_loader_batch_size=64
		Unsupervised loss weight:
		max_val=50;
		ramp_up_multi = -2
		max_epochs=300
		n_labeled = 163
		n_samples = 2729

Table S9. General information on selected hyperparameters for each labeling process. n_neighbors, n_components, min_dist, random_state are the hyperparameters of UMAP; min_cluster_size, min_samples, cluster_selection_epsilon are the hyperparameters of HDBSCAN; max_evals is the hyperparameter of *Fmin* function from *hyperopt* packages. label_count is the hyperparameter of label cost function. Following hyperparameters were set as default along the tunning**:** n_components = 3; min_cluster_size = 2; min_samples = None; random_state = 42; max_evals = 100.

Stage	**Range for hyperparameters bayesian optimization**	**Optimized hyperparameters**
Initial Class label to Class label A	n_neighbors = [20, 50], step = 1	n_neighbors = 46
	min_dist = [0.1, 1], step = 0.1	min_dist = 1
	cluster_selection_epsilon = [1, 2], step = 0.2	cluster_selection_epsilon = 1.6
	label_count = 3	cluster_count = 3
Class label A to Class label B	n_neighbors = [20, 46], step = 1	n_neighbors = 35
	min_dist = [0.1, 1], step = 0.1	min_dist = 0.3
	cluster_selection_epsilon = [0.8, 1.6], step = 0.2	cluster_selection_epsilon = 0.8
	label_count = 4	cluster_count = 4
Class label B to Class label C	n_neighbors = [20, 35], step = 1	n_neighbors = 20
	min_dist = [0.02, 0.3], step = 0.01	min_dist = 0.05
	cluster_selection_epsilon = [0.6, 1.4], step = 0.2	cluster_selection_epsilon = 0.8
	label_count = 5	cluster_count = 5
Class label C to Class label D	n_neighbors = [10, 20], step = 1	n_neighbors = 14
	min_dist = [0.02, 0.1], step = 0.01	min_dist = 0.02
	cluster_selection_epsilon = [0.2, 1], step = 0.2	cluster_selection_epsilon = 0.4
	label_count = 6	cluster_count = 6
Class label D to Class label E	n_neighbors = [5, 14], step = 1	n_neighbors = 12
	min_dist = [0.02, 0.1], step = 0.01	min_dist =0.02
	cluster_selection_epsilon = [0.2, 1], step = 0.2	cluster_selection_epsilon = 0.6
	label_count = 7	cluster_count = 7
Class label E to Class label F	n_neighbors = [5, 14], step = 1	n_neighbors = 9
	min_dist = [0.0, 0.05], step = 0.01	min_dist =0.03
	cluster_selection_epsilon = [0.2, 0.8], step = 0.2	cluster_selection_epsilon = 0.4
	label_count = 8	cluster_count = 8
Class label F to Final Class label	n_neighbors = [3, 15], step = 1	n_neighbors = 7
	min_dist = [0.0, 0.05], step = 0.01	min_dist = 0.0
	cluster_selection_epsilon = [0.0, 0.8], step = 0.2	cluster_selection_epsilon = 0.6
	label_count = 9	cluster_count = 9

References

1. S. Chen, C. Zhang and L. Zhang, *Angewandte Chemie International Edition*, 2022, **61**, e202202286.

2. L. McInnes, J. Healy and S. Astels, *Open Source Softw.*, 2017, **2**, 205.

3. L. McInnes, J. Healy and J. Melville, *arXiv preprint arXiv:.03426*, 2018, DOI: 10.48550/arXiv.1802.03426.

4. S. Laine and T. Aila, *arXiv:1610.02242*, 2016, DOI: 10.48550/arXiv.1610.02242.

5. K. Lee, K. Lee, H. Lee and J. Shin, *Advances in neural information processing systems*, 2018, **31**.

6. Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli and Y. Shmueli, *Science*, 2023, **379**, 1123-1130.

