Tokens, Topologies, Taxa:
Towards Declarative Biology and Bioengineering

Thesis by
Zachary A Martinez

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2026
Defended May 21, 2026

Zachary A Martinez
ORCID: 0000-0002-7830-3162

ACKNOWLEDGEMENTS

To Dennis Mishler: I owe my start in scientific research to you. I still remember your patience and amusement on my first day of the Freshman Research Initiative (FRI), when I jammed the wrong-sized tip onto a micropipette. Although I was terrified of breaking lab equipment, you assured me everything was fine and efficiently removed the tip. After the Microbe-Hackers class and my first round of iGEM, I knew I was hooked on synthetic biology. Simply giving me the mentorship and resources to do research would have been more than enough, but you went above and beyond by training me as a Teaching Assistant and letting me mentor several undergraduate researchers at once on the same project. From writing research papers to planning and performing experiments, I learned an incredible amount during my time in Microbe-Hackers. You also gave me a head start on presenting science, both oral talks and posters, which has proven invaluable throughout graduate school.

To Sean Leonard: I have had the privilege of watching you, at least partially, "take off" in your scientific career, from your PhD days to becoming a Post-Doc. Your continued mentorship and friendship over the years has been key to my success as a scientist. When deciding what kind of research I wanted to pursue and what my scientific identity would be, I often think back to my awe at your ability to combine computational and experimental science into a sum greater than its parts. Your research philosophy, weaving together wet lab and dry lab work so that each makes the other stronger, continues to inspire how I approach my own science, and your flexibility and ingenuity set a standard I still aspire to.

To Howard Ochman and Nancy Moran: Thank you for giving me the opportunity to continue pursuing my scientific career. Y'all not only gave me my first full-time job, but made me feel like part of the scientific community by including me in group meetings and journal clubs. It was such a privilege to learn from scientists of your caliber so early in my career. I learned so much about biology and truly fell in love with the field during my time as a lab technician. Most of all, y'all validated that I was indeed a scientist, despite my own reservations, and encouraged me not just to apply to graduate school, but to aim for Caltech, something I would never have done without your support.

To Paul Kirchberger: I owe a large chunk of my PhD success to you specifically. My love of phages, and viruses in general, comes straight from you. You opened my

eyes to the infinite sea of virology. Although I had done scientific research before, you were my immediate supervisor for my first full-time scientific job, and I could not have asked for a better mentor. From letting me design my own experiments to encouraging me to develop new software for our research, my time working with you firmly cemented my desire to pursue science as a career. I've learned so much from your mentorship style, and I continue to draw on it when mentoring my own students. Your encouragement and critiques of my first foray into deep learning, when I was trying to generate divergent microvirus capsids, directly inspired me to continue this line of research and ultimately led to the creation of TRILL, which makes up two-thirds of my PhD. Beyond the science, I've cherished bonding with you over our shared love of horror movies.

To Matt Thomson: Although I had no idea who you were when applying to Caltech, I felt truly blessed to be paired with you during interviews. We hit it off talking about kombucha, my first research experience, and just waxing philosophical about science. After hearing you speak during the summer bootcamp, I became deeply interested in your research and ideas. I've always worried about pursuing research that was too niche, or spending too long in a single area, since I find so many fields fascinating. You showed me that with the right people, you can run a lab spanning many scientific interests, something both encouraging and intimidating, and in doing so you gave me permission to be a multi-interest scientist myself. Your physics and quantitative background profoundly shaped my research and trajectory. One of the biggest impacts you've had on me was the freedom to conduct research independently and propose my own projects, which gave me the experience and scientific maturity that ultimately grew into this PhD. I'm also grateful for your guidance on crafting impactful presentations, advice I return to again and again.

To Richard Murray: The scientific lessons I've learned over the years in your lab will forever be ingrained in me. Although I had synthetic biology experience before, your engineering rigor was entirely novel to me, and deeply moving. Your comparisons between mature engineering disciplines and bioengineering have been eye-opening and have profoundly shaped my scientific trajectory. I've always dreamt of synthetic cells, and you gave me the opportunity to work in this field and even to mentor two undergraduates pursuing two different "crazy" ideas of mine. The fact that you've expanded your expertise to the point of running two essentially independent labs blows my mind to this day, and your work ethic and meticulousness have been an inspiration to learn from. I'm especially grateful for how well you

and Matt worked together, both personally and scientifically. Your dual mentorship let me feel supported and independent all at once, working at the intersection of bioengineering and AI.

To Justin Bois: You have been one of my biggest influences at Caltech. From the very first class I took in graduate school, 103a, I was immediately hooked on your philosophy and approach to quantitative science, especially in programming contexts. I like to think of getting "better" at science as collecting more tools for a toolbelt, and I have gained so many from you alone. Even as I struggled with imposter syndrome when I first arrived at Caltech, you surprised me with the chance to be a teaching assistant alongside you, finally giving me a sense of belonging. From data visualization packages to cowboy boots to metal music to ADHD, our conversations have always been a treat and made Caltech feel like home.

To the rest of my committee, Steve Mayo, Sarkis Mazmanian, and Kaihang Wang: your combined expertise has greatly shaped both my thesis and my career trajectory. Steve, who served on my candidacy committee, taught me an immense amount about proteins, something I was woefully ignorant about before graduate school. Sarkis, your biological insight was essential to my microbiome-related research, giving me a grounding in biology that I leaned on throughout. Kaihang, thank you for supporting my experimental plans with TRILL and for offering me a Post-Doc position in your lab. Thank y'all for the support and the opportunity to learn.

To Manisha Kapasiawala: Thank you for all your help whenever I bugged you for things, from using the Echo to simply helping me calm down and encouraging me to keep pushing through graduate school. You always went out of your way to include me and make me feel welcome and part of the lab, which is no small feat given my anxiety. I cannot wait to see the changes you make in the world. Our country needs more people like you.

To Zoila Jurado: You have been such a supportive peer during graduate school, from helping me troubleshoot cell-free reactions to always lending an ear for both scientific and personal problems. With you in the lab, I always felt cared for and listened to. Of all the departures I've experienced in science, yours left the biggest hole.

To Yan Zhang: I have the privilege of calling you not just a friend and co-mentor, but a collaborator as well. All of our phage ideas kept me passionate about science

during my PhD and helped prevent my burnout. From your generous presentations on navigating the job market to your cell-free expertise, I have learned a wealth of knowledge from you. I'm also grateful for your advocacy, both for TRILL and for me as a scientist.

To Miki Yun: You have been not just a perfect lab manager, but a wonderful friend as well. A lab needs to be a well-oiled machine to run successfully, and you are the oil. I will always treasure our impromptu conversations, and I hope to continue them as I stick around Caltech a while longer.

To Arjuna Subramanian: You have been the biggest user of TRILL and one of my biggest supporters since the day I joined the Thomson lab. I've benefitted greatly from your generous scientific insights, but more than that, your friendship has been another key to my PhD. You can converse about an incredible range of topics, from serious research to comical current events, and I could talk to you for hours. Your patience with buggy TRILL and your tips for improvements have been critical to the platform.

To Alec Lourenço: You are one of the few people I've met who truly "gets" my kind of science, and it has been such a privilege to brainstorm and collaborate with you. Another big user of TRILL, you have helped me find countless bugs and advocated not just for the platform, but for me as well. You also gave me the opportunity to help out with the IEA iGEM team, something that was incredibly fulfilling and informative for designing TRILL.

To My Parents: Put simply, and disregarding literal biology, I would not be here today without y'all. I am blessed to have parents with unconditional love and support. Even when I felt like I couldn't finish my undergraduate degree, much less my PhD, y'all have always been my biggest fans and supporters. Your constant love, sacrifices, and lessons have made me the man I am. From Dad's ingenuity and creativity to Mom's drive and diligence, to the compassion you both share, these are some of the traits that laid the foundation for my success. The FaceTimes, the early-morning phone calls to keep me awake, the sacrifice of flying out to visit several times a year, and even taking a programming bootcamp at Caltech are just the tip of the iceberg when it comes to all y'all do for me. I love y'all with all my heart.

To Allison Martinez: The love of my life and my best friend. Being able to grow up alongside you has been such a blessing. Even though I've already mentioned so many others who were critically important to my PhD, they all pale in comparison to

the impact you've had. While most people never see me at my lowest, you are always there to pick me up and say the exact words I need to hear to keep going. You moved across the country for me, and as surprising as it may sound, that was probably easier than the daily support you so tirelessly and selflessly give. I owe my happiness and sanity, and therefore my success, to you.

ABSTRACT

This thesis spans a software platform for deep-learning based protein analysis and design, a body of exemplary tasks run on top of it, and a predicted community-scale structural proteome of a defined gut microbiome. The argument underneath all three is that for experimentalists who use rather than build contemporary deep-learning methods, the cost of composing them now outpaces the cost of running them, and that a declarative interface layer is the appropriate response.

TRILL aspires to jump the technical hurdle facing many researchers today when attempting to use state-of-the-art artificial intelligence. It is open-source, runs locally, and wraps contemporary protein, nucleic-acid, and small-molecule deep-learning models behind a uniform vocabulary of thirteen top-level commands. The platform's value is not the count of integrated models but the property that one model can be swapped for another with a one-argument change rather than a pipeline rewrite, and that fast learned screens are paired with physics-based validation at the points where overconfidence costs most.

Eight protein language models are benchmarked on biophysical regression tasks under joint compression sweeps, where representations are routinely bloated by one to two orders of magnitude. Classifiers for cellulase, antimicrobial, and toxin activity, trained on a few thousand examples each, are applied to a unified scan of more than two hundred million proteins from the NCBI non-redundant catalogue. An end-to-end pipeline takes seventeen predicted toxins of unknown function through structure prediction, binder design, and molecular dynamics on nearly nine hundred designed complexes. In the same vein of biosecurity, a multi-method screening cascade detects toxin mimetics designed by generative AI that classical sequence-alignment screening misses by construction.

hCom2, a defined synthetic gut consortium, is the substrate for the third contribution. Roughly four hundred thousand atomic-resolution structures of its proteome were predicted with a sequence-only structure predictor, segmented into roughly eight hundred thousand structural domains, and assigned to a curated structural-classification hierarchy. A worked case study uses the resource as a query target and pulls out nineteen commensal carriers of the *Helicobacter pylori* virulence-factor TIPalpha fold across fourteen strains where sequence-only annotation comes up empty. Techniques such as co-folding and signal-peptide prediction are then used to attempt to

identify the putative immunomodulatory mechanism.

In summary, this work attempts to apply deep-learning in a declarative fashion, for both basic fundamental biology as well as applied bioengineering. The TRILL platform aims to be the "middle-ground" between bleeding-edge AI and experiments, allowing end-users with limited coding background to design and orchestrate bespoke workflows, without having to sacrifice creativity by getting lost in the technical-weeds.

PUBLISHED CONTENT AND CONTRIBUTIONS

Martinez, Zachary A, Joseph C. Boktor, Sarkis K. Mazmanian, and Matt W. Thomson (2026). "A community-scale structural proteome of the synthetic gut microbiome hCom2 enables the identification of structural homologs of the *H. pylori* TIPalpha fold".

Z.A.M. designed and built the computational pipeline and wrote the manuscript. Chapter 4 includes content based on this article. In preparation.

Martinez, Zachary A., Richard M. Murray, and Matt W. Thomson (Oct. 27, 2023). *TRILL: Orchestrating Modular Deep-Learning Workflows for Democratized, Scalable Protein Analysis and Engineering*. Preprint, bioRxiv. DOI: 10.1101/2023.10.24.563881.

Z.A.M. conceived, designed, and implemented the TRILL framework, ran all experiments, and wrote the manuscript. Chapters 2–3 include content from this article and provide updates.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Tools versus pipelines

Computational biology has been reshaped by the parallel growth of public biological data and the deep-learning methods built to operate on it. Public sequence, structure, and functional archives have expanded by roughly an order of magnitude at every level of biological organization, and the repertoire of foundation models that consume those archives has proliferated across modalities and tasks. The less visible consequence is that the engineering cost of composing those methods into reproducible pipelines has become a significant time cost of asking a biological question with them. This thesis treats that cost as the actionable problem and argues that the appropriate response is a declarative interface layer at which a biologist specifies what they want computed, not how to compute it, built across the modalities and tasks that contemporary methods cover.

The bottleneck is no longer the question "is there a method for this?". For many well-posed biological questions there are now several. To answer a question as deliberately simple as "given these unfamiliar bacterial sequences, which look like they bind a small molecule, and what does the binding pose look like?", a question that not long ago would have required either a sequence-homology lookup against a sparsely annotated database, a biologist today must locate, install, and orchestrate analyses across several packages, each with its own input format, CUDA versions, license, biological assumptions, and other subtle differences. State-of-the-art models are published constantly, but the end users who would benefit often cannot install them or have the computational resources available, let alone use them confidently. Some groups chase spots on benchmarking leaderboards, drawing attention towards headline-results that are often not generalizable, while impactful methods go overlooked.

This thesis argues that the composition tax is a major hurdle for further adoption of cutting-edge methods, and that it is best addressed at the abstraction layer rather than one pipeline at a time. Chapter 2 introduces TRILL, first described in a 2023 bioRxiv preprint (Martinez, Murray, and Thomson, 2023), an open-source command-line tool that brings many contemporary protein, nucleic-acid, and small-molecule model

families behind a single declarative interface. The user writes what they want done (`trill embed proteins.fasta`, `trill fold proteins.fasta`, `trill dock receptor.pdb ligands.csv`), and the framework handles technical minutiae like tokenization, VRAM utilization, model sharding, mixed precision, checkpoint caching, structural format conversion, and safe model deserialization. Section 1.3 develops the declarative-interface argument explicitly and describes the TRILL platform itself. Chapter 3 puts the framework to use across protein discovery and design, including a tree-of-life-scale screen for toxins and the subsequent generation of antitoxins that follows a TRILL pipeline from classifier training and evaluation, NR-scale screening, through ESMFold prediction, binder design, docking, and short molecular dynamic simulations. Chapter 4 leverages TRILL and other tools at a community scale, building a structural proteome of hCom2, a 120-strain defined human gut microbiome (Cheng et al., 2022), illustrated with a worked case study of the *Helicobacter pylori* virulence factor TIPalpha and its structural homologs in commensal gut microbes.

The title of this thesis, *Tokens, Topologies, Taxa*, names three units of biological information at three scales of granularity. *Tokens* are the per-residue or per-position representations that protein, nucleic-acid, and small-molecule models share. These discrete units are part of what Chapter 2 builds on and Chapter 3 deploys. *Topologies* are predicted three-dimensional folds, the unit of structural information that Chapter 4 generates at scale and that Section 4.5 queries against in the TIPalpha case study. *Taxa* describes the organisms a synthetic community is composed of, the unit at which hCom2 phenotypes are investigated. The thesis runs the chain from tokens through topologies into taxa, and argues that a declarative framework is what makes the chain tractable for an experimentalist with limited programming skills rather than only for a specialized computational researcher.

1.2 A brief account of the methods landscape

Data has outgrown annotation

A short tour of the public repositories makes the scale concrete. The Sequence Read Archive and its INSDC counterparts hold on the order of sixty petabase-pairs of publicly accessible reads across more than half a million studies (Katz et al., 2022; Karasikov et al., 2025). The non-redundant protein database (NCBI nr), a pooled catalogue assembled from GenBank CDS translations, RefSeq, PDB, Swiss-Prot, and related deposits, has continued to double roughly every two years and by late 2025 was approaching a billion sequences. Metagenome-derived catalogues now

dwarf this curated pool. MGnify, EMBL-EBI's microbiome resource, exposes more than 2.4 billion predicted proteins clustered at 90% identity, drawn from publicly archived environmental and host-associated assemblies (Richardson et al., 2023). At the organism level the Genome Taxonomy Database organises over 900,000 bacterial and archaeal genomes, most of them metagenome-assembled and uncultured, into roughly 190,000 species clusters (Parks et al., 2026). Functional readouts have followed. NCBI's Gene Expression Omnibus archives more than 200,000 studies and 6.5 million samples of transcriptomic and epigenomic data (Clough et al., 2024).

Resolved structure has not kept pace. The Protein Data Bank, the canonical archive of experimentally determined macromolecular structures, contained roughly 240,000 entries by mid-2025, with more than 20,000 deposited in that year alone (Burley et al., 2025). That is fast by historical standards but still four orders of magnitude short of the protein catalogues described above, and the gap widens every year. The structures that most computational pipelines now consume are therefore predictions rather than measurements. The AlphaFold Database (AFDB) released its initial set of predicted structures in 2021 and now hosts more than two hundred million entries derived primarily from UniProt reference proteomes, with clustering studies indicating that structural space at this scale is substantially less crowded than sequence space alone implies (Barrio-Hernandez et al., 2023). Systematic domain-level dissection of the AFDB by The Encyclopedia of Domains (TED) has carved this corpus into 365 million domains spanning more than a million taxa and surfaced roughly 7,400 candidate novel folds (Lau et al., 2024). Mapping those into the CATH (Class, Architecture, Topology, Homologous superfamily) classification expanded the fold count from 1,349 to 2,078, raised superfamilies from 5,841 to 6,573, and lifted architectures from 41 to 77, though the curators are careful to note that new entries derived from deep-learning predictions remain provisional until checked against experiments (Waman et al., 2025).

The ESM Metagenomic Atlas complements all of this with more than 700 million ESMFold-predicted structures of MGnify-derived metagenomic proteins, covering swathes of sequence space underrepresented in AlphaFold's UniProt-anchored release (Lin et al., 2023). Conformational ensembles are beginning to follow. mdCATH provides standardised all-atom molecular-dynamics trajectories across 5,398 CATH domains, run as five replicas at each of five temperatures and amounting to roughly 62 ms of cumulative trajectory (Mirarchi, Giorgino, and De Fabritiis, 2024).

From string matching to learned representations

The dominant computational primitive of pre-deep-learning bioinformatics was the pairwise sequence alignment. BLAST (Altschul et al., 1990) made it cheap, and profile hidden Markov models, especially as implemented in HMMER (Eddy, 2011), extended it to families. The Pfam catalogue (now part of InterPro) and the SCOP (Structural Classification Of Proteins) and CATH structural-family hierarchies were built on top of this. The strength of the alignment-based methods is that two sequences carry enough information about each other to be compared, and with care, one can use metrics such as the bitscore, E-value, percent identity, and query/subject coverage to quantify the significance of a match. Coverage statistics complement these measures by reporting the fraction of each sequence that participates in the alignment, guarding against high-scoring but locally restricted matches that may not reflect true homology over the full length of either sequence. Its weakness is that two sequences may share common ancestry without recognizable sequence similarity. The *twilight zone* between approximately twenty and thirty-five percent identity is where alignment-based homology calls become unreliable (Rost, 1999), and structure is conserved several times longer than sequence (Illergård, Ardell, and Elofsson, 2009), so the alignment-based method systematically underestimates the size of biologically meaningful protein families.

Two innovations of the deep-learning era address these limitations. Early efforts paired the abundance of unannotated sequence data with recurrent neural networks. UniRep, trained as a multiplicative LSTM on roughly 24 million UniRef50 sequences with a simple next-amino-acid objective, was a representative landmark of this period, recovering structural, evolutionary, and biophysical features from sequence alone, performing competitively with state-of-the-art methods on stability and function benchmarks, and yielding roughly two orders of magnitude in efficiency on a directed-evolution task (Alley et al., 2019). Recurrent architectures, however, propagate information sequentially, which forces long-range dependencies to flow through many intermediate steps and limits the model's ability to attend directly to distant context. The transformer, introduced in 2017, replaced recurrence with self-attention, computing pairwise interactions between every pair of positions in a sequence in parallel (Vaswani et al., 2017). For proteins this turned out to be the right inductive bias. A linear string of words mostly carries its meaning in local context, where neighbouring tokens dominate, but a folded protein is a three-dimensional object in which residues sitting at opposite ends of the primary sequence may be in direct physical contact, hydrogen-bonding to each other or coordinating the

same active-site. Self-attention naturally accommodates this geometry by attending to every residue pair. ESM-1 demonstrated the consequences at scale, training transformers up to roughly 650 million parameters on 250 million UniRef sequences with a masked-language-modelling objective and recovering secondary structure, residue-residue contacts, mutational effects, and remote homology directly from the learned representations (Rives et al., 2021). Crucially, none of this required labels. Self-supervised pretraining lets these models consume the deluge of raw protein sequences that public archives have accumulated faster than any annotation pipeline can keep up with, allowing them to learn what is fairly described as the language of life.

A practical consequence of the embedding-based representation is that protein comparison is no longer constrained to pairwise dynamic-programming alignment scaling with the product of the two sequence lengths. Per-protein embeddings reduce comparison to a vector operation, scored using techniques such as cosine similarity, potentially indexable by approximate-nearest-neighbor data structures, and scanned across whole databases at speeds that pairwise alignment cannot match. knnProtT5 uses ProtT5 embeddings directly with a k-nearest-neighbor search to recover hits below the twilight-zone identity threshold where alignment loses its signal (Schütze et al., 2022). pLM-BLAST extends the same idea to local alignment by treating per-residue embeddings as a context-dependent substitution matrix (Kaminski et al., 2023), and the broader family of embedding-based annotation-transfer and similarity-search methods reviewed by Leclercq and Droit (2026) sits in the same regime.

The second innovation is the end-to-end structure predictor. AlphaFold2 combined MSA-derived coevolution features with a differentiable structure module, achieving state-of-the-art performance, an achievement recognized by the 2024 Nobel Prize in Chemistry (Jumper et al., 2021). The "folding" problem is not solved, since dynamics, intrinsic disorder, and multimeric assemblies remain difficult, but for well-behaved single chains, prediction is now an everyday capability rather than a research milestone. ESMFold showed that an ESM2-scale language model can substitute for the MSA at a quantifiable but acceptable per-protein cost. A recent post-2022 PDB benchmark places ESMFold at 76 percent and AlphaFold2 at 88 percent on monomeric targets (Mahtha, Venkadesan, and Mohanty, 2026), and the trade is roughly an order-of-magnitude speedup, which is what makes proteome-scale folding routine. The next wave of structure predictors has since pushed the technique

beyond the single-chain backbone. AlphaFold3 replaced the AlphaFold2 Evoformer with a diffusion-based architecture that predicts the joint structure of complexes containing proteins, nucleic acids, small molecules, and modified residues in a single inference, collapsing what used to be a structure-then-dock pipeline into one step (Abramson et al., 2024). Boltz-1 was the first openly released cofolding model with comparable scope to AlphaFold3 (Wohlwend et al., 2024), and Boltz-2 extended it with a quantitative binding-affinity head that produces a pIC_{50} estimate alongside the predicted complex, approaching the accuracy of free-energy-perturbation calculations several orders of magnitude faster than the physics-based reference (Passaro et al., 2025).

Where the gap lies

Each of these models is, individually, an artifact of substantial engineering effort and cost. Each is usually distributed as a research codebase, typically a GitHub repository, sometimes a HuggingFace checkpoint, with its own preferred environment, its own input format, and its own assumptions about whether the user has access to one GPU, many GPUs, or none. The gap in the landscape is not at the level of individual models. The gap is at the level of composition. There has not been a tooling layer that brings these models behind a single uniform interface, exposes them to a non-specialist user as a coherent command-line vocabulary, and handles the substrate concerns (distributed execution, mixed precision, format conversion, safe deserialization, model caching) once rather than thirty times. Bio_embeddings (Dallago et al., 2021b) was an early step in this direction for the embedding subset of the problem. ColabFold (Mirdita et al., 2022) is the canonical example for structure prediction. Several 2025–2026 entrants extend the pattern into adjacent task families. Ovo (Prihoda et al., 2025), a Nextflow-based de novo design ecosystem developed with both command-line and graphical front-ends and a dedicated quality-control module for designed proteins, focuses on the design subproblem. evedesign (Hopf et al., 2026) is a unified biosequence-design framework exposed primarily through a web interface. ProteinMCP (Xu et al., 2026) takes an agentic-AI route, using the Model Context Protocol to expose roughly thirty-eight tools to a language-model orchestrator. Each occupies a slightly different point in the design space, and each postdates TRILL's first description on bioRxiv in 2023 (Martinez, Murray, and Thomson, 2023). None of these integrates representation, generation, prediction, docking, and simulation across protein, nucleic acid, and small molecule under one CLI. The position taken in Chapter 2 is that this

integrated declarative layer is the right unit of intervention, and that the operational test of the proposition is whether the same framework can drive both single-question applications (Chapter 3) and a community-scale structural proteomics resource (Chapter 4).

1.3 The declarative move

The argument so far is that the methods landscape has expanded faster than the engineering layer that connects methods to working biologists, and that the appropriate response is a declarative interface.

What a declarative interface is

The distinction between *imperative* and *declarative* interfaces is something most life-scientists are not aware of. An imperative interface specifies a sequence of operations (open file, allocate buffer, decode tensor, move to GPU, run forward pass, write output). A declarative interface specifies a goal, like "compute ESM2 embeddings for this FASTA file." The system handles the operational sequence. The general advantage of declarative interfaces is that the same goal description survives changes in implementation. A high-level language like Python expresses `for residue in sequence` and lets the interpreter handle memory allocation, pointer arithmetic, and checks for syntactical errors. The same loop written in C or assembly forces the programmer to manage those details by hand and potentially rewrite the program if the hardware changes.

Pipeline-level analogues exist for computational biology. Snakemake and Nextflow are workflow managers that ask the user to declare input-output relationships between pipeline steps, with the execution graph and scheduling derived automatically. Their adoption in bioinformatics over the last decade demonstrates that the workflow community has already decided that the declarative pattern is worth the up-front conceptual cost. While TRILL does not offer fault-tolerant jobs, automatic slurm handling, and other quality of life features that these mature workflow managers offer, this thesis argues that TRILL provides a declarative interface one level deeper, at the model-invocation layer rather than the step-graph layer. The biologist writes `trill embed esm2 proteins.fasta`. The framework handles tokenizer selection, batch padding, device placement, mixed precision, multi-GPU sharding, and output serialization. When ESM2 is replaced by Ankh, the user changes one argument and the rest of the pipeline is unchanged. When an embedding-consuming step downstream is replaced by a different classifier, the embedding step does not move.

The skill threshold for running a contemporary deep-learning pipeline drops from "competent ML engineer" to "competent life-scientist who can read a CLI manual."

Declarative bioengineering

Two features of bioengineering make a declarative interface more valuable here than in many adjacent fields. The first is the asymmetry of expertise. Domain experts in related areas of study (molecular biologists, biochemists, microbial ecologists) typically know what they want to ask in precise scientific terms, but the operational cost of asking it has historically required ML/coding skills that sit in a different department. The second is the rapid turnover of state-of-the-art models. The ESM family went from ESM-1 to ESM-2 to ESM-3 in roughly four years and new, alternative methods appear often.

Lowering the barrier to running these models, and exposing them through a vocabulary in which any model can be substituted for any other compatible model, gives the end user a kind of creative freedom that imperative pipelines cannot offer. A biologist can assemble a bespoke pipeline for a question no one has asked before, mixing a structure predictor from one group with an inverse-folding model from another and a thermostability predictor from a third, or fall back on a tried-and-true workflow that has already been validated by others. The argument is not that every researcher will want to build custom pipelines, but that the choice of whether to build one or to reach for an off-the-shelf workflow should belong to the scientist asking the question rather than to the engineering layer above them. Subramanian et al. (2023) exemplifies the creative spirit of TRILL. By composing existing TRILL commands, they developed Foldtuning, an iterative algorithm for generating biomimetic proteins with progressively decreasing sequence similarity to the inputs, a use case not originally envisioned by TRILL's author. Foldtuning has since been integrated directly into TRILL, allowing end users to invoke it through a single command rather than manually chaining together the workflow described in the original paper.

Overconfident artificial intelligence

O'Shea-Wheller and Murray (2026) argue that the field has a *transferability crisis*. Published model benchmarks routinely fail to predict deployment performance, because benchmark test sets do not capture the full variance of deployment data. The implication for tool builders is that flashy benchmark numbers cannot substitute for hands-on testing on the user's specific data, and that infrastructure which lowers the cost of running models on the user's own data, not on someone else's benchmark,

is what produces useful biological insight. This critique names a failure mode this thesis is trying to avoid.

Furthermore, deep-learning models are efficient but can sometimes be confidently wrong. Structure-based searches built on predicted folds retrieve larger candidate-homolog lists than sequence-based methods at the cost of a higher false-positive fraction (Mutti, Ocaña-Pallarès, and Gabaldón, 2025), and predicted-confidence scores like predicted Local Distance Difference Tests (pLDDT) have been shown to misalign with experimental accuracy in regimes including conformational-switching proteins and de novo designs that express but fail to fold into their intended structures. TRILL responds to this asymmetry by empowering users to pair fast deep-learning screens with physics-based downstream checks, rather than trusting a model verdict alone or paying the throughput cost of running force-field-based methods end-to-end.

What TRILL is

TRILL is a Python-based command-line platform, installable from GitHub, that as of version 1.10.0 exposes thirteen top-level commands (Table 2.1, Chapter 2) spanning embedding, fine-tuning, generation, inverse folding, diffusion-based backbone design, structure prediction, docking, classification, regression, molecular dynamics, zero-shot scoring, and an end-to-end workflow. The full payload covers more than thirty model integrations across three input modalities (protein, nucleic acid, and small-molecule SMILES). Under the hood, each command is a Python module that registers itself with a top-level argument parser, and in cases where the underlying model is a transformer, wraps the model as a PyTorch Lightning `LightningModule`. The framework handles distributed execution through Lightning, Accelerate and DeepSpeed ZeRO, reproducibility through full-state RNG seeding and timestamped log files, and model caching through a persistent on-disk cache. Standardized I/O formats make the output of one TRILL command usable as the input of another without glue code. A related tool, Bio_embeddings (Dallago et al., 2021b), integrated the embedding subset of the problem. ColabFold (Mirdita et al., 2022) and ColabDesign are a focused tools for protein structure prediction and design. HuggingFace `transformers` (Wolf et al., 2020) is the canonical declarative interface for modern transformers, but still requires manual coding. TRILL's wager is that the same declarative discipline can extend across representation, generation, prediction, docking, simulation, and zero-shot scoring, across protein, nucleic acid, and small-molecule modalities, under one CLI vocabulary.

1.4 Community-scale biology and the structural proteomics of defined microbiomes

The fourth chapter of this thesis moves from individual proteins to entire microbial community proteomes.

The gut microbiome warrants scientific focus

Gut microbes participate causally in host physiology such as immune development and metabolism. Mazmanian et al. (2005) demonstrated that polysaccharide A from *Bacteroides fragilis* directs maturation of the host immune system in gnotobiotic mice. Round and Mazmanian (2010) extended that line by showing that *B. fragilis* induces $Foxp3^{+}$ regulatory T cells through TLR2 signaling, providing direct protection and even curing experimentally induced colitis in animals. Both of these examples share a structural feature relevant to the present thesis. A defined microbial molecule, attached to a defined microbial species, drives a defined host phenotype. The molecule, the species, and the phenotype are all in principle resolvable, but in practice the species is usually known long before the molecule is.

The dominant experimental scaffolds for going from species to molecule have changed over the past decade. Stool-derived fecal microbiota transplantation in gnotobiotic mice gives community-level signal but can obscure strain-level contributions. Monocolonization gives strain attribution but cannot capture community interactions. Defined synthetic communities, fully sequenced and reconstitutable, attempt to bridge the gap. Cheng et al. (2022) describe the human Commensal Community version 2 (hCom2), a 119-strain defined consortium constructed by augmenting an earlier version with strains identified by an in-vivo competition assay as resilient colonizers. The hCom2 community is therefore a natural unit of analysis for protein-level interrogation of the human gut microbiome.

Structures allow for deep evolutionary and functional comparisons of proteins

Sequence-level catalogues of gut microbial proteins exist (Tierney et al., 2019), but several biologically important questions are awkward to ask at the sequence level. Remote homology between gut commensal proteins and characterized virulence factors, immunomodulators, or toxin scaffolds often sits below the twilight zone described in Section 1.2, where alignment-based homology calls become unreliable. Fold-level recognition of a known scaffold in an unannotated commensal protein is therefore a structural question, not a sequence question, and the Foldseek (Van Kempen et al., 2024) and DALI (Holm, 2020; Holm, 2022) generations of structural-

search tools have made the question tractable. The AlphaFold Database expansion documented by Fleming et al. (2025), the structural clustering work of Barrio-Hernandez et al. (2023), and the CATH and ECOD mappings to AFDB at scale (Bordin et al., 2023; Lau et al., 2024; Schaeffer et al., 2024; Waman et al., 2025) have made fold-resolved catalogues of structural space increasingly comprehensive.

Chapter 4 builds the hCom2 structural proteome, more than 400,000 ESMFold-predicted structures across the 120 strains, quality-analyzed along four independent axes (pTM, pLDDT, MolProbity, Z-DOPE) (Eramian et al., 2008; Williams et al., 2018; Lin et al., 2023), and CATH-annotated via Merizo-search (Lau, Kandathil, and Jones, 2023). The TIPalpha case study in Section 4.5 is a worked example. A structural query against the *Helicobacter pylori* virulence factor TIPalpha (Tsuge et al., 2009; Suganuma et al., 2008; Gao et al., 2012) recovers nineteen hCom2 carriers of CATH fold 3.10.129.140 across fourteen gut commensal strains. The relevant comparator at metagenome scale, Liu et al. (2026), provides a structure-aware gut catalogue of 2.7 million proteins across 1,255 phage and 968 bacterial genomes. The contribution of Chapter 4 is complementary rather than competitive, offering strain-resolved structures attached to a defined consortium that can be experimentally manipulated rather than a metagenome-scale catalogue whose individual entries cannot be added to or removed from a mouse on demand.

1.5 Summary of contributions

This thesis contributes three artifacts and one argument. The argument is that the engineering cost of composing contemporary deep-learning methods has become the rate-limiting step in their biological application, and that the appropriate response is a declarative interface layer that hides operational detail without sacrificing methodological flexibility. The first artifact is TRILL itself, the open-source declarative platform described in Chapter 2 that spans embedding, generation, prediction, docking, simulation, and scoring across protein, nucleic-acid, and small-molecule modalities. The second is a set of TRILL-driven discovery and design pipelines in Chapter 3, including a tree-of-life-scale toxin screen and the structure-conditioned generation of candidate antitoxins. The third is the hCom2 structural proteome of Chapter 4, including more than 400,000 ESMFold-predicted and CATH-classified structures, together with a worked TIPalpha case study showing how the resource can be leveraged.

Chapter 2

TRILL: TOWARDS DECLARATIVE BIOENGINEERING

2.1 Introduction

Even though the first genome was sequenced in 1977 (Sanger et al., 1977), in less than 50 years there are now more than 36 petabytes of sequencing data on the NCBI Sequence Read Archive (Katz et al., 2022). This scientific feat has enabled scientists to utilize the breadth of nature for their research, whether it be for revealing unculturable organisms that live in extreme environments (Venter et al., 2004), discovering novel antimicrobial peptides (Torres et al., 2021), or finding new ways to engineer genomes (Gao et al., 2017). While there are seemingly endless amounts of sequencing data, actually leveraging this treasure-trove of biological novelty is a non-trivial task. Classical methods for comparing biological sequences usually involve pairwise comparisons (Altschul et al., 1990) or by using hidden Markov models (HMMs) (Eddy, 2011). While these methods rely on evolutionary relationships to link related sequences through homology, machine learning based methods have shown success for functional comparisons without needing shared ancestry. Researchers have, for example, been able to predict whether a given protein is a cell-penetrating peptide, regardless of actual homology (Yadahalli and Verma, 2020). These predictions were enabled by extracting amino acid frequencies and biochemical properties for each protein and using this data to train random-forest classifiers. However, these methods tend to rely heavily on feature selection, which not only requires background knowledge but also introduces potential bias by focusing on select features.

Deep learning methods, such as recurrent neural networks, have proven to be able to extract functional information of proteins from sequence alone (Alley et al., 2019). However, the application of natural language processing (NLP) methods to protein sequences has truly led to breakthroughs for state-of-the-art in-silico protein analysis. At the heart of these breakthroughs is the Transformer, a deep-learning architecture first published in 2017, which uses self-attention to effectively capture long-range dependencies and intricate patterns in protein sequences that were previously challenging for traditional methods to discern (Vaswani et al., 2017). Analogous to using words and sentences to train typical large language models

(LLMs), Transformer-based models such as ESM-2 use amino acids and protein sequences to learn the "grammar" of life. These protein language models (PLMs) learn in a self-supervised manner, where the model attempts to predict the identity of a random 15% of the amino acids per sequence using the unmasked portions. ESM-2 was pre-trained on the masked language modeling task with 65 million unique protein sequences from UniRef (Lin et al., 2023). After this extensive training, scientists are able to use these pre-trained models to extract high-dimensional representations for their proteins of interest. These vectors can then be used for clustering, regression, classification, and other downstream tasks for functional comparisons (Bernhofer and Rost, 2022). However useful deep-learning based approaches are, large PLMs are unwieldy, with ESM-2 parameter sizes ranging from millions to billions of parameters, much too big for many GPUs. The sheer size of these types of models limits widespread adoption in the biological community, where most end-users are not able to efficiently wield these powerful tools due to hardware constraints. Furthermore, end-to-end analyses for proteins often require chaining together methods, including other types of models such as denoising diffusion probabilistic models and graph-based message passing neural networks.

Despite the rapid proliferation of powerful individual methods, the field faces a persistent gap between the tools that exist and the people who could use them most productively. Of the more than 240 million protein sequences in UniProt, fewer than 0.3% have experimentally validated functional annotations, representing an enormous opportunity for computational methods to guide discovery. Yet the researchers best positioned to apply these tools to real-world problems are often unable to use them. The overhead of configuring computational environments, resolving dependency conflicts, writing boilerplate PyTorch code, and managing model checkpoints creates a barrier that is orthogonal to the scientific questions being asked. Meanwhile, much of the deep learning engineering community remains focused on developing the next model that improves benchmark performance by a few percentage points, rather than ensuring that existing methods reach the hands of researchers who could apply them to problems that matter.

TRILL originated from a project investigating whether protein language models could "look beyond sequence similarity" to identify distantly related viruses in the *Microviridae* family. I had spent months building a dataset through iterative HMM searches in prior research and began experimenting on it with ESM in Jupyter notebooks. While notebooks are standard for computational research, I found

the experience clunky and recognized that most of the code I was writing was boilerplate, from sequence parsing and tokenization to PyTorch model loading and GPU management. The actual scientific decisions (which model to use, what proteins to embed, how to classify the results) were buried under layers of engineering code. I realized that the people who would benefit most from these methods, experimentalists with deep domain knowledge but limited computational experience, would experience great difficulties writing that code themselves. This led to a core design insight that has guided TRILL ever since. Through obfuscation of the underlying computational complexity, clarity and creativity could emerge for the end-user. By hiding the engineering details behind a simple command-line interface, researchers could focus on *what* they wanted to accomplish rather than *how* to accomplish it.

This gap motivated me to create TRILL (**TR**aining and **I**nference using the **L**anguage of **L**ife). TRILL offers a broad suite of capabilities spanning de novo sequence design, three-dimensional structure prediction, protein-ligand docking, molecular dynamics simulation, and property prediction, all accessible through a unified command-line interface. Its modular design allows users to compose complex multi-step workflows by chaining independent commands, pairing AI-driven generative models with physics-based simulation and scoring. I believe that the greatest impact of protein language models and related methods has not yet been realized, not because the models themselves are insufficient, but because they have not yet reached the researchers with the domain expertise to apply them to the problems where they would make the most difference. Even models that the field considers "outdated" could yield transformative results in the hands of a wet-lab scientist who understands the relevant biology but has never written a line of Python. By making these tools free, open-source, permissive, and completely local (requiring no cloud services or API keys), TRILL aims to be safely usable by academics, companies, and government researchers alike, removing not only technical barriers but also regulatory ones.

2.2 Design philosophy

The design philosophy of TRILL is rooted in a vision called *declarative bioengineering*, a paradigm in which users interact with complex computational pipelines through intuitive, high-level commands. Rather than engaging directly with low-level tasks such as molecular docking or structure prediction, users would instead be able to specify goals such as "design a binder for this target" or "optimize this enzyme for increased thermostability and substrate specificity." While TRILL currently operates in a more imperative mode, recent additions, such as the Foldtuning workflow

integrated into the `workflow` command, reflect a deliberate effort to lower the barrier to entry for performing advanced protein design and analysis.

This vision evolved organically. The TRILL repository was first created on October 25, 2021, initially as a tool for distant homology detection. As the project grew and I recognized the broader need for accessible, composable tooling in computational protein science, the scope expanded substantially. The first public release on GitHub was v0.2.2, published on December 11, 2022. Since then, TRILL has undergone major architectural refactors and has grown to over 5,000 lines of Python code across more than 800 commits, arriving at version 1.10.0 as of this writing.

Four core principles have guided the development of TRILL throughout its evolution.

Accessibility

The primary barrier to adoption of deep-learning models in biology is not conceptual but practical. Researchers who could benefit from protein language models, structure predictors, or generative design tools are often unable to use them because of the engineering overhead involved in setting up the required computational environment. Installing the correct versions of PyTorch, CUDA, and model-specific dependencies, resolving conflicts between packages, configuring GPU access, writing data loading and preprocessing code, and managing model checkpoints all require skills unrelated to the biological questions being asked. For a biologist who wants to fold a set of protein sequences, the relevant scientific decision is which model to use, not how to configure a batch converter or handle mixed-precision inference.

TRILL abstracts these complexities behind a command-line interface where a single command can accomplish what would otherwise require dozens to hundreds of lines of Python. A user who wants to embed proteins with Ankh, fine-tune ProtGPT2 on a protein family, or predict structures with ESMFold can do so with a one-line command that handles model downloading, tokenization, batching, GPU allocation, and output formatting automatically. Through this abstraction, TRILL reduces the lines of code needed to be written by end-users by a factor of 10–100. Crucially, this accessibility does not come at the cost of flexibility. TRILL exposes the relevant hyperparameters and configuration options for each model, so advanced users retain full control over the computational details when needed, while novice users can rely on sensible defaults.

Composability

Each of TRILL's commands is self-contained yet interoperable. Commands communicate through standardized file formats, FASTA for sequences, PDB for structures, and CSV for tabular results, so the output of any command can be passed directly as input to any other without format conversion or custom glue scripts. This transforms TRILL from a collection of model wrappers into a platform for creative experimentation, where models can be freely substituted within any step of a pipeline. Concrete workflow examples are presented in Section 2.11.

Scalability

TRILL is designed to function across the full spectrum of computational environments that researchers have access to, from free cloud platforms like Google Colab with a single GPU (or no GPU at all) to departmental workstations with one or two GPUs to institutional high-performance computing clusters with hundreds of GPUs across multiple nodes. The same TRILL command that runs on a laptop also runs on a SLURM-managed supercomputer, with the only difference being the number of GPUs specified as a command-line argument.

This scalability is achieved through the PyTorch Lightning abstraction layer, which handles things like device placement, distributed data parallelism, fully sharded data parallel (FSDP), and DeepSpeed ZeRO memory optimization. For models that exceed the memory of a single GPU, TRILL supports CPU parameter offloading through DeepSpeed ZeRO stages 2 and 3, allowing users to run models with billions of parameters on hardware that would otherwise be insufficient. Mixed-precision (16-bit) inference is enabled by default for most models, roughly halving memory consumption with minimal impact on output quality. TRILL also supports CPU-only execution for all models, ensuring that users without any GPU hardware can still access the full suite of capabilities, albeit with longer runtimes. This range of deployment options means that a researcher can prototype on a laptop, scale to a cluster for production runs, and share reproducible commands with collaborators regardless of their local hardware.

Integration of deep learning and physics-based methods

The recent success of deep-learning models in protein science has generated enormous excitement, and rightly so. Models like ESMFold, RFDiffusion, and LigandMPNN can perform tasks that were considered intractable just a few years ago, from

predicting protein structures without evolutionary information to generating entirely novel protein backbones from atomic noise. However, deep-learning models are fundamentally probabilistic, and their predictions are only as reliable as the patterns present in their training data. They can produce confident outputs for inputs that fall outside their training distribution, and they offer no guarantee that a generated protein will be physically stable, energetically favorable, or synthetically tractable. Relying solely on deep-learning predictions without independent validation risks propagating confident but incorrect results through a design pipeline.

Physics-based methods such as molecular dynamics simulations, force field-based energy minimization, and binding free energy estimations operate on fundamentally different principles. They model the physical forces governing molecular behavior through explicit equations of motion, electrostatics, van der Waals interactions, and solvation effects. While these methods are computationally expensive and limited in the conformational space they can explore, they provide an independent and complementary source of evidence grounded in physical law rather than statistical correlation. A protein sequence that scores well under a language model's pseudo-log-likelihood, adopts a plausible fold according to a structure predictor, and also exhibits low energy and stable dynamics in a molecular simulation is far more likely to succeed experimentally than one validated by any single approach alone.

TRILL is designed to make this hybrid approach practical rather than aspirational. By integrating deep-learning models for rapid hypothesis generation and physics-based tools for rigorous validation within the same command-line interface, TRILL enables workflows in which the speed and creativity of AI-driven design are tempered by the physical grounding of classical simulation. A user can generate thousands of candidate proteins with a language model in minutes, filter them with zero-shot scoring, predict their structures, and then subject the most promising candidates to molecular dynamics, all without leaving the TRILL environment. This combination allows researchers to leverage the efficiency and generative power of deep learning while maintaining the interpretive rigor and physical validity checks that have long been the foundation of computational biology.

2.3 The TRILL framework

This section describes the concrete structure of TRILL as of v1.10.0. The platform integrates over 50 deep-learning models and physics-based tools for protein design and analysis, supported by utilities for data preprocessing, visualization, and hyperpa-

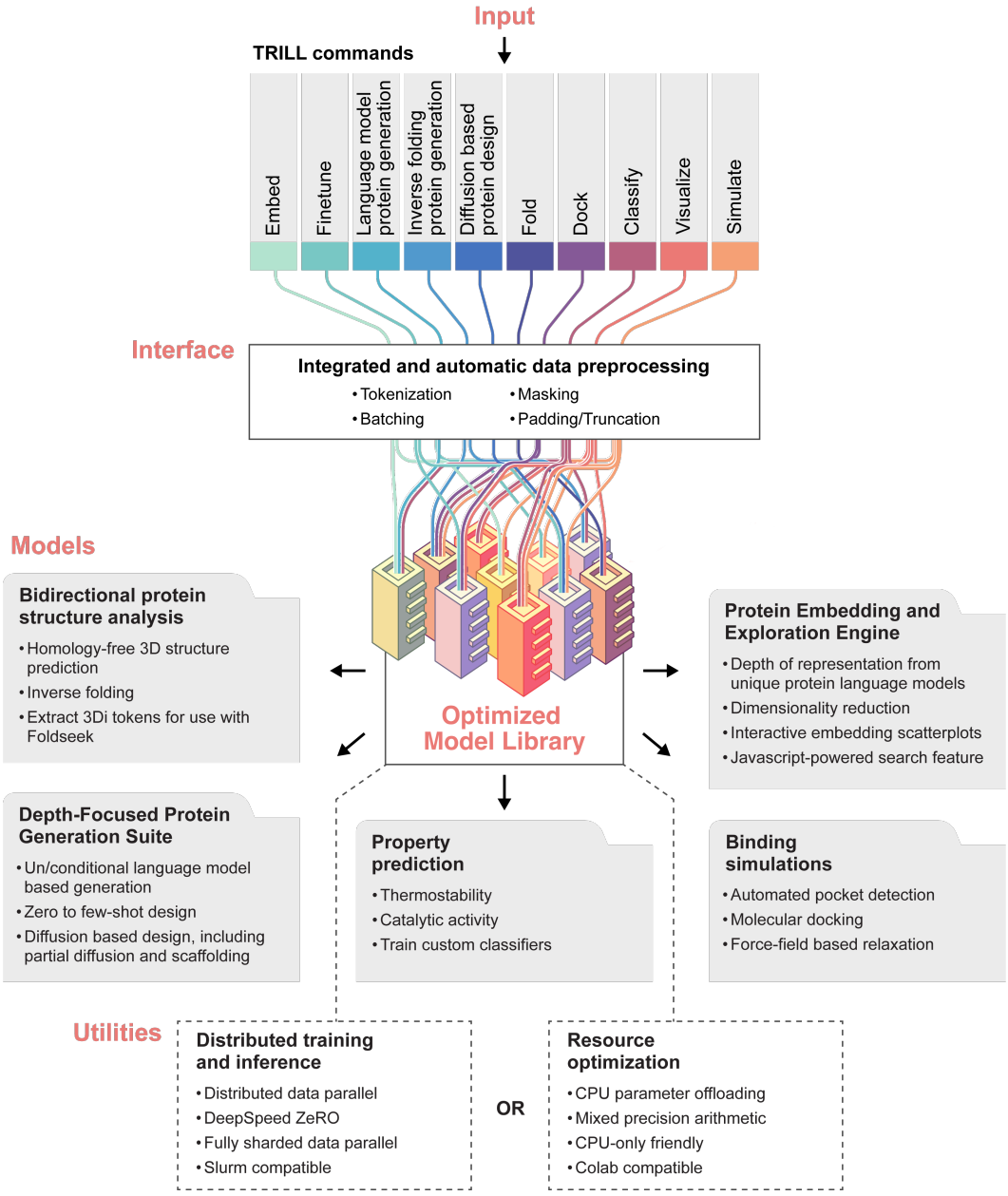


Figure 2.1: Abstracted TRILL architecture.

rameter optimization. All functionality is exposed through a unified command-line interface organized into thirteen commands (Table 2.1), enabling users to move fluidly between analytical approaches within a single session.

Command-line interface

The TRILL interface follows a consistent pattern. Every invocation begins with a run name and the number of GPUs to use, followed by a command and its associated arguments. The following examples illustrate this pattern.

```
$ trill my_run 1 embed esm2_t30_150M query.fasta --avg
$ trill my_run 1 finetune ProtGPT2 training.fasta --epochs 10
$ trill my_run 1 fold ESMFold query.fasta
```

This design means that users need only remember the high-level command structure. The details of model loading, tokenization, batching, and hardware configuration are handled internally. Setting the GPU count to 0 triggers CPU-only execution where supported, enabling use on machines without dedicated GPU hardware.

Commands and supported models

As of v1.10.0, TRILL provides thirteen commands, each encapsulating a distinct category of protein analysis or design functionality. Table 2.1 provides a full breakdown.

The breadth of supported models reflects TRILL's ambition to serve as an all-in-one platform. Notably, the **Embed** command spans multiple molecular modalities, including proteins (via ESM2, ProtT5-XL, Ankh, ProstT5, SaProt), nucleic acids (via RNA-FM, mRNA-FM, CaLM), and small molecules (via MolT5, SMI-TED, SELFIES-TED, MMELLON). This multi-modal embedding capability is, to my knowledge, unique among existing protein engineering platforms.

Integrated data preprocessing

A critical but often overlooked contribution of TRILL is its automatic handling of data preprocessing. When a user provides a FASTA file to any command, TRILL handles tokenization, batching, masking (for masked language model training), and padding/truncation appropriate to the selected model. Different models require fundamentally different preprocessing pipelines. ESM2 uses a custom alphabet with batch converter, ProtT5-XL uses a HuggingFace tokenizer, SaProt requires

Table 2.1: Breakdown of TRILL's commands and supported models as of v1.10.0. Full citations for each model are provided in the corresponding sections of the text.

Command	**Function**	**Available Models**
Embed	Generate numerical representations of biological sequences. Supports proteins, RNA/DNA, and small-molecule SMILES.	ESM2, MMELLON, MolT5, ProtT5-XL, ProstT5, Ankh, CaLM, mRNA-FM, RNA-FM, SaProt, SELFIES-TED, SMI-TED
Visualize	Dimensionality reduction and interactive 2D visualization of embeddings.	PCA, t-SNE, UMAP
Finetune	Fine-tune protein language models on user-provided sequences.	ESM2, ProtGPT2, ZymCTRL, ProGen2
Lang. Gen.	Generate proteins using pretrained or fine-tuned language models.	ESM2, ProtGPT2, ZymCTRL, ProGen2
Inv. Fold Gen.	Design sequences predicted to fold into a given 3D structure.	ESM-IF1, LigandMPNN, ProstT5
Diff. Gen.	Generate protein backbones via denoising diffusion.	Genie2, RFDiffusion
Fold	Predict 3D protein structures from sequence. Also co-folds with nucleic acids and/or small molecules.	ESMFold, ProstT5, Chai-1, Boltz-2
Dock	Predict binding poses of ligands to proteins.	DiffDock-L, AutoDock Vina, Gnina, LightDock, GeoDock
Classify	Predict protein properties or train custom classifiers on embeddings.	CataPro, CatPred, M-Ionic, PSICHIC, PSALM, TemStaPro, EpHod, ECPICK, LightGBM, XGBoost, Isolation Forest, ESM2+MLP
Regress	Train regression models on embeddings.	Linear, LightGBM
Simulate	Molecular dynamics with automated post-simulation analysis.	OpenMM
Score	Zero-shot scoring of sequences or structures.	ESM1v/ESM2, ProteinMPNN, LigandMPNN, COMPSS, SC
Workflow	End-to-end protein design workflows.	Foldtuning

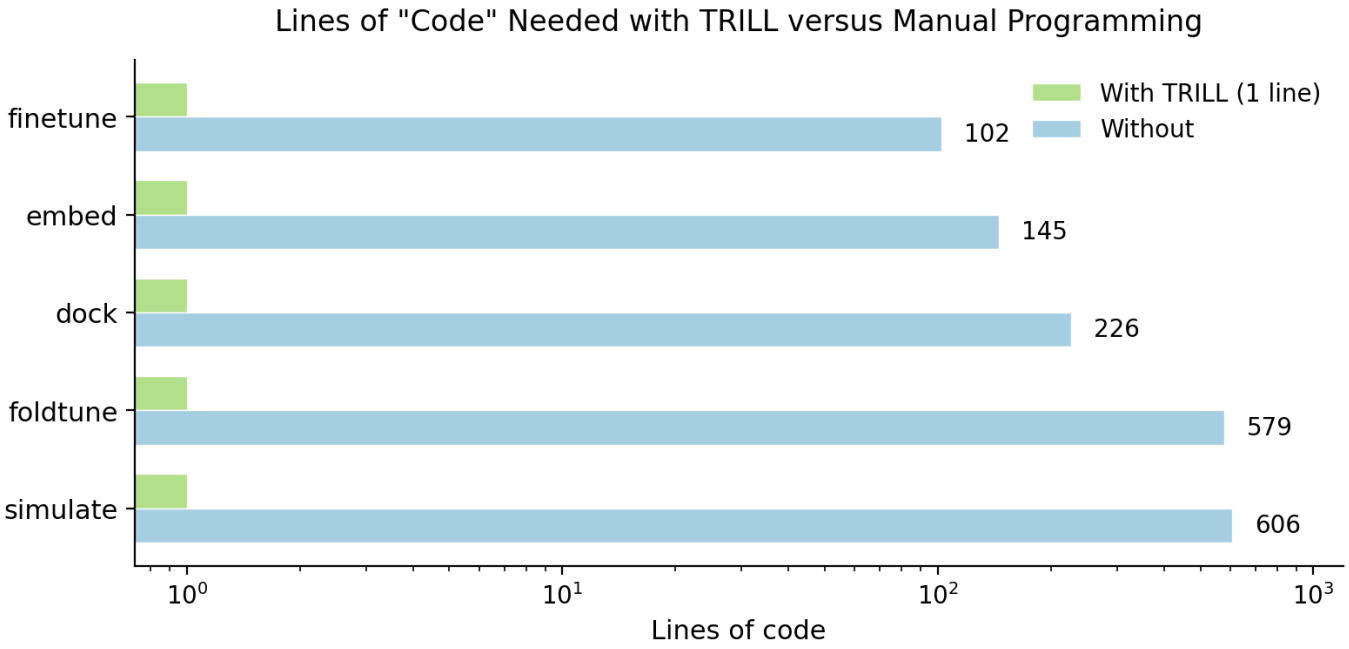


Figure 2.2: Comparing lines of code needed to run select TRILL commands

structure-aware tokens derived from PDB files, and ProGen2 uses a custom BPE tokenizer. TRILL abstracts all of these differences behind a uniform interface. Input file formats are standardized across the platform, with FASTA used for nucleotide, protein, and SMILES inputs, and PDB format used for structures.

2.4 Architecture and implementation

Entry point and command dispatch

TRILL's entry point (`trill_main.py`) implements a two-level argument parsing architecture using Python's `argparse` module. The first level captures global parameters that apply to all commands, including the run name, GPU count, number of compute nodes (`-nodes`), random number generator seed (`-RNG_seed`, default 123), output directory (`-outdir`), and number of CPU workers (`-n_workers`). The second level uses subparsers to dispatch to one of the thirteen command modules, each of which registers its own model-specific and task-specific arguments. This architecture means that every command shares a consistent invocation pattern (run name, GPU count, then command and arguments) while retaining full flexibility to define its own parameter space.

At startup, TRILL seeds all random number generators (PyTorch, NumPy, and HuggingFace Transformers) using `pl.seed_everything` and `set_seed` for repro-

ducibility. It configures logging via the Loguru library (with both console and file outputs), and records the total wall-clock time for each run. Each command module follows a uniform contract, exporting a `setup(subparsers)` function that registers arguments and a `run(args)` function that executes the command. This plugin-like architecture makes it straightforward to add new commands without modifying the core entry point.

Software stack

TRILL's dependency graph reflects its position at the intersection of deep learning, bioinformatics, cheminformatics, and physics-based simulation. The dependencies can be organized into several functional layers.

The deep-learning foundation consists of PyTorch (Paszke et al., 2019) for tensor computation and automatic differentiation, PyTorch Lightning (Falcon et al., 2020) for training abstractions and hardware management, HuggingFace Transformers (Wolf et al., 2020) for model loading and tokenization, HuggingFace Accelerate (Gugger et al., 2022) for multi-device execution, and DeepSpeed (Rajbhandari et al., 2020) for memory-efficient distributed training. For models that use graph neural networks (such as DiffDock and Genie2), TRILL depends on the Deep Graph Library (DGL), PyTorch Geometric (PyG), and the e3nn equivariant neural network library. Flash Attention and Triton are included for accelerated transformer inference on supported hardware.

The bioinformatics layer includes Biopython (Cock et al., 2009) for sequence parsing and manipulation, fair-esm for interfacing with the ESM model family, Foldseek for structural alphabet operations and remote homology searches, and SeqKit for efficient sequence processing. MDAnalysis (Gowers et al., 2016) provides trajectory analysis for molecular dynamics simulations.

The cheminformatics and simulation layer includes RDKit (Landrum et al., 2026) for molecular representation and manipulation, Open Babel (O'Boyle et al., 2011) for file format conversions between the many molecular file types encountered in docking workflows (PDB, PDBQT, MOL2, SDF), Meeko for preparing ligands for AutoDock-family docking, OpenMM (Eastman et al., 2017) for molecular dynamics, the Open Force Field toolkit for SMIRNOFF-based small-molecule parameterization, PDBFixer for automated structure preparation, ProLIF (Bouysset and Fiorucci, 2021) for molecular interaction fingerprinting, and AmberTools (Case et al., 2023) for MM/GBSA calculations via cpptraj and MMPBSA.py. FPocket (Le Guilloux,

Schmidtke, and Tuffery, 2009) is integrated through the BioExcel Building Blocks (biobb) library for binding pocket detection.

The machine learning and analysis layer includes XGBoost (Chen and Guestrin, 2016), LightGBM (Shi et al., 2026), and Scikit-Learn (Pedregosa et al., 2011) for classical machine learning classifiers and regressors, Optuna (Akiba et al., 2019) for Bayesian hyperparameter optimization, and Bokeh for interactive visualization. UMAP, t-SNE (via Scikit-Learn), and PCA are available for dimensionality reduction.

TRILL manages this complex dependency graph using Pixi, a cross-platform package manager that resolves packages from both conda (for compiled scientific libraries such as OpenMM, AmberTools, RDKit, and CUDA toolkits) and PyPI (for Python-specific packages and git-based dependencies). Several dependencies are installed from custom GitHub forks maintained by the author (for LightDock, CaLM, ECPICK, SCASA, and DiffDock), where modifications were necessary for compatibility with TRILL's interface or to resolve upstream dependency conflicts. PyTorch Geometric extensions (torch-sparse, torch-cluster, torch-spline-conv) are installed from pre-built wheels pinned to the specific PyTorch and CUDA versions to avoid compilation issues.

Runtime model integration

Not all models integrated into TRILL are installed as static dependencies. Several models are cloned from GitHub repositories at runtime on first use and cached in TRILL's local cache directory (`~/.trill_cache`). This strategy is used for models whose codebases have complex or conflicting dependencies that cannot be cleanly resolved alongside TRILL's core packages, or for models that are distributed as standalone research repositories rather than installable packages. Examples include Boltz-2, Chai-1, RFDiffusion, Genie2, DiffDock, LigandMPNN, MMELLON, PSICHIC, CatPred, and CataPro.

For each runtime-integrated model, TRILL uses GitPython to clone the repository into the cache directory, inserts the cloned path into `sys.path`, and imports the necessary modules. In some cases, TRILL programmatically patches the cloned source code to remove incompatible dependencies or CLI-specific decorators. For example, the Boltz-2 integration comments out Flash Attention import lines (which may not be available on all systems) and removes Click CLI decorators from the main prediction function so it can be called as a library. This pragmatic approach, while not elegant, reflects the reality of integrating rapidly evolving research codebases

that were not designed for use as importable libraries.

Safe model serialization

TRILL implements a safe model loading and saving utility (`safe_load.py`) that addresses security concerns associated with Python's `pickle`-based serialization, which is used by `torch.load` and can execute arbitrary code during deserialization. The utility preferentially loads models in the safetensors format when available, a format developed by HuggingFace that stores only tensor data without executable code. When loading traditional `.pt` or `.pth` files, TRILL uses PyTorch's `weights_only=True` parameter (available in PyTorch 2.6+) to restrict deserialization to tensor data only. If a model requires full pickle deserialization for compatibility (such as older checkpoints that include non-tensor metadata), TRILL falls back to `weights_only=False` with a warning. The utility also provides conversion functions for migrating existing PyTorch checkpoints to safetensors format, and supports dual-format saving (both `.pt` and `.safetensors`) for forward compatibility.

The PyTorch Lightning conversion

A key architectural contribution of TRILL lies in the conversion of most supported protein language models into PyTorch Lightning `LightningModule`s. This design choice was born of necessity. Early in TRILL's development, I manually wrote distributed training code in raw PyTorch. While deeply informative, this approach was fragile, error-prone, and difficult to maintain across the growing number of models. The discovery of PyTorch Lightning, with its clean abstractions for multi-GPU training, mixed-precision inference, and checkpoint management, was transformative. Porting models to Lightning modules was a clear architectural improvement that enabled a host of advanced features.

Several integrations required upstream modifications to the original model repositories to support the Lightning interface. The Lightning-wrapped models are organized in a central `lightning_models.py` module that defines `LightningModule` subclasses for ESM2, ProtGPT2, ZymCTRL, ProtT5-XL, ProstT5, Ankh, SaProt, CaLM, RNA-FM, and others. Each module implements the standard Lightning interface (`training_step`, `predict_step`, `configure_optimizers`) along with model-specific data handling. A shared `CustomWriter` callback handles the collection and serialization of distributed prediction outputs across multiple GPUs, merging per-device result files into consolidated output after inference completes.

Logging and reproducibility

TRILL uses the Loguru library for structured logging, with color-coded console output and plain-text file output. Every run generates a timestamped log file in the output directory that records the RNG seed, all model parameters, and the total elapsed time. This log file provides an audit trail for reproducing results.

Resource optimization

TRILL employs several strategies to minimize the computational and storage overhead of working with large models.

Model weights are cached in a persistent directory (`~/.trill_cache`) and downloaded only on first use. HuggingFace models are cached through the standard HuggingFace Hub mechanism, while models distributed as direct downloads (such as CaLM) are fetched via HTTP requests and stored locally. Runtime-cloned GitHub repositories are also cached in this directory. This caching strategy ensures that repeated runs do not incur redundant network transfers, and that TRILL can operate in bandwidth-limited or offline environments after initial setup.

The GPU memory optimization strategies described in Section 2.2 (mixed-precision inference, DeepSpeed ZeRO sharding, and CPU offloading) are implemented at this layer. At the implementation level, CPU-only execution is achieved by setting the `CUDA_VISIBLE_DEVICES` environment variable to disable CUDA entirely, which required lightweight modifications to several model codebases that assumed GPU availability.

2.5 Embedding and exploration

The case for learned representations

Biological sequences are conventionally represented as one-dimensional strings of characters, namely ATGC for DNA, AUGC for RNA, and the ARNDCQEGHILKMFPSTWYV for proteins. While these representations correctly encode immediate neighbors (position 2 is adjacent to positions 1 and 3), they fail to capture the three-dimensional macromolecular interactions that govern biological function. A residue at position 100 is 99 characters away from position 1 in sequence space, but the two may be in direct physical contact due to protein folding. Sequence alignment-based computational biology, especially with the adoption of hidden Markov models (Eddy, 2011), has enabled great discoveries, but string-based representations carry fundamental limitations.

Chief among these is the problem of *distant homology detection*. Two sequences may share common ancestry but lack recognizable sequence similarity, rendering alignment-based methods ineffective. This challenge is further compounded by *analogy detection*, where two sequences have convergently evolved to perform similar functions without necessarily sharing similarity at the sequence level. Functional screens are similarly constrained, since sequence alignments cannot infer actual functionality. A single point mutation can cause a variant to lose function entirely, yet the alignment score barely changes. Furthermore, pairwise comparison is an inherent bottleneck, as comparing n sequences requires $O(n^2)$ operations. Finally, sequence-based methods are limited by existing annotated databases. Since most of biology remains unsequenced, these approaches are inherently restricted when studying highly divergent or non-model organisms.

Transformer-based language models offer a fundamentally different approach. During training, these architectures learn high-dimensional representations of the "language" they are modeling, whether English or protein sequences, as a byproduct of performing their training objective (e.g., masked language modeling for ESM2, or causal language modeling for ProtGPT2). While the raw internal representations are sequence-length dependent, researchers typically *pool* them (e.g., by averaging across residue positions) to obtain a fixed-length vector for each sequence, enabling comparison of proteins regardless of length or similarity.

These embeddings offer several practical advantages over raw sequences. Because they capture latent structural and functional signals learned from millions of evolutionary examples, they can reveal relationships between proteins that are invisible to alignment-based methods, including distant homologs with diverged sequences and functional analogs that arose through convergent evolution. Embeddings are also inherently reusable. Since a given model's representation of a sequence is deterministic, embeddings can be precomputed once and stored in a database for rapid reuse across multiple downstream tasks without re-running the model. Finally, as fixed-length numerical vectors, embeddings benefit from efficient linear algebra operations, greatly accelerating sequence classification, comparison, clustering, and regression relative to the iterative pairwise operations required by alignment-based approaches.

Supported embedding models

TRILL integrates 20 embedding models spanning three molecular modalities, namely proteins, nucleic acids, and small molecules. Table 2.3 provides a detailed breakdown.

Table 2.3: Embedding models available in TRILL as of v1.10.0, organized by molecular modality.

Model	Molecule	Parameters
ESM2-8M	Protein sequence	8M
ESM2-35M	Protein sequence	35M
ESM2-150M	Protein sequence	150M
ESM2-650M	Protein sequence	650M
ESM2-3B	Protein sequence	3B
ESM2-15B (Lin et al., 2023)	Protein sequence	15B
Ankh (Elnaggar et al., 2023)	Protein sequence	450M
Ankh-Large	Protein sequence	1.15B
ProtT5-XL (Elnaggar et al., 2022)	Protein sequence	3B
ProstT5 (Heinzinger et al., 2024)	Protein sequence	3B
SaProt (Su et al., 2023)	Protein sequence + structure	650M
CaLM (Outeiral and Deane, 2024)	DNA	86M
RNA-FM (Chen et al., 2022)	RNA	99M
mRNA-FM (Chen et al., 2022)	RNA	239M
MMELLON (Suryanarayanan et al., 2024)	Small-molecule SMILES	84M
MolT5-Small	Small-molecule SMILES	77M
MolT5-Base	Small-molecule SMILES	250M
MolT5-Large (Edwards et al., 2022)	Small-molecule SMILES	800M
SMI-TED (Soares et al., 2024)	Small-molecule SMILES	289M
SELFIES-TED (IBM Research and Priyadarsini, 2026)	Small-molecule SMILES	358M

The following subsections describe each model family in detail, organized by molecular modality.

Protein sequence models

ESM2 ESM2 (Lin et al., 2023) is a family of BERT-style transformer encoder protein language models trained on approximately 65 million unique protein sequences sampled from UniRef. The training objective is masked language modeling (MLM), where 15% of residue positions are randomly replaced with a mask token and the model is optimized to recover the original amino acid at each masked position using only the unmasked sequence as input. ESM2 is available in TRILL at six scales with the following configurations. The 8M parameter variant has 6 layers, 20 attention heads, and produces 320-dimensional per-residue embeddings. The 35M variant has

12 layers, 20 heads, and 480-dimensional embeddings. The 150M variant has 30 layers, 20 heads, and 640-dimensional embeddings. The 650M variant has 33 layers, 20 heads, and 1,280-dimensional embeddings. The 3B variant has 36 layers, 40 heads, and 2,560-dimensional embeddings. The 15B variant has 48 layers, 40 heads, and 5,120-dimensional embeddings. The input to TRILL is a protein sequence in FASTA format. The output is a per-residue embedding matrix of dimension $L \times d$, where L is the sequence length and d is the model-dependent hidden dimension listed above. The same paper also introduced ESMFold, a structure prediction module that operates directly on ESM2 representations, showing that these embeddings encode enough structural signal to fold proteins into three-dimensional coordinates without requiring multiple sequence alignments.

Ankh Ankh (Elnaggar et al., 2023) is a T5-style encoder-decoder protein language model focused on optimizing performance per parameter rather than maximizing scale. It is available in two sizes. Ankh has 450M parameters, with 48 encoder layers, 24 decoder layers, 12 attention heads, a feed-forward dimension of 3,072, and a hidden dimension of 768. Ankh-Large has 1.15B parameters, with 48 encoder layers, 24 decoder layers, 16 attention heads, a feed-forward dimension of 3,840, and a hidden dimension of 1,536. Both models were trained on UniRef50 using a 1-gram random token masking strategy at a 20% masking probability with full sequence reconstruction. The input to TRILL is a protein sequence in FASTA format. Embeddings are extracted from the encoder's last hidden states, producing per-residue vectors of dimension 768 (Ankh) or 1,536 (Ankh-Large).

ProtT5-XL ProtT5-XL (Elnaggar et al., 2022) is a 3-billion parameter T5-style encoder-decoder model from the ProtTrans project. The architecture has 24 encoder layers and 24 decoder layers, with 32 attention heads and a hidden dimension of 1,024. It was trained on data from UniRef50 and the BFD (Big Fantastic Database) using a denoising objective that corrupts and reconstructs single tokens at a 15% masking probability. Unlike BERT-family models that use absolute position embeddings, the T5 architecture employs per-head relative position biases shared across all layers, allowing the model to generalize to sequences longer than those seen during training by reasoning about inter-residue distances rather than fixed positional indices. The input to TRILL is a protein sequence in FASTA format, where each amino acid is treated as a single token separated by spaces. The output is a 1,024-dimensional per-residue embedding extracted from the encoder's last hidden state.

ProstT5 ProstT5 (Heinzinger et al., 2024) is a "bilingual" 3-billion parameter T5-style model that shares the ProtT5-XL architecture (24 encoder layers, 24 decoder layers, 32 attention heads, hidden dimension 1,024), fine-tuned on both amino acid sequences and 3Di structural alphabet tokens. The 3Di alphabet, developed for the Foldseek structural search tool (Van Kempen et al., 2024), encodes each residue's local backbone geometry as one of 20 discrete tokens, yielding a one-dimensional structural "sequence" that can be processed by the same language modeling machinery used for amino acid sequences. Training proceeded in two stages. First, the ProtT5-XL checkpoint was adapted to 3Di tokens through continued denoising pretraining on both sequence types. Second, the model learned sequence-to-structure and structure-to-sequence translation tasks simultaneously. The model was trained on 17 million high-quality, non-redundant 3D predictions from the AlphaFold Database. The input to TRILL is a standard FASTA file of protein or 3Di sequences. The output is a 1,024-dimensional per-residue embedding from the encoder.

SaProt SaProt (Su et al., 2023) is a 650-million parameter BERT-style masked language model whose key innovation is a dual-alphabet tokenization scheme, where each residue is represented by both its amino acid identity and its Foldseek-derived 3Di structural state, creating a combined "structure-aware" (SA) token vocabulary. The architecture follows the ESM2-650M configuration with 33 layers, 20 attention heads, and a hidden dimension of 1,280. Unlike ProstT5, which translates between amino acid and 3Di alphabets, SaProt interleaves them into a combined SA-token sequence where each position carries both sequence and structure information (e.g., "VpLEKdSI" alternates residue and structure characters). The model was trained on approximately 40 million protein sequence-structure pairs from the AlphaFold Database using the standard MLM objective at 15% masking probability. The input to TRILL is protein structure files in PDB format, provided either as a directory of PDB files or a text file listing paths. TRILL handles the Foldseek-based conversion to SA-tokens automatically. The output is a 1,280-dimensional averaged embedding for each protein.

Nucleic acid models

CaLM CaLM (Codon Adaptation Language Model) (Outeiral and Deane, 2024) is an 86-million parameter transformer encoder model that operates on codons (nucleotide triplets) rather than amino acids. Because multiple codons encode the same amino acid, the genetic code is a many-to-one mapping, implying that

codon-level sequences carry information beyond what is captured by the amino acid translation alone. Experimental evidence has connected the choice among synonymous codons to several biological processes, including the kinetics of co-translational folding, the efficiency of protein maturation, and the regulation of gene expression levels. CaLM exploits this additional information by training directly on protein-coding DNA (cDNA) sequences. The architecture stacks 12 transformer encoder blocks (hidden dimension 768) between a learned codon embedding layer at the input and a prediction head at the output, where the prediction head reuses the transpose of the input embedding matrix to project back to the codon vocabulary. It was trained on 9 million non-redundant cDNA sequences from the European Nucleotide Archive, with species-level clustering and a 40% similarity cutoff between training and held-out sets. The input to TRILL is a FASTA file of DNA sequences. The output is a 768-dimensional per-codon embedding that can be pooled to a fixed-length vector.

RNA-FM and mRNA-FM RNA-FM (Chen et al., 2022) is a 99-million parameter foundation model for non-coding RNA that employs a 12-layer bidirectional transformer encoder (BERT-style) with a hidden dimension of 640. The model was pretrained with a self-supervised masked token prediction objective on a corpus of 23.7 million non-coding RNA sequences drawn from the RNAcentral repository. The learned representations encode both local sequence context and deeper evolutionary signals, enabling tasks such as tracing the mutational trajectory of SARS-CoV-2 lineages and reconstructing phylogenetic relationships among long non-coding RNAs. mRNA-FM is a related variant with 239M parameters specifically designed for messenger RNA, requiring that input sequences have lengths that are multiples of three (corresponding to codon boundaries). The input to TRILL for both models is a FASTA file of RNA sequences in single-letter nucleotide notation. The output is a 640-dimensional per-nucleotide embedding.

Small-molecule models

MolT5 MolT5 (**Mol**ecular **T5**) (Edwards et al., 2022) is a T5-style encoder-decoder framework that undergoes joint self-supervised pretraining on two monolingual corpora, one of English text and one of SMILES (Simplified Molecular-Input Line-Entry System) molecular notation strings. After pretraining, MolT5 supports two cross-modal translation tasks. The first is molecule captioning, which converts a SMILES structure into a free-text molecular description. The second is text-guided

generation, which designs a novel SMILES structure matching a user-provided textual specification. It is available in three sizes following standard T5 configurations. MolT5-Small has 77M parameters with 6 encoder and 6 decoder layers, 8 attention heads, and a hidden dimension of 512. MolT5-Base has 250M parameters with 12 encoder and 12 decoder layers, 12 heads, and a hidden dimension of 768. MolT5-Large has 800M parameters with 24 encoder and 24 decoder layers, 16 heads, and a hidden dimension of 1,024. All variants were pretrained on a denoising objective over both text and SMILES corpora. The input to TRILL is a FASTA-formatted file where headers identify compounds and sequences are SMILES strings. The output is a 512, 768, or 1,024-dimensional embedding from the encoder, depending on model size.

SMI-TED SMI-TED (Soares et al., 2024) is a 289-million parameter encoder-decoder chemical foundation model. Its pretraining corpus consists of 91 million curated SMILES representations drawn from PubChem, totaling roughly 4 billion molecular tokens. The architecture has 12 layers, 12 attention heads, and a hidden dimension of 768, incorporating a variant of rotary position embeddings (RoFormer) that applies rotation-based relative encoding to the query and key vectors. The training objective combines masked language modeling (15% token masking) on the encoder with token reconstruction on the decoder, using a two-phase strategy where the token encoder is pretrained first (95% of data) before jointly training both components. The input to TRILL is a SMILES string in FASTA format. The output is a fixed-length 768-dimensional embedding from the encoder.

SELFIES-TED SELFIES-TED (IBM Research and Priyadarsini, 2026) replaces SMILES with SELFIES (SELF-referencing Embedded Strings) as the molecular input representation. SELFIES is an alternative molecular string notation in which discrete tokens, delimited by square brackets and representing atoms, bonds, or branching operations (e.g., `[C]`, `[=O]`, `[Branch1]`), are guaranteed to always encode a chemically valid molecule, avoiding the syntactic invalidity issues that can arise with SMILES. SELFIES-TED employs the BART encoder-decoder framework, which pairs a bidirectional encoder with an autoregressive decoder, using 12 layers and 16 attention heads in each component. The large variant has 358M parameters and was pretrained with a denoising objective (15% token masking) on 1 billion molecule samples from ZINC-22 and PubChem. The input to TRILL is a FASTA-formatted file of SMILES strings, which are internally converted to SELFIES. The output is a

1,024-dimensional embedding for each molecule.

MMELLON MMELLON (**M**ulti-view **M**olecular **E**mbedding with **L**ate Fusion) (Suryanarayanan et al., 2024) departs from single-representation strategies by fusing three complementary molecular perspectives, a graph encoding of atomic connectivity, a rendered 2D structural depiction, and the linear SMILES string, into a single embedding. Each view is processed by its own pretrained foundation model, and the resulting embeddings are combined through an attention-based pretrained aggregator. Each modality-specific encoder was independently pretrained on corpora containing as many as 200 million molecular examples. Unlike the other small-molecule models in TRILL, MMELLON does not rely on a single transformer architecture but instead fuses representations from modality-specific models. The input to TRILL is a FASTA-formatted file of SMILES strings. MMELLON internally generates the graph and image views from the SMILES representation and produces a unified multi-view 768-dimensional embedding as output.

Implementation in TRILL

Most of the embedding models described above have been ported into PyTorch Lightning (Falcon et al., 2020) `LightningModule`s within TRILL, inheriting the mixed-precision inference and distributed execution capabilities described in Section 2.4.

Embedding modalities and pooling

The `embed` command supports three output modalities. **Averaged embeddings** (`-avg`) produce a single fixed-length vector per sequence by mean-pooling across residue positions, suitable for sequence-level classification, regression, and clustering tasks. **Per-residue embeddings** (`-per_AA`) retain the full residue-level representation matrix, useful for tasks requiring positional resolution such as residue contact prediction or binding site identification. **Pool PaRTI embeddings** (`-poolparti`) (Tartici, Nayar, and Altman, 2025) offer a more sophisticated alternative to simple mean pooling. Pool PaRTI applies the PageRank algorithm to attention-derived token interaction graphs within the transformer, yielding importance-weighted residue contributions that produce richer fixed-length embeddings and highlight functionally significant positions in the sequence. Users may also embed sequences using their own fine-tuned models via the `-finetuned` flag, enabling downstream analyses that leverage domain-adapted representations.

Interactive visualization

To facilitate exploratory data analysis, TRILL includes an interactive "exploration engine" through the `visualize` command. Users specify a dimensionality reduction technique, such as PCA (*Principal Component Analysis* 2002), t-SNE (Maaten and Hinton, 2008), or UMAP (McInnes, Healy, and Melville, 2018), and TRILL generates an interactive 2D visualization of their protein embeddings as a shareable HTML document. These interactive scatter plots, powered by the Bokeh library, support real-time text-based searching and filtering of labels, enabling rapid identification of clusters and outliers. The HTML files can be easily shared and viewed by anyone with an internet browser, providing an accessible entry point for researchers to explore the structure of their embedding spaces without writing any visualization code.

Precomputed embeddings

To reduce startup cost, precomputed ProtT5-XL embeddings, both full per-residue and averaged, are available for major model organisms including *Arabidopsis thaliana*, *Caenorhabditis elegans*, *Escherichia coli*, *Homo sapiens*, *Mus musculus*, *Rattus norvegicus*, and SARS-CoV-2, as well as for the entire UniProtKB. These precomputed resources allow users to rapidly prototype classification or clustering experiments without the computational overhead of embedding from scratch.

2.6 Fine-tuning and language model protein generation

TRILL provides two complementary commands for adapting and deploying generative protein language models. The `finetune` command adapts a pretrained model to a user-provided set of protein sequences, and the `lang_gen` command generates novel sequences from either base or fine-tuned models. Together, these commands enable a workflow in which a user can specialize a general-purpose protein language model on a family of interest and then sample new members of that family with tunable diversity.

Supported generative models

TRILL supports four families of generative protein language models, spanning two training paradigms.

ProtGPT2 ProtGPT2 (Ferruz, Schmidt, and Höcker, 2022) is a 738-million parameter autoregressive transformer with a decoder-only architecture of 36 layers, trained using the causal language modeling (CLM) objective on approximately

50 million non-redundant protein sequences from UniRef50. In autoregressive generation, the probability of a full sequence is decomposed as the product of conditional next-token probabilities, and the model is trained by minimizing the negative log-likelihood over the training corpus. ProtGPT2 generates proteins by iteratively sampling the next amino acid conditioned on all previously generated residues, starting from a user-specified seed sequence (default "M").

ZymCTRL ZymCTRL (Munsamy et al., 2024) extends the autoregressive protein generation paradigm with conditional control tags. Built on a GPT-2 architecture, ZymCTRL was trained on enzyme sequences from UniProt with Enzymatic Commission (EC) numbers prepended as control tags, allowing the model to generate enzymes conditioned on a specific functional class. For example, providing the tag "1.1.1.1" directs the model to generate alcohol dehydrogenases. The input to TRILL for fine-tuning is a FASTA file of enzyme sequences paired with a user-specified EC number via the `-ctrl_tag` flag. During generation, the same flag specifies the desired enzyme class, and ZymCTRL produces sequences that respect both the learned protein grammar and the functional constraint imposed by the tag.

ProGen2 ProGen2 (Nijkamp et al., 2023) is a family of autoregressive protein language models scaled from 151M to 6.4B parameters, trained on different sequence datasets drawn from genomic, metagenomic, and immune repertoire databases totaling over one billion proteins. The models use a standard causal transformer decoder architecture. ProGen2-small has 151M parameters with 12 layers and 16 attention heads. ProGen2-medium and ProGen2-base each have 764M parameters with 27 layers and 16 heads. ProGen2-large has 2.7B parameters with 32 layers and 32 heads. ProGen2-xlarge has 6.4B parameters with 32 layers and 16 heads but a head dimension of 256. ProGen2 also supports flexible control tags, which can represent any arbitrary class of proteins. In TRILL, multi-class fine-tuning is possible through CSV-formatted input files that map FASTA files to their respective control tags.

ESM2 (Gibbs Sampling) Although ESM2 is primarily a masked language model rather than an autoregressive generator, TRILL implements a Gibbs sampling strategy that repurposes ESM2 for iterative sequence generation. Starting from a seed sequence, the algorithm repeatedly masks one or more positions and samples replacements from the model's predicted probability distribution over the amino

acid vocabulary. Over many iterations, this process produces a new sequence that is consistent with the statistical patterns learned by ESM2. Users can control whether positions are filled in order or randomly via the `-random_fill` flag, how many positions are regenerated per iteration via `-num_positions`, and the sampling temperature via `-temp`.

Table 2.4: Generative language-based models available in TRILL as of v1.10.0.

Model	**Pretraining Objective**	**Parameters**
ESM2 (Gibbs Sampling) (Lin et al., 2023)	Masked Language Modeling	8M – 15B
ProtGPT2 (Ferruz, Schmidt, and Höcker, 2022)	Causal Language Modeling (CLM)	738M
ZymCTRL (Munsamy et al., 2024)	Conditional CLM with EC Tags	738M
ProGen2-small (Nijkamp et al., 2023)	CLM	151M
ProGen2-medium	CLM	764M
ProGen2-base	CLM	764M
ProGen2-large	CLM	2.7B
ProGen2-oas	CLM on Antibodies	764M
ProGen2-BFD90	CLM	764M
ProGen2-xlarge	CLM	6.4B

The finetune command

The `finetune` command adapts a pretrained model to a user-provided dataset of protein sequences. TRILL supports two distinct training objectives depending on the model family. For ESM2, training uses the masked language modeling (MLM) objective, where a configurable fraction of residues (default 15%, adjustable via `-mask_fraction`) is randomly masked and the model learns to reconstruct them. Users can also provide pre-masked FASTA files via `-pre_masked_fasta` for custom masking strategies, such as masking only active site residues or conserved motifs. For the autoregressive models (ProtGPT2, ZymCTRL, and ProGen2), training uses the causal language modeling (CLM) objective, where the model learns to predict each token given all preceding tokens.

The command exposes a wide range of training hyperparameters. These include learning rate (`-lr`, default 0.0001), number of epochs (`-epochs`, default 10), batch size (`-batch_size`, default 1), and training strategy (`-strategy`). The strategy

parameter accepts PyTorch Lightning strategies, including Distributed Data Parallel and DeepSpeed stages 1 through 3, with optional CPU offloading for training models that exceed GPU memory. For ProGen2 specifically, TRILL also supports gradient accumulation steps (`-grad_accum_steps`), learning rate schedulers (`-scheduler`, with options including linear, cosine, cosine with restarts, polynomial, constant, constant with warmup, and inverse square root), and warmup steps (`-warmup_steps`).

Checkpoint management is handled through two mechanisms. The `-save_on_epoch` flag saves a model checkpoint after every completed epoch, enabling users to compare generations from models at different stages of training. The `-finetuned` flag allows loading a previously saved checkpoint to continue training, enabling iterative refinement without restarting from the pretrained weights. For models trained with DeepSpeed, TRILL automatically converts the sharded ZeRO checkpoints into a single full-precision model file.

Language model based generation

The `lang_gen` command generates novel protein sequences from either base or fine-tuned language models. Users specify the model, the number of sequences to generate (`-num_return_sequences`), a seed sequence to initiate generation (`-seed_seq`, default "M"), and the maximum sequence length (`-max_length`, default 100). Generation can be batched for efficiency via `-batch_size`, depending on memory constraints.

Several sampling parameters allow users to control the diversity and quality of generated sequences. The sampling temperature (`-temp`) controls the sharpness of the probability distribution over the amino acid vocabulary, where lower temperatures produce more conservative, high-confidence predictions and higher temperatures increase diversity. Top-k filtering (`-top_k`, default 950) restricts sampling to the k most probable next tokens. A repetition penalty (`-repetition_penalty`, default 1.2) discourages the model from generating the same amino acid or motif repeatedly. Users can also disable stochastic sampling entirely via `-do_sample` to use greedy decoding, which deterministically selects the most probable token at each step.

The output of `lang_gen` is a FASTA file containing the generated sequences, with headers that encode the run name, model, and sequence index for traceability. When using a fine-tuned model, users pass the checkpoint path via the `-finetuned` flag, and for ZymCTRL/ProGen2 the `-ctrl_tag` flag specifies the desired protein class

for conditional generation.

2.7 Structure-based protein generation

In addition to language model-based generation, TRILL provides two commands for generating proteins conditioned on three-dimensional structural information. The `inv_fold_gen` command addresses the inverse folding problem, designing amino acid sequences that are predicted to fold into a given backbone structure. The `diff_gen` command addresses the complementary problem of de novo backbone generation, creating entirely new three-dimensional protein structures from noise. Together, these two commands form a natural pipeline in which diffusion models generate novel backbones and inverse folding models then design sequences to realize those backbones. Table 2.6 summarizes the available models.

Table 2.6: Structure-based protein generation models available in TRILL.

Model	Task	Approach	Input	Design Modes in TRILL
Inverse Folding (`inv_fold_gen`)				
ESM-IF1	Seq. from structure	GVP encoder + Transformer decoder (142M params)	PDB	Full redesign, contiguous region redesign
LigandMPNN	Seq. from structure + context	Message-passing GNN (2.6M params)	PDB (with optional ligand, nucleic acid, metal)	5 model variants (soluble, membrane, etc.), per-residue control, AA biases, symmetry constraints
ProstT5	Seq. from 3Di tokens	Bilingual T5 translation (3B params)	PDB	No structure-specific training needed, Foldseek conversion handled automatically
Diffusion-Based Generation (`diff_gen`)				
RFDiffusion	De novo backbone generation	Denoising diffusion on SE(3) frames	Optional PDB (for scaffolding/ binder design)	Unconditional, motif scaffolding, partial diffusion, binder design
Genie2	De novo backbone generation	SE(3)-equivariant diffusion	Optional PDB (for scaffolding)	Unconditional, multi-motif scaffolding

Inverse folding

Inverse folding is the problem of predicting a protein sequence from its backbone atom coordinates. Given a fixed three-dimensional backbone structure, the goal is to find an amino acid sequence that will fold into that structure. This is the reverse of the structure prediction ("folding") problem and is a core task in computational protein design. TRILL supports three inverse folding models through the `inv_fold_gen` command.

ESM-IF1 ESM-IF1 (Hsu et al., 2022) is a 142-million parameter hybrid model that combines a Geometric Vector Perceptron (GVP) encoder with a generic autoregressive Transformer decoder. The GVP encoder layers extract rotation-equivariant geometric features from backbone atom coordinates (N, Cα, and C atoms), which are then converted to rotation-invariant features through a change-of-basis into local reference frames and passed to 8 Transformer encoder layers and 8 Transformer decoder layers. ESM-IF1 was trained on approximately 12 million AlphaFold2-predicted structures from UniRef50, combined with approximately 16,000 experimentally determined structures from CATH. The input to TRILL is a PDB file containing the target backbone structure. Users can control the sampling temperature (`-temp`), the number of sequences to generate (`-num_return_sequences`), and the maximum sequence length (`-max_length`, default 500). TRILL also exposes a `-contig_redesign` flag that allows users to specify a contiguous range of residues to redesign while holding the rest of the sequence fixed. For example, `-contig_redesign 1:100` would redesign only the first 100 amino acids during sampling, leaving the remainder of the structure's native sequence unchanged. This is useful for targeted redesign of specific regions such as loops or binding interfaces without perturbing the rest of the protein.

LigandMPNN LigandMPNN (Dauparas et al., 2025) extends the ProteinMPNN (Dauparas et al., 2022) message-passing neural network architecture to explicitly model non-protein atomic context during sequence design. ProteinMPNN represents protein backbones as sparse graphs with residues as nodes and edges defined by Cα-Cα distances, processing them through three encoder layers with 128 hidden dimensions followed by an autoregressive decoder. LigandMPNN augments this with two additional protein-ligand encoder layers that operate on a joint graph containing both protein residues and ligand atoms as nodes. The protein-ligand edges encode distances between backbone atoms (N, Cα, C, O, and virtual Cβ) and the 25 closest ligand atoms per residue. LigandMPNN has 2.62 million parameters compared to ProteinMPNN's 1.66 million. This context-aware design is critical for applications such as enzyme design (where sequences must accommodate a bound substrate), nucleic acid-binding protein design, and binder engineering. The input to TRILL is a PDB file that may contain protein chains alongside small molecules, nucleic acids, or metal ions. For batch processing of multiple structures, users can provide a JSON file listing PDB paths via `-pdb_path_multi`.

TRILL exposes an extensive set of LigandMPNN parameters for fine-grained control

over the design process. Five model variants are available via `--lig_mpnn_model`, each trained for a different context. The default ProteinMPNN variant performs general-purpose backbone sequence design. The Soluble variant is optimized for soluble proteins. The Global_Membrane and Local_Membrane variants are trained for membrane protein design with global and local membrane context respectively. The Side-Chain_Packing variant is trained for simultaneous sequence and side-chain conformation prediction. Each variant is available at five noise levels via `--lig_mpnn_noise` (002, 005, 010, 020, 030), where the number corresponds to the standard deviation of Gaussian noise in Angstroms added to backbone coordinates during training. Lower noise levels produce more conservative designs closer to the native sequence, while higher noise levels increase sequence diversity. Note that the 002 noise level is only available for the Soluble and Side-Chain_Packing models.

Residue-level control is provided through several complementary flags. The `--fixed_residues` flag specifies positions that should retain their native amino acid identity during design, using a chain-residue format such as "A12 A13 A14 B2 B25". Conversely, `--redesigned_residues` specifies which positions should be redesigned while fixing everything else. Both flags have corresponding `--fixed_residues_multi` and `--redesigned_residues_multi` variants that accept JSON files mapping per-PDB residue specifications for batch processing.

Amino acid composition can be controlled at both the global and per-residue level. The `--omit_AAs` flag globally excludes specified amino acids from all designed positions (for example, "AC" would prevent alanine and cysteine from appearing anywhere in the output). The `--omit_AA_per_residue` flag provides position-specific exclusions via a JSON mapping (for example, {"A12": "APQ"} would exclude alanine, proline, and glutamine at position 12 of chain A). The `--bias_AA` flag applies global compositional biases as comma-separated key-value pairs (for example, "A:-1.024,P:2.34,C:-12.34"), where negative values disfavor and positive values favor the specified amino acid. Per-residue biases are supported via `--bias_AA_per_residue` using JSON mappings.

For symmetric oligomer design, `--symmetry_residues` specifies groups of residues that should receive identical amino acid assignments, formatted as pipe-separated lists (for example, "A12,A13,A14|C2,C3"). Corresponding weights for each symmetry group are provided via `--symmetry_weights`. For the common case of homo-oligomer design with equal weighting across all chains, the `--homo_oligomer` flag automatically configures the symmetry settings.

ProstT5 ProstT5 (Heinzinger et al., 2024) offers a fundamentally different approach to inverse folding that does not require a neural network trained specifically on structure-to-sequence pairs. Instead, it leverages ProstT5's bilingual capability (described in Section 2.5) to translate 3Di structural tokens back into amino acid sequences. Given a protein structure in PDB format, TRILL first converts it to a 3Di string using Foldseek, then feeds the 3Di string to ProstT5's encoder and decodes an amino acid sequence from the decoder. Users can control the generation with sampling temperature (`-temp`), top-p nucleus sampling (`-top_p`), repetition penalty (`-repetition_penalty`), and a flag to switch from stochastic sampling to greedy decoding (`-dont_sample`).

Diffusion-based protein generation

Denoising diffusion probabilistic models (DDPMs) generate data by learning to reverse a gradual noising process. In the context of protein design, the forward process progressively corrupts protein backbone coordinates with Gaussian noise until the structure is indistinguishable from random noise, and the reverse (generative) process learns to iteratively denoise random coordinates into physically plausible protein backbones. TRILL supports two diffusion models through the `diff_gen` command.

RFDiffusion RFDiffusion (Watson et al., 2023) adapts the RoseTTAFold (RF) structure prediction network into a generative diffusion framework. The key insight is that the RF network, originally trained to predict protein structures from sequences and evolutionary information, can be repurposed as the denoising network in a DDPM by fine-tuning it to predict clean structures from noisy coordinate inputs. RFDiffusion represents each residue as a rigid-body frame (comprising a Cα coordinate and an N-Cα-C orientation) and applies 3D Gaussian noise to the coordinates and Brownian motion on the manifold of rotations during the forward process. Generation proceeds over a trajectory of typically 200 timesteps, with the model using a self-conditioning strategy in which its own previous prediction is recycled as additional input at each step. RFDiffusion supports several design modes in TRILL. Unconditional generation creates novel protein backbones from pure noise, with the user specifying the desired length range via `-contigs` (for example, `-contigs 100-200` generates proteins between 100 and 200 residues). Motif scaffolding holds a functional motif from an input PDB in place while generating a surrounding scaffold, specified via the `-query` flag for the input PDB and `-Inpaint` for the residues to hold fixed.

Partial diffusion starts from an existing structure and applies a controlled amount of noise (set by `-partial_T`) before denoising, producing structural variants of the input while optionally holding specified residues fixed via `-partial_diff_fix`. Binder design generates a protein that interacts with a specified target, with the user defining contact residues via `-hotspots`. Alternative RFDiffusion model checkpoints, such as the ActiveSite checkpoint for small motif scaffolding, can be selected via `-RFDiffusion_Override`.

Genie2 Genie2 (Lin et al., 2024) is a diffusion model for protein structure generation that uses asymmetric representations for the forward and reverse processes. The forward process applies isotropic Gaussian noise to Cα point clouds through a cosine variance schedule over 1,000 diffusion steps. The reverse process uses an SE(3)-equivariant denoiser that operates on clouds of reference frames rather than raw coordinates. The denoiser architecture consists of an SE(3)-invariant encoder that transforms per-residue and residue-residue pair features into single and pair representations, and an SE(3)-equivariant decoder that uses Invariant Point Attention to update reference frames. Genie2 introduces a multi-motif scaffolding framework that can design proteins incorporating multiple functional motifs with unspecified inter-motif positions and orientations. In TRILL, unconditional generation is controlled by `-contigs` for the desired length range and `-scale` for the sampling noise scale (default 0.6, where higher values inject more noise during the reverse process). For motif scaffolding, users provide an input PDB via `-query` and specify which regions to scaffold via `-motifs`, using a chain-start-end format (for example, `-motifs A-15-100 B-100-150` scaffolds two motifs simultaneously). The number of designs is set by `-num_return_sequences` (default 5). Both RFDiffusion and Genie2 output PDB-formatted backbone coordinates that can be passed directly to the `inv_fold_gen` command for sequence design, forming a complete structure-to-sequence generation pipeline within TRILL.

2.8 Structure prediction

Predicting the three-dimensional structure of a protein from its amino acid sequence is one of the central problems in computational biology. Accurate structure predictions enable downstream analyses such as molecular docking, dynamics simulations, binding site identification, and functional annotation. TRILL provides access to four structure prediction methods through the `fold` command, spanning single-sequence monomeric folding to multi-chain complex prediction with non-protein molecules.

Table 2.7 provides an overview.

Table 2.7: Structure prediction models available through the `fold` command.

Model	Capability	Input	Output	Key Features
ESMFold	Single-sequence protein structure prediction	FASTA	PDB	No MSA required, seconds per structure, chunking for long sequences
Chai-1	Multi-modal complex prediction (protein, ligand, nucleic acid)	FASTA with chain type annotations	PDB + confidence scores	AlphaFold3-style architecture, optional MSA via ColabFold
Boltz-2	Complex prediction with binding affinity estimation	FASTA with chain type annotations	PDB + confidence + affinity	Trained on structural ensembles, distance constraints, ~1000× faster than FEP
ProstT5	Structural alphabet prediction (3Di tokens)	FASTA	CSV of 3Di strings	Enables rapid structure-based search via Foldseek without full 3D prediction

ESMFold (Lin et al., 2023) is an end-to-end single-sequence protein structure prediction model that operates entirely on the internal representations of the ESM-2 language model, without requiring multiple sequence alignments (MSAs) or structural templates. The ESM-2 language model processes the input sequence through its transformer layers, and the resulting per-residue representations are passed to a folding trunk consisting of 48 blocks that alternate between sequence-level and pairwise attention updates. The folding trunk produces a predicted structure and per-residue confidence scores (pLDDT). Because ESMFold bypasses the computationally expensive MSA construction step that is required by AlphaFold2 and similar methods, it can predict a structure in seconds rather than minutes or hours, making it practical for high-throughput applications. The input to TRILL is a FASTA file of protein sequences. The output is a PDB file for each predicted structure. For proteins that exceed GPU memory, TRILL provides a `-strategy` flag that controls the chunk size used by the folding trunk (for example, passing 64 or 32 reduces peak memory at the cost of speed). TRILL also handles out-of-memory errors gracefully. If a protein is too long for the available GPU memory, TRILL logs the error and continues to the next sequence rather than terminating the entire run.

Chai-1 (Chai Discovery et al., 2024) is a multi-modal foundation model for biomolecular structure prediction developed by Chai Discovery. Its architecture largely follows AlphaFold3. The model processes these inputs through an MSA module (4 blocks), an input embedding module using pair-bias attention (4 blocks), a

structure prediction module based on diffusion (48 blocks), and 4 rounds of recycling through a 16-block refinement stage, followed by a confidence head (4 blocks). Chai-1 can predict structures for protein-protein complexes, protein-ligand interactions, and protein-nucleic acid assemblies from raw sequence and chemical composition alone. The model also supports single-sequence mode without MSAs, producing strong predictions by incorporating ESM-derived language model embeddings as an additional input track. The input to TRILL is a FASTA file formatted with chain type annotations following the Chai-1 convention (for example, `>A|protein` or `>B|smiles`). Users can enable MSA generation through the ColabFold server via the `-msa` flag, which may improve prediction quality for challenging targets. The output consists of predicted PDB structures and per-model confidence scores.

Boltz-2 (Passaro et al., 2025) is a structural biology foundation model that builds on its predecessor Boltz-1 with improved structure prediction accuracy and the additional capability of predicting binding affinity. The model extends its training data beyond static crystal structures to include experimental and computational structural ensembles from sources including the PDB, MISATO, ATLAS, and mdCATH molecular dynamics datasets. This training strategy exposes the model to local fluctuations and global structural ensembles rather than single equilibrium conformations only. Boltz-2 also introduces controllability features, including conditioning on experimental methods, user-defined distance constraints, and multi-chain template integration. For affinity prediction, Boltz-2 leverages the latent representation driving its cofolding process, since this representation inherently encodes information about the strength of biomolecular interactions. The model achieves binding affinity predictions that approach the accuracy of free-energy perturbation (FEP) methods while being at least 1000 times more computationally efficient. The input to TRILL is a FASTA file with chain type annotations following Boltz-2 conventions. As with Chai-1, users can enable MSA generation via the `-msa` flag. The output consists of predicted structures in PDB format along with confidence metrics.

ProstT5 (Heinzinger et al., 2024) serves a different purpose in the `fold` command compared to the three structure prediction models described above. Rather than producing three-dimensional atomic coordinates, ProstT5 translates amino acid sequences into 3Di structural alphabet tokens. The 3Di alphabet, used by the Foldseek (Van Kempen et al., 2024) structural search tool, encodes each residue's

local backbone geometry as one of 20 discrete states. These 3Di strings enable rapid structure-based database searches and remote homology detection through sequence alignment algorithms applied to the structural alphabet, bypassing the need for full three-dimensional coordinate comparisons. Because ProstT5 predicts 3Di tokens directly from amino acid sequences using its bilingual T5 architecture (described in Section 2.5), it provides a computationally lightweight alternative to running a full structure prediction followed by Foldseek conversion. The input is a FASTA file of protein sequences. The output is a CSV file containing the predicted 3Di string and sequence label for each protein. Users can adjust the batch size via `-batch_size`.

2.9 Molecular docking and simulation

TRILL integrates molecular docking and molecular dynamics simulation into a unified structural bioinformatics pipeline through two commands. The `dock` command predicts the preferred binding orientation of a ligand to a protein receptor. The `simulate` command performs molecular dynamics simulations with automated post-simulation analysis. Both commands share a common PDB preprocessing step implemented through PDBFixer, which standardizes input structures by replacing nonstandard residues, removing heterogens, adding missing residues and atoms, and adding hydrogens at pH 7.0. This automated structure preparation ensures that user-provided PDB files, including those generated by structure prediction models like ESMFold, are compatible with the downstream docking and simulation engines.

Molecular docking

The `dock` command provides access to five docking methods spanning deep-learning-based, physics-based, and protein-protein approaches. