Predicting and generating antibiotics against future pathogens with ApexOracle

Tianang Leng*
Department of Bioengineering
University of Pennsylvania
Philadelphia, PA 19104
ltianang@seas.upenn.edu

Fangping Wan*
Perelman School of Medicine
University of Pennsylvania
Philadelphia, PA 19104
fangping.wan@pennmedicine.upenn.edu

Marcelo Der Torossian Torres
Perelman School of Medicine
University of Pennsylvania
Philadelphia, PA 19104
marcelot@pennmedicine.upenn.edu

Cesar de la Fuente-Nunez
Perelman School of Medicine
University of Pennsylvania
Philadelphia, PA 19104
cfuente@pennmedicine.upenn.edu

Abstract

Antimicrobial resistance (AMR) is accelerating beyond antibiotic development. We introduce ApexOracle, a multimodal framework that unifies antibacterial efficacy modeling and *de novo* molecule generation against unseen pathogens. ApexOracle integrates heterogeneous modalities—molecular features from a discrete diffusion model together with genomic- and text-based strain embeddings derived from multiple foundation models—enabling pathogen-specific modeling. Across diverse species and modalities, it outperforms state-of-the-art predictors, generalizes to novel strains with little data, and generates "new-to-nature" molecules with high predicted efficacy. This unique representation–generation framework provides a scalable approach to address future AMR and emerging infectious threats.

1 Introduction

Antibiotic resistance is a global crisis, making once-curable infections increasingly deadly, with an estimated 4.95 million deaths annually linked to drug-resistant pathogens[1]. Yet developing a new antibiotic takes years, and future pathogens—natural or engineered—may evade all existing drugs. AI has accelerated antimicrobial discovery[2–6], but most work covers only parts of the pipeline. Many efforts [7–12] predict general AMPs without pathogen-specific activity. APEX [5] handles multiple fixed strains but struggles to generalize due to target rigidity and scarce MIC data against new pathogen strains. Small-molecule models [13, 14] are likewise strain-specific and data-hungry. A recent study [15] attempted to overcome this by embedding each genome into a vector of k-mer composition profiles; however, this strategy effectively erases gene-level signals, produces non-interpretable representations, and was evaluated on only three strains, leaving its broader applicability untested. Overall, these approaches employ separate peptide-only or small-molecule-only models, restricting cross-modal generalization and leaving valuable data under-utilized. Generation models [16, 9, 11] suffer from the same limitation. Therefore, a unified approach that can leverage heterogeneous molecular data (e.g., small molecules and peptides) and generalize across diverse bacterial strains is urgently needed.

*These authors contributed equally to this work.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: 2nd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences.

To bridge these gaps, we developed ApexOracle (Fig.1), a unified machine learning platform for pathogen knowledge-aware antimicrobial discovery. ApexOracle uniquely integrates multimodal inputs, including pathogen genome, textual knowledge, and molecular features. By conditioning on pathogenic context, ApexOracle is designed to generalize to unseen strains, inferring susceptibilities and proposing new molecules. Trained on SELFIES [17], ApexOracle can seamlessly handle both small molecules and peptides, including those bearing non-canonical residues and modifications. This is critical because 71.5% of DBAASP AMPs [18] contain non-canonical elements, yet many conventional predictors assume purely canonical, linear peptides and thus oversimplify the AMP landscape.

We evaluate ApexOracle across diverse bacterial and fungal strains, showing that its multimodal design achieves state-of-the-art accuracy in predicting antimicrobial efficacy for both small molecules and peptides. By conditioning its generative module on pathogen embeddings and guiding it with an internal predictor, ApexOracle also designs de novo candidates with high predicted potency. This integrated approach enables simultaneous prediction and generation of therapeutics across a vast chemical space, offering a new paradigm for antimicrobial discovery—one that proactively responds to resistance by leveraging pathogen-informed design. We conclude with a discussion of ApexOracle's broader implications for combating infectious disease and future directions to align AI-driven drug design with urgent clinical needs.

2 ApexOracle: Omnimodel for pRedictive and generAtive antimiCrobiaL discovEry

2.1 ApexOracle architecture

To enable generalization to previously unseen strains, we built ApexOracle as a multimodal deep learning architecture (Fig. 1). The model accepts three inputs representing a pathogen-drug scenario: (1) the pathogen's genome sequence, (2) a textual description of the pathogen's traits, and (3) a molecule (either an existing drug or a candidate structure). From these, ApexOracle produces two kinds of outputs: a predicted efficacy (how well the given antibiotic would work against the pathogen) and/or a generated molecule (a new chemical structure predicted to be effective against a pathogen of interest). The design enables prediction and generation within one unified framework.

Text modality for pathogen descriptions. We utilized Qwen2.5-Max [19] (version 01.25.2025) to search for important strain traits from reliable scientific publications. These traits include: description of the species to which this strain belongs, notable physiological traits, distinctive genetic mutations recorded to have impact on metabolic pathways or plasma membrane, and antibiotics and antimicrobial peptides to which this strain is known to be sensitive or resistant. See Appendix.E for the full prompt. Obtained descriptions were then inputted to Me-LLaMA [20] to obtain latent embeddings. For each strain, the descriptive text is first preprocessed by replacing the specific strain name with "this strain" to prevent overfitting to particular strain names.

Genome modality for pathogens. Pathogen genomes downloaded from ATCC or NCBI typically consist of multiple contigs. For each contig, we employ a sliding window approach to fragment it. Specifically, we use a step size of 10,000 nt and a window length of 11,000 nt, ensuring that each window is 1,000 bases longer than the step size to prevent genes from being truncated and thus feed imperfect information to the downstream model. Each extracted fragment is then inputted into Evo-2 (40B version), from which we extract the latent embedding of the 46^{th} layer for all bases. Contig-level embedding is the mean pooling embedding of the nucleotide embeddings in the contig. We concatenate all contig-level embeddings to create genome-level embedding.

Molecular modality via diffusion language model We choose to use diffusion language model (DLM) to model discrete SELFIES tokens of molecules for molecule representation learning and molecule generation. We choose the MDLM framework [21], whose forward process progressively adds noise by replacing actual SELFIES tokens with `<MASK>` tokens, and reverse process uses a language model to reconstructs the original SELFIES tokens from these `<MASK>` tokens.

Assuming we have a sequence of L SELFIES tokens $\mathbf{x}_t^{1:L}$ (tokens are represented by one-hot encoding) at time t, and that the tokens in $\mathbf{x}_{t-1}^{1:L}$ are conditionally independent given $\mathbf{x}_t^{1:L}$ and the neural network denoiser $\text{NN}_\theta(\mathbf{x}_t^{1:L}, t)$ parameterized by θ, then under a log-linear noise schedule, where $\alpha_t = \exp(-r(t))$ and $r(t) = -\log(1-t)$, the optimization objective of the DLM—namely,

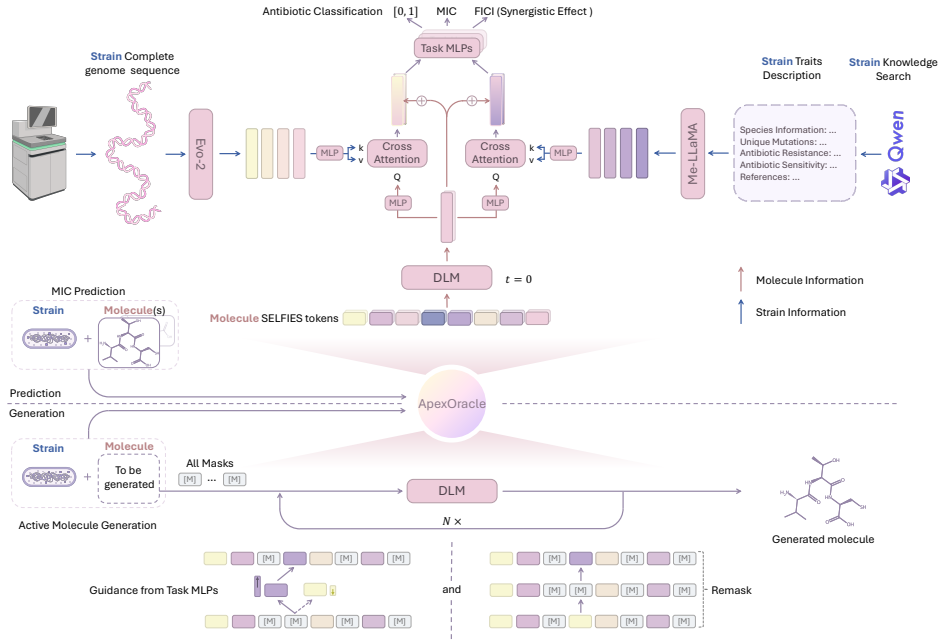

Figure 1: **Overview of ApexOracle:** a unified multimodal model integrating strain genome, textual knowledge, and molecular representations for both antimicrobial activity prediction and *de novo* molecule generation.

the negative evidence lower bound—can be defined as:

$$\mathcal{L}_{\text{DLM}} = -\mathbb{E}_{t\sim\mathcal{U}(0,1],\, q(\mathbf{x}_t^{1:L}|\mathbf{x}_0^{1:L})} \left[\frac{1}{t} \sum_{l=1}^{L} \log \left\langle \text{NN}_\theta^l(\mathbf{x}_t^{1:L}, t),\, \mathbf{x}_0^l \right\rangle \right]. \tag{1}$$

Here $\mathcal{U}(0,1]$ is an uniform distribution, and $< \cdot, \cdot >$ stands for dot product.

We further add a multitask regression (MTR) loss $\mathcal{L}_{\text{MTR}}$: the MSE between an MLP's predictions and 209 RDKit descriptors (Fig. 4). The MLP takes the DiT final-layer embedding from a clean SELFIES. We train with $\mathcal{L}_{\text{ApexOracle-DLM}} = \mathcal{L}_{\text{DLM}} + \lambda\, \mathcal{L}_{\text{MTR}}$ using $\lambda = 0.1$.

Molecule-Strain Knowledge Fusion. Once we have a trained molecule DLM, we can feed a clean molecule SEIFIES string into DLM and treat the beginner token embedding from the last transformer layer of DiT as the its feature representation. ApexOracle then uses MLPs to transform molecule feature representations, the genome and text embeddings of pathogens into new hidden spaces, and fuse the molecular information with pathogen strain-information via the cross attention mechanism. Here we treat the molecule's representation as a single query Q that attends in parallel to two separate pathogen strain-knowledge banks: one comprising genome-derived key–value pairs $(K_{\text{genome}}, V_{\text{genome}})$ and the other comprising text-derived key–value pairs $(K_{\text{text}}, V_{\text{text}})$ (Fig.1). In each cross-attention block, the molecule query (Q) computes attention weights over the respective keys and then updates itself with the corresponding values via a residual connection. The resulting molecule-genome and molecule-text context vectors are concatenated into a unified embedding that seamlessly combines molecular features with pathogen strain-specific insights. This fused representation is finally passed through task-specific neural network heads to predict antimicrobials.

3 Experiments

We evaluated ApexOracle on antimicrobial activity prediction tasks and a controlled antibioitc generation task.

3.1 ApexOracle generalizes antimicrobial prediction to unseen strains

To simulate a "future pathogen" scenario, we designed three hierarchical evaluation strategies for regression of MICs retrieved from antimicrobial peptide database DBAASP [18], each holding out a different level of clusters of strains entirely during training.

We began with the most evolution-like test, strain-wise evaluation, which probes generalization within a single species. For each species, we conducted three-fold validation: two-thirds of its strains were used for training, and the remaining one-third for testing. ApexOracle achieved an average R2 of 0.5793 with an ensemble of seven models (trained under different random seeds) and 0.5032 without ensembling on held-out unseen strains. We further benchmarked several alternative atom-level models as molecular encoders within ApexOracle and found that our DLM consistently outperformed them in the pathogen strain-knowledge-aware setting; without ensembling, the DLM achieved a 13.5% higher R2 than the next-best model (Table.1 and Fig.3b).

Table 1: Benchmarking results on strain-wise MIC prediction

	ChemBERTa (MLM)	ChemBERTa (MTR)	MolFormer	PeptideCLM	**DLM (Ours)**	**DLM (Ours, 7 models ensemble)**
R^2 ↑	0.3477	0.4433	0.3779	0.4602	0.5032	**0.5793**
Spearman ↑	0.6016	0.6732	0.6264	0.6645	0.7143	**0.7472**
Pearson ↑	0.6118	0.6766	0.6370	0.6763	0.7239	**0.7597**

Beyond strain-wise evaluation, we applied stricter species-level clustering based on taxonomic distance: one with three clusters and another with eleven. The 3-cluster setting (Fig.3c) grouped strains into Fungi, Pseudomonadota (Gram-negative), and Bacillota (Gram-positive), while the 11-cluster setting (Fig.3d) further subdivided them (2, 4, and 5 subgroups, respectively). In both cases, one cluster was held out for testing (Fig.3e, 3f). The 3-cluster-wise evaluation posed the greatest challenge by maximizing taxonomic divergence, yet ApexOracle still achieved an average R^2 of 0.3744 (Fig.3g). In the 11-cluster evaluation, performance improved to $R^2 = 0.4337$, excluding the sole exception of cluster 9 (Mycoplasmatota, Fig.3d, 3f), whose atypical membranes complicate prediction. Altogether, these results highlight ApexOracle's ability to generalize across unseen pathogen clades, leveraging genetic and molecular context rather than memorizing strain-specific patterns—an essential capacity for anticipating novel pathogens.

3.2 Designing novel antibiotics for unseen strains via predictor-guided generation

Beyond prediction, ApexOracle functions as a *de novo* antimicrobial designer. Its generative module accepts our pathogen knowledge-informed antimicrobial prediction as conditional signals and outputs candidate SELFIES strings predicted to exploit that organism's specific vulnerabilities. We used D-CBG in [22] to perform predictor-guided generation. To test this capacity in a "future-pathogen" scenario, we instructed ApexOracle to propose molecules active against two high-priority strains it had never seen: *E. coli* BAA-3170 (colistin-resistant) and *P. aeruginosa* BAA-3197 (fluoroquinolone-, beta-lactam-, and carbapenem-resistant). Relative to unconditional sampling, our predictor-guided

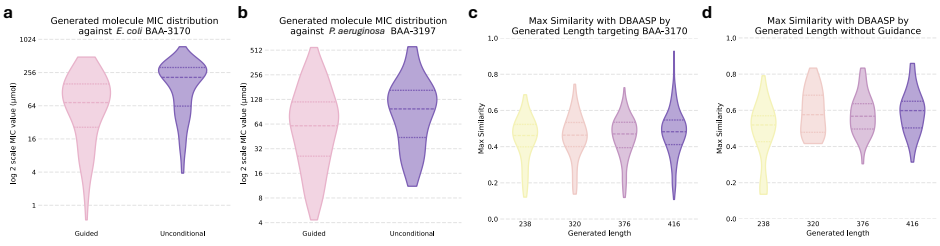

Figure 2: **Predicted MIC values and novelty of generated molecules. a.** Predicted MIC distributions for molecules generated with versus without pathogen guidance targeting *E. coli* BAA-3170. **b.** Same comparison for *P. aeruginosa* BAA-3197. **c.** Max Tanimoto similarity of pathogen-guided molecules (binned by token length) to training data, targeting *E. coli* BAA-3170. **d.** Same comparison for unconditionally generated molecules.

generation produced predicted MIC distributions strongly shifted toward lower values (Fig.2a, 2b). We then computed each generated molecule's maximum Tanimoto similarity to compounds in

DBAASP—the same dataset that trained the MIC predictor steering generation. Guided molecules showed lower similarity than those from unconditional runs (Fig.2c, 2d), indicating that ApexOracle is designing genuinely novel chemotypes rather than copying training examples.

4 Conclusion

ApexOracle introduces an AI-driven framework that integrates molecular features, pathogen genomics and text-based biological knowledge to both predict antibiotic effectiveness and generate novel antibacterials. By unifying prediction and generation, the platform enables pathogen-specific drug discovery and explores unconventional chemical space to combat resistance. Overall, ApexOracle represents a step toward proactive, next-generation antibiotic design capable of addressing emerging and resistant pathogens. Future versions will incorporate results from wet lab experiments.

References

[1] JIM O'neill. Antimicrobial resistance: tackling a crisis for the health and wealth of nations. In *Rev. Antimicrob. Resist.*, 2014.

[2] Felix Wong, Cesar de la Fuente-Nunez, and James J Collins. Leveraging artificial intelligence in the fight against infectious diseases. *Science*, 381(6654):164–170, 2023.

[3] Jacqueline RMA Maasch, Marcelo DT Torres, Marcelo CR Melo, and Cesar de la Fuente-Nunez. Molecular de-extinction of ancient antimicrobial peptides enabled by machine learning. *Cell Host & Microbe*, 31(8):1260–1274, 2023.

[4] Célio Dias Santos-Júnior, Marcelo DT Torres, Yiqian Duan, Álvaro Rodríguez Del Río, Thomas SB Schmidt, Hui Chong, Anthony Fullam, Michael Kuhn, Chengkai Zhu, Amy Houseman, et al. Discovery of antimicrobial peptides in the global microbiome with machine learning. *Cell*, 187(14):3761–3778, 2024.

[5] Fangping Wan, Marcelo DT Torres, Jacqueline Peng, and Cesar de la Fuente-Nunez. Deep-learning-enabled antibiotic discovery through molecular de-extinction. *Nature Biomedical Engineering*, 8(7):854–871, 2024.

[6] Marcelo DT Torres, Marcelo CR Melo, Laurice Flowers, Orlando Crescenzi, Eugenio Notomista, and Cesar de la Fuente-Nunez. Mining for encrypted peptide antibiotics in the human proteome. *Nature Biomedical Engineering*, 6(1):67–75, 2022.

[7] Jing Xu, Fuyi Li, André Leier, Dongxu Xiang, Hsin-Hui Shen, Tatiana T Marquez Lago, Jian Li, Dong-Jun Yu, and Jiangning Song. Comprehensive assessment of machine learning-based methods for predicting antimicrobial peptides. *Briefings in bioinformatics*, 22(5):bbab083, 2021.

[8] Katarzyna Sidorczuk, Przemysław Gagat, Filip Pietluch, Jakub Kała, Dominik Rafacz, Laura Bąkała, Jadwiga Słowik, Rafał Kolenda, Stefan Rödiger, Legana CHW Fingerhut, et al. Benchmarks in antimicrobial peptide prediction are biased due to the selection of negative data. *Briefings in Bioinformatics*, 23(5):bbac343, 2022.

[9] Rui Wang, Tao Wang, Linlin Zhuo, Jinhang Wei, Xiangzheng Fu, Quan Zou, and Xiaojun Yao. Diff-amp: tailored designed antimicrobial peptide framework with all-in-one generation, identification, prediction and optimization. *Briefings in Bioinformatics*, 25(2):bbae078, 2024.

[10] Wenxuan Xing, Jie Zhang, Chen Li, Yujia Huo, and Gaifang Dong. iamp-attenpred: a novel antimicrobial peptide predictor based on bert feature extraction method and cnn-bilstm-attention combination model. *Briefings in bioinformatics*, 25(1):bbad443, 2024.

[11] Junhang Cao, Jun Zhang, Qiyuan Yu, Junkai Ji, Jianqiang Li, Shan He, and Zexuan Zhu. Tg-cddpm: text-guided antimicrobial peptides generation based on conditional denoising diffusion probabilistic model. *Briefings in Bioinformatics*, 26(1):bbae644, 2025.

[12] Nanjun Chen, Jixiang Yu, Liu Zhe, Fuzhou Wang, Xiangtao Li, and Ka-Chun Wong. Tp-lmmsg: a peptide prediction graph neural network incorporating flexible amino acid property representation. *Briefings in Bioinformatics*, 25(4):bbae308, 2024.

[13] Felix Wong, Erica J Zheng, Jacqueline A Valeri, Nina M Donghia, Melis N Anahtar, Satotaka Omori, Alicia Li, Andres Cubillos-Ruiz, Aarti Krishnan, Wengong Jin, et al. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 626(7997):177–185, 2024.

[14] Gary Liu, Denise B Catacutan, Khushi Rathod, Kyle Swanson, Wengong Jin, Jody C Mohammed, Anush Chiappino-Pepe, Saad A Syed, Meghan Fragis, Kenneth Rachwalski, et al. Deep learning-guided discovery of an antibiotic targeting acinetobacter baumannii. *Nature Chemical Biology*, 19(11):1342–1350, 2023.

[15] Chia-Ru Chung, Chung-Yu Chien, Yun Tang, Li-Ching Wu, Justin Bo-Kai Hsu, Jang-Jih Lu, Tzong-Yi Lee, Chen Bai, and Jorng-Tzong Horng. An ensemble deep learning model for predicting minimum inhibitory concentrations of antimicrobial peptides against pathogenic bacteria. *Iscience*, 27(9), 2024.

[16] Shuwen Jin, Zihan Zeng, Xiyan Xiong, Baicheng Huang, Li Tang, Hongsheng Wang, Xiao Ma, Xiaochun Tang, Guoqing Shao, Xingxu Huang, et al. Ampgen: an evolutionary information-reserved and diffusion-driven generative model for de novo design of antimicrobial peptides. *Communications Biology*, 8(1):1–14, 2025.

[17] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

[18] Malak Pirtskhalava, Anthony A Amstrong, Maia Grigolava, Mindia Chubinidze, Evgenia Alimbarashvili, Boris Vishnepolsky, Andrei Gabrielian, Alex Rosenthal, Darrell E Hurt, and Michael Tartakovsky. Dbaasp v3: database of antimicrobial/cytotoxic activity and structure of peptides as a resource for development of new therapeutics. *Nucleic acids research*, 49(D1): D288–D297, 2021.

[19] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[20] Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, et al. Me-llama: Foundation large language models for medical applications. *Research square*, pages rs–3, 2024.

[21] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[22] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallatorre, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

[23] Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[24] William Peebles and Saining Xie. Scalable diffusion models with transformers. *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[25] Walid Ahmad, Elana Simon, Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta-2: Towards chemical foundation models. *arXiv preprint arXiv:2209.01712*, 2022.

[26] Daniel M Lowe, Peter T Corbett, Peter Murray-Rust, and Robert C Glen. Chemical name to structure: Opsin, an open source solution, 2011.

[27] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022.

[28] Aaron L Feller and Claus O Wilke. Peptide-aware chemical language model successfully predicts membrane diffusion of cyclic peptides. *Journal of Chemical Information and Modeling*, 2025.

A Evaluation of ApexOracle prediction performance

A.1 Detailed modification information is crutial for AMP efficacy modeling

Conventional AMP models often ignore non-canonical features like modified amino acids, which are present in 71.5% of AMPs in DBAASP. To test the importance of these features, we benchmarked our DLM against other atom-level models and the state-of-the-art residue-level model, APEX, on predicting MIC for 19 bacterial strains. Our DLM, trained with a multi-task regression (MTR) objective, achieved the best performance, surpassing the next-best model by 27.1% in R^2 (Fig.3a). In contrast, APEX performed poorly because it cannot process non-canonical structures. This result underscores the critical role of atomic-level modifications in accurately predicting AMP activity while also demonstrating that the MTR training objective significantly improves molecular representation learning.

A.2 ApexOracle outperforms existing methods on small molecule antibiotics discovery without strain-specific labels

Beyond its strong capacity in AMP activity prediction, ApexOracle also excels at predicting small molecule antibiotic activity. Previous models typically depend on sizeable, direct training sets for each target strain [13, 14, 23], but ApexOracle leverages knowledge distilled from tens of thousands of molecules and pathogens—spanning multiple data modalities—to make accurate predictions even when no small molecule antibiotic labels exist for the strain of interest.

To quantify this advantage, we benchmarked ApexOracle against three state-of-the-art small-molecule classifiers [13, 14, 23], each trained on data measured against a specific pathogen strain *S. aureus* RN4220, *E. coli* BW25113, and *A. baumannii* ATCC 17978, respectively. In a zero-shot setting—without fine-tuning on strain-specific small-molecule data—ApexOracle, trained on AMP data along with the other two test-strain-irrelavant small-molecule antibiotics datasets, matched or outperformed two of the four fine-tuned baseline models (Fig.3h, pink).

For a direct comparison with prior work, we then mimicked each study's evaluation by using five-fold cross-validation with ensemble predictions from ten independently seeded models per fold. Under this setting, fine-tuned ApexOracle decisively outperformed every baseline, delivering average gains of 8.3% in AUROC and 37.7% in AUPRC across the three strains (Fig.3h, yellow).

These findings demonstrate that the broad molecular-pathogen knowledge captured by ApexOracle transfers effectively to new strains and molecules, providing substantial accuracy improvements without the prohibitive cost of assembling new strain-specific antibiotic dataset.

A.3 Modality ablation studies

To pinpoint which information streams drive ApexOracle's high accuracy, we performed systematic ablation experiments that removed one modality at a time—genomic features, pathogen-specific textual descriptions, or a small-molecule antibiotic classification task—and measured the impact on MIC regression performance (Fig.3i). Ablations were conducted on a curated subset of DBAASP containing 67,304 AMP–strain activity pairs for which both genome and text data were available, ensuring a fair comparison across settings. To accelerate testing while keeping all ablations orthogonal to molecular encoding, we substituted the DLM's molecule encoder with ChemBERTa, leaving the remainder of the architecture unchanged.

Eliminating any single modality degraded predictive performance, confirming that each contributes meaningfully. Removing genomic embeddings caused the sharpest drop, closely followed by excluding textual annotations, indicating strong synergy between those two pathogen-context channels. Dropping the auxiliary small-molecule classification task produced a smaller but still non-negligible decline, showing that cross-modal auxiliary learning refines molecular representations and improves AMP MIC generalization.

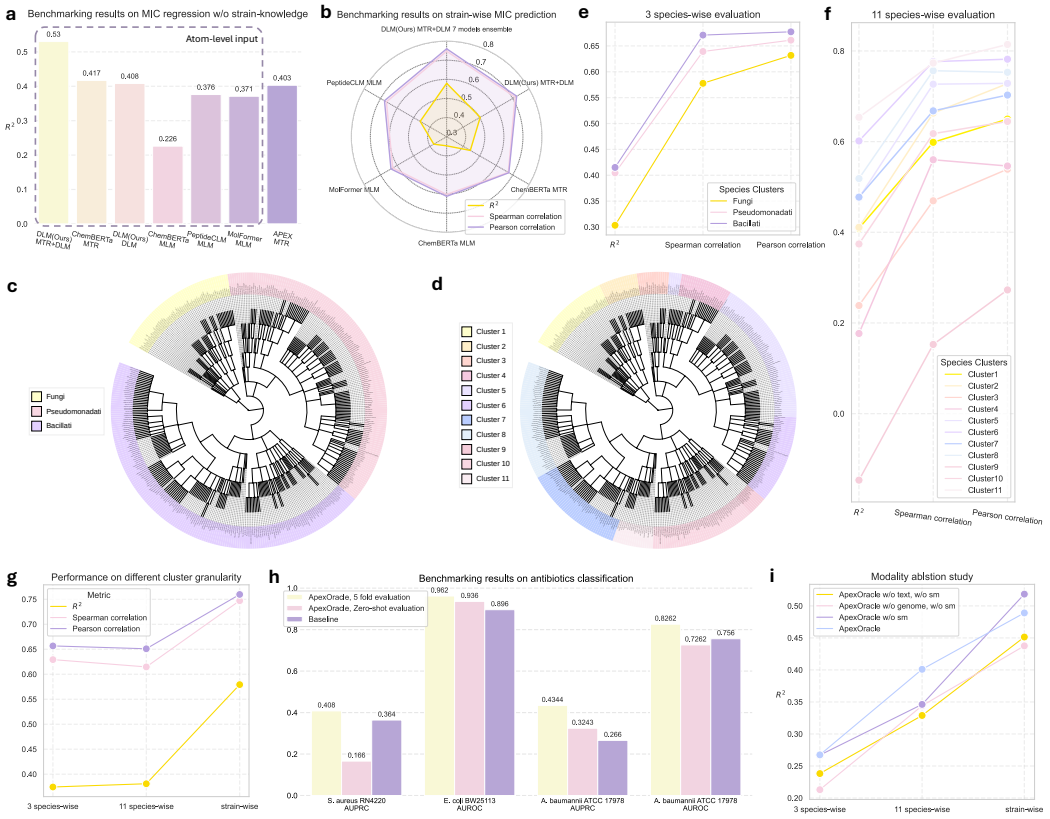

Figure 3: **Evaluation of ApexOracle prediction performance.** **a.** Benchmarking molecular representations from our diffusion language model (DLM) against alternatives for pathogen strain-knowledge-unaware MIC regression (R^2). All comparators except APEX were language models pretrained solely on atom-level inputs. APEX is a deep neural network that predicts antimicrobial activity from amino acid-level peptide inputs. **b.** Pathogen strain-knowledge–aware MIC prediction benchmarking under the strain-wise setting: our DLM and its ensemble consistently outperformed rival models. **c.** Hierarchical clustering of the dataset into three major taxonomic groups: Fungi, Bacillati (Gram-positive), and Pseudomonadati (Gram-negative). **d.** Hierarchical clustering into eleven distinct species-level clusters. **e.** Performance across three major taxonomic divisions (R^2, Spearman correlation, Pearson correlation). **f.** Performance across eleven finer species clusters. Robust accuracy was maintained for most clusters; the lone outlier (cluster 9, Mycoplasmatota) likely reflects its atypical biology. **g.** Impact of cluster granularity on ApexOracle performance. Results indicate that ApexOracle benefits from finer cluster granularity (strain-wise $>$ 11 species-wise $>$ 3 species-wise), while still maintaining moderate antimicrobial prediction performance on phylogenetically distant species. **h.** Small-molecule antibiotic classification benchmark. Even in zero-shot tests, ApexOracle surpassed baseline methods on two of three strains; with five-fold cross-validation and ensembling, it outperformed all baselines [13, 14, 23]. **i,** Modality ablation study for ApexOracle. Genome and text (annotation) information are of comparable importance and contribute synergistically to overall prediction performance.

B Diffusion language model.

Diffusion models are latent variable generative models characterized by a forward and a reverse Markov process. The forward process $q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ corrupts the data $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$ into a sequence of increasingly noisy latent variables $\boldsymbol{x}_{1:T} = \boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T$. The learned reverse Markov process $p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ gradually denoises the latent variables towards the data distribution, where θ represents the learnable parameters.

In our DLM, both the forward and reverse processes operate in the discrete SELFIES token space. During the forward process, noise is progressively added by replacing actual SELFIES tokens with <MASK> tokens. In the reverse process, a language model reconstructs the original SELFIES tokens from these <MASK> tokens. Let $\mathbf{x}_0$ be a K-dimensional one-hot encoding of a SELFIES token, where K stands for the vocabulary size of SELFIES token space (note that special tokens like <MASK> are included in the space). We followed the Masked Diffusion Language Model (MDLM) [21] workflow to define the absorbing-state forward diffusion process $q(\mathbf{x}_t|\mathbf{x}_0)$ as

$$q(\mathbf{x}_t|\mathbf{x}_0) = \text{Cat}(\mathbf{x}_t; \alpha_t \mathbf{x}_0 + (1 - \alpha_t)\mathbf{m}). \tag{1}$$

Here, $\alpha_t \in [0, 1]$ is a strictly decreasing function with respect to t, where $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$, representing the probability that $\mathbf{x}_0$ remains unchanged at time $t \in [0, 1]$. The vector $\mathbf{m}$ denotes the one-hot encoding of the <MASK> token, and $\text{Cat}(\cdot; \boldsymbol{\pi})$ denotes a categorical distribution over the K SELFIES tokens, parameterized by the probability simplex $\boldsymbol{\pi}$. The posterior $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ can be written as

$$q\big(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\big) = \begin{cases} \text{Cat}\big(\mathbf{x}_{t-1};\, \mathbf{x}_t\big), & \mathbf{x}_t \neq \mathbf{m}, \\ \text{Cat}\Big(\mathbf{x}_{t-1};\, \frac{(1-a_{t-1})\,\mathbf{m} + (a_{t-1}-a_t)\,\mathbf{x}_0}{1-a_t}\Big), & \mathbf{x}_t = \mathbf{m}. \end{cases} \tag{2}$$

In practice, we don't have access to the ground truth $\mathbf{x}_0$ in $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. Therefore, we seek to use a neural network $\text{NN}_\theta(\cdot, \cdot)$ parameterized by θ to take the time step t and its corresponding noisy sample $\mathbf{x}_t$ as inputs and estimate the clean data $\mathbf{x}_0$:

$$p_\theta\big(\mathbf{x}_{t-1} \mid \mathbf{x}_t, t\big) = \begin{cases} \text{Cat}\big(\mathbf{x}_{t-1};\, \mathbf{x}_t\big), & \mathbf{x}_t \neq \mathbf{m}, \\ \text{Cat}\Big(\mathbf{x}_{t-1};\, \frac{(1-a_{t-1})\,\mathbf{m} + (a_{t-1}-a_t)\,\text{NN}_\theta(\mathbf{x}_t, t)}{1-a_t}\Big), & \mathbf{x}_t = \mathbf{m}. \end{cases} \tag{3}$$

Regarding the output $\text{NN}_\theta(\mathbf{x}_t, t)$, we follow the SUBS parameterization in MDLM to further replace the logit output corresponding to the <MASK> token by $-\infty$ to make sure the neural network denoiser does not output <MASK> tokens.

Assuming we have a sequence of L SELFIES tokens $\mathbf{x}_t^{1:L}$ at time t, and that the tokens in $\mathbf{x}_{t-1}^{1:L}$ are conditionally independent given $\text{NN}_\theta(\mathbf{x}_t^{1:L}, t)$, then under a log-linear noise schedule, where $\alpha_t = \exp(-r(t))$ and $r(t) = -\log(1 - t)$, the optimization objective of the DLM—namely, the negative evidence lower bound—can be defined as:

$$\mathcal{L}_{\text{DLM}} = -\mathbb{E}_{t\sim\mathcal{U}(0,1],\, q(\mathbf{x}_t^{1:L}|\mathbf{x}_0^{1:L})} \left[\frac{1}{t} \sum_{l=1}^{L} \log \left\langle \text{NN}_\theta^l(\mathbf{x}_t^{1:L}, t),\, \mathbf{x}_0^l \right\rangle \right]. \tag{4}$$

Here $\mathcal{U}(0, 1]$ is an uniform distribution, and $< \cdot, \cdot >$ stands for dot product. We implemented our $\text{NN}_\theta(\cdot, \cdot)$ as a 12-layer Diffusion Transformer (DiT) [24] with a latent dimension of 768.

As $\mathcal{L}_{\text{DLM}}$ can be viewed as a weighted average of masked language modeling losses over different noise levels, we think DLM can also be used as a feature extractor from clean data inputs. Inspired by [25], we further enhance the representation learning capability of our DLM by introducing an additional multilayer perceptron (MLP), which takes as input the embedding of the beginner token (i.e., the <CLS> token) from the final transformer layer of DiT, computed from a clean SELFIES string. The MLP is trained to predict the corresponding 209-dimensional vectorized computational descriptors generated by RDKit for the same SELFIES string (Fig.4). We calculate the mean squared loss between predictions and labels, and term this loss as multitask regression loss (MTR) $\mathcal{L}_{\text{MTR}}$. Our final loss to train the DML is a weigthed combination of the two losses defined above:

$$\mathcal{L}_{\text{ApexOracle-DLM}} = \mathcal{L}_{\text{DLM}} + \lambda \cdot \mathcal{L}_{\text{MTR}}, \tag{5}$$

where we simply set $\lambda = 0.1$.

Molecule-Strain Knowledge Fusion. Once we have a trained molecule DLM, we can feed a clean molecule SEIFIES string into DLM and treat the beginner token embedding from the last transformer layer of DiT as the its feature representation. ApexOracle then uses MLPs to transform molecule feature representations, the genome and text embeddings of pathogens into new hidden spaces, and fuse the molecular information with pathogen strain-information via the cross attention mechanism. Here we treat the molecule's representation as a single query Q that attends in parallel to two separate pathogen strain-knowledge banks: one comprising genome-derived key–value pairs $(K_{\text{genome}}, V_{\text{genome}})$ and the other comprising text-derived key–value pairs $(K_{\text{text}}, V_{\text{text}})$ (Fig.1). In each cross-attention block, the molecule query (Q) computes attention weights over the respective keys and then updates itself with the corresponding values via a residual connection. The resulting molecule-genome and molecule-text context vectors are concatenated into a unified embedding that seamlessly combines molecular features with pathogen strain-specific insights. This fused representation is finally passed through task-specific MLP heads to predict antibiotic classification, minimum inhibitory concentration (MIC), and fractional inhibitory concentration index (FICI).

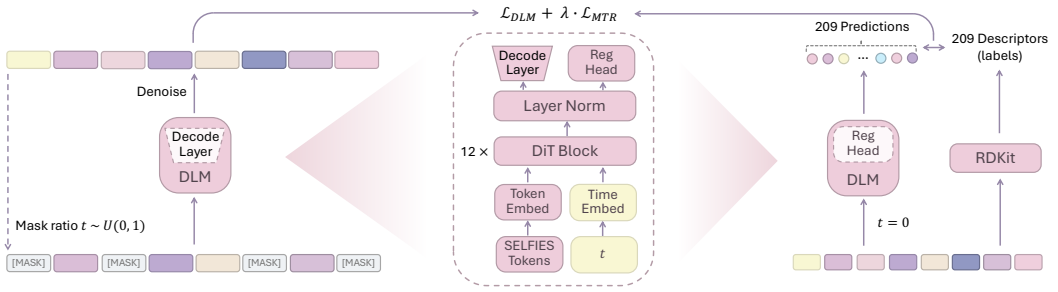

Figure 4: **DLM training regimen:** the model is trained not only to reconstruct molecules from scratch, but also to perform multi-target regression—learning high-quality latent embeddings that link specific token sequences to molecular properties. The 209 target descriptors were computed by RDKit.

C Data

Antimicrobial Peptides. The antimicrobial peptide (AMP) dataset comprises two components: (1) our in-house peptide collection and (2) the Database of Antimicrobial Activity and Structure of Peptides (DBAASP) [18]. From our in-house collection, we obtained 1,642 canonical linear peptides, covering 11 pathogen strains and 15,718 minimum inhibitory concentration (MIC) measurements. From DBAASP, we retrieved 16,408 peptides—71.5% of which contain noncanonical amino acids or modifications—alongside 5,630 pathogen strains and 105,547 MIC measurements. We then merged the two sources, excluding any strains with incorrect identifiers or names and any peptides containing noncanonical residues that were erroneous or lacked a valid SELFIES representation. After filtering, the consolidated dataset comprises 17,988 peptides, 5,632 pathogen strains, and 121,265 MIC measurements. All peptide sequences were converted to SELFIES strings using PepLink.

All MIC values originally reported in $\mu g/mL$ were converted to μmol. Values annotated with special operators (e.g., "$\leq$," "$>$") were handled according to the procedures summarized in Table 2. For model training, each MIC measurement was further transformed as $-\log_{10}(\frac{\text{MIC}}{10})$.

PepLink. We developed PepLink, a versatile converter that transforms amino acid sequences—including noncanonical residues, intrachain bonds, and terminal modifications—into SELFIES strings, and also supports reverse conversion from SELFIES back to amino acid sequences. It currently supports 404 distinct noncanonical amino acids, 11 intrachain-bond types, 242 N-terminal modifications, and 56 C-terminal modifications. The conversion proceeds in three steps: (1) concatenating the SMILES of canonical and noncanonical residues into a linear backbone; (2) introducing specified intrachain bonds to generate cyclic or bridged structures; and (3) appending the desired N- and C-terminal modifications.

Table 2: DBAASP MIC label pre-processing rules.

Operators	Transformations	Example
$>, >=, \geq$	Double	Record: $> V$, Transformed: $2 \times V$
$\gg$	Three times	Record: $\gg V$, Transformed: $3 \times V$
$- =>, - >=, - >=, - >, -$	Average	Record: $V_a - > V_b$, Transformed: $\frac{V_a+V_b}{2}$
$\pm$	Average	Record: $V_a \pm V_b$, Transformed: V_a
Others	Keep	Record: V_a, Transformed: V_a

All noncanonical residues were curated from DBAASP. We retrieved SMILES for the 20 canonical amino acids—and for many noncanonical ones—directly from PubChem. For those DBAASP entries lacking PubChem records, we used ChatGPT-o1 to standardize their DBAASP-provided IUPAC names, then converted these names to SMILES via OPSIN [26]. In total, this process yielded 404 unique noncanonical amino acids present in the DBAASP dataset.

Small Molecule Antibiotics. The small molecule antibiotic dataset comprises 49,331 *(molecule, strain)* pairs with binary classification labels of antibiotic activity against three different bacterial strains: *S. aureus* RN4220 (39,312 molecules from [13]), *E. coli* BW25113 (2,335 molecules from [23]), and *A. baumannii* ATCC 17978 (7,684 molecules from [14]).

Genome Data. Pathogen genome data were downloaded from the ATCC Genome Portal, except for the following four strains: *E. coli* MG1665, *E. coli* UMNK88, *P. aeruginosa* PA14, and *S. aureus* USA300. Since the genome information of these four strains is not recorded in ATCC, we instead obtained their data from NCBI.

Strain Description Data. Leveraging the powerful search and memory capabilities of modern large language models, we utilized Qwen2.5-Max [19] (version 01.25.2025) to search for important strain traits from reliable scientific publications. These traits include:

1. Species Information: Description of the species to which this strain belongs, including whether it is Gram-positive, Gram-negative, Fungi, Archaea, or Protozoa. Also, its notable physiological traits.
2. Unique Mutations: Any distinctive genetic mutations recorded to have impact on metabolic pathways or plasma membrane. And how these mutations modify bacterium's behavior and develop antimicrobial resistance.
3. Antibiotics or antimicrobial peptides resistance and sensitivity: Antibiotics and antimicrobial peptides to which this strain is known to be sensitive or resistant.

See Appendix.E for the prompt.

Molecules for diffusion language model training. We curated a diverse set of sequences from five sources: PubChem (downloaded via MolFormer [27]; 111,378,206 molecules), peptides from SmProt v2.0 (825,632 sequences), UniRef (<=50 residues; 6,972,866 sequences), UniProt (<=50 residues; 3,749,540 sequences), and modification-rich peptides generated by CycloPS from PeptideCLM [28] (10,000,000 sequences). We used IBM's SELFIES tokenizer [27] to parse these molecules. Any molecule having more than 1,024 tokens after tokenization was discarded, and duplicate entries between UniRef and UniProt were removed. After these filtering steps, we obtained 121.6 million molecules. We further used RDKit to calculate a 209-dimensional property vector for these molecules (Fig.1b), which quantifies a molecule's size, topology, electronic and surface properties, and functional group composition. We randomly withheld 1% of the data for evaluation and used the remaining 99% for training.

Synergy Data. All synergy data were curated from the DBAASP database, encompassed a broader range of compounds than solely AMPs. In total, our curated dataset comprised 2,732 unique molecule synergy-strain pairs, of which 88% were AMP-small molecule interactions and the remaining 12% were AMP-AMP interactions. Recognizing the challenges posed by the scarcity and inherent noise in synergy data, we binarized the FICI values using a 0.5 cutoff—labeling values below 0.5 as synergistic (1) and those equal to or above 0.5 as indifferent (0).

D Synergistic effect prediction with ApexOracle

Antibiotic synergy provides a powerful strategy to achieve more potent bacterial killing, reduce the emergence of resistance, and lower toxicity by enabling effective treatment at reduced drug doses. To explore pathogen strain-knowledge-aware antibiotic combination outcome prediction, thereby expanding ApexOracle's applicability, we concatenated the ApexOracle-fused molecular features of two molecules and fed the combined representation into a downstream neural network to predict antimicrobial synergy. This ApexOracle-based synergy model was trained and evaluated using a strain-wise, 3-fold cross-validation methodology, as detailed in Section 3.1.

The synergy training data, curated from the DBAASP database, encompassed a broader range of compounds than solely AMPs. In total, our curated dataset comprised 2,732 unique molecule synergy-strain pairs, of which 88% were AMP-small molecule interactions and the remaining 12% were AMP-AMP interactions. Recognizing the challenges posed by the scarcity and inherent noise in synergy data, we binarized the FICI values using a 0.5 cutoff—labeling values below 0.5 as synergistic (1) and those equal to or above 0.5 as indifferent (0)—and employed an ensemble strategy. By assembling 7 models, we achieved a mean 3-fold strain-wise AUROC of 0.7539 and AUPRC of 0.7454.

E Strain knowledge searching prompt

Strain knowledge searching prompt

messages=[
{'role': 'system', 'content': 'You are a scientist working on Antibiotic resistance.'},
{'role': 'user', 'content': f"""Your task is to provide a concise and informative description of the bacterial strain {strain_name}, including:

1. Species Information: Identify the species to which this strain belongs, specifying whether it is Gram-positive, Gram-negative, Fungi, Archaea, or Protozoa. Describe its notable physiological traits.
2. Unique Mutations: Describe any distinctive genetic mutations identified in this strain compared to the wild-type strain of the same species, particularly those affecting virulence factors, metabolic pathways, or the plasma membrane. Explain how these mutations modify the bacterium's behavior, physiology, or pathogenicity, and how they may contribute to the development of antimicrobial resistance.
3. Antibiotics and antimicrobial peptides Resistance: Outline any known antibiotics and antimicrobial peptides resistance mechanisms associated with this strain, including specific resistance genes or mutations. Describe the molecular mechanisms by which these confer resistance to particular antibiotics. Make sure to include the corresponding MIC value if you can find it. If you can't find corresponding MIC and molecular mechanisms, don't mention anything about that and keep your response as concise as you can.
4. Antibiotics and antimicrobial peptides Sensitivity: Identify antibiotics and antimicrobial peptides to which this strain is known to be sensitive. Explain the mechanisms by which these antibiotics exert their effects on the strain. Make sure to include the corresponding MIC value if you can find it. If you can't find corresponding MIC and molecular mechanisms, don't mention anything about that and keep your response as concise as you can.

In your response, do not provide a summary; instead, list the information as separate points as follows:

- Species Information: ...
- Unique Mutations: ...
- Antibiotic and antimicrobial peptides Resistance: ...
- Antibiotic and antimicrobial peptides Sensitivity: ...
- References: ... (This is just a place holder section, you MUST Ensure that the description is based on current scientific knowledge and has a reference but don't include this section in your response)

Important Instructions:
1. Ensure that the description is based on current scientific knowledge and includes relevant references where applicable, do not insert references before the Reference section.
2. Although the strain ID I provided is from ATCC, the same strain may also be cataloged under different identifiers in other databases, such as DSM, KCTC, NCTC, JCM, or other unique numbering systems beyond these examples. Please make sure to cross-reference these alternative ID systems when searching for relevant information.
3. Do not include any bulletin points in your response.
4. If information about the strain is unavailable or cannot be found, JUST respond with 'None' in the corresponding section! Do not respond with any further explanation!"""}],
extra_body={ "enable_search": True }