7. S. Andrews, *Journal*, 2010.

8. R. R. Wick, L. M. Judd, C. L. Gorrie and K. E. Holt, *Microb Genom*, 2017, **3**, e000132.

9. R. R. Wick, L. M. Judd, C. L. Gorrie and K. E. Holt, *PLoS Comput Biol*, 2017, **13**, e1005595.

10. S. Koren, B. P. Walenz, K. Berlin, J. R. Miller, N. H. Bergman and A. M. Phillippy, *Genome Res*, 2017, **27**, 722-736.

11. R. Vaser, I. Sovic, N. Nagarajan and M. Sikic, *Genome Res*, 2017, **27**, 737-746.

12. T. Seemann, *Bioinformatics*, 2014, **30**, 2068-2069.

13. G. D. Hannigan, D. Prihoda, A. Palicka, J. Soukup, O. Klempir, L. Rampula, J. Durcak, M. Wurst, J. Kotowski, D. Chang, R. R. Wang, G. Piizzi, G. Temesi, D. J. Hazuda, C. H. Woelk and D. A. Bitton, *Nucleic Acids Research*, 2019, **47**.

14. B. Kai, S. Simon, K. Alexander M, C.-P. Zach, V. W. Gilles P, M. Marnix H and W. Tilmann, *Nucleic acids research*, 2021, **49**, W29-W35.

15. M. Wang, J. J. Carver, V. V. Phelan, L. M. Sanchez, N. Garg, Y. Peng, D. D. Nguyen, J. Watrous, C. A. Kapono, T. Luzzatto-Knaan, C. Porto, A. Bouslimani, A. V. Melnik, M. J. Meehan, W. T. Liu, M. Crusemann, P. D. Boudreau, E. Esquenazi, M. Sandoval-Calderon, R. D. Kersten, L. A. Pace, R. A. Quinn, K. R. Duncan, C. C. Hsu, D. J. Floros, R. G. Gavilan, K. Kleigrewe, T. Northen, R. J. Dutton, D. Parrot, E. E. Carlson, B. Aigle, C. F. Michelsen, L. Jelsbak, C. Sohlenkamp, P. Pevzner, A. Edlund, J. McLean, J. Piel, B. T. Murphy, L. Gerwick, C. C. Liaw, Y. L. Yang, H. U. Humpf, M. Maansson, R. A. Keyzers, A. C. Sims, A. R. Johnson, A. M. Sidebottom, B. E. Sedio, A. Klitgaard, C. B. Larson, C. A. B. P, D. Torres-Mendoza, D. J. Gonzalez, D. B. Silva, L. M. Marques, D. P. Demarque, E. Pociute, E. C. O'Neill, E. Briand, E. J. N. Helfrich, E. A. Granatosky, E. Glukhov, F. Ryffel, H. Houson, H. Mohimani, J. J. Kharbush, Y. Zeng, J. A. Vorholt, K. L. Kurita, P. Charusanti, K. L. McPhail, K. F. Nielsen, L. Vuong, M. Elfeki, M. F. Traxler, N. Engene, N. Koyama, O. B. Vining, R. Baric, R. R. Silva, S. J. Mascuch, S. Tomasi, S. Jenkins, V. Macherla, T. Hoffman, V. Agarwal, P. G. Williams, J. Dai, R. Neupane, J. Gurr, A. M. C. Rodriguez, A. Lamsa, C. Zhang, K. Dorrestein, B. M. Duggan, J. Almaliti, P. M. Allard, P. Phapale, L. F. Nothias, T. Alexandrov, M. Litaudon, J. L. Wolfender, J. E. Kyle, T. O. Metz, T. Peryea, D. T. Nguyen, D. VanLeer, P. Shinn, A. Jadhav, R. Muller, K. M. Waters, W. Shi, X. Liu, L. Zhang, R. Knight, P. R. Jensen, B. O. Palsson, K. Pogliano, R. G. Linington, M. Gutierrez, N. P. Lopes, W. H. Gerwick, B. S. Moore, P. C. Dorrestein and N. Bandeira, *Nat Biotechnol*, 2016, **34**, 828-837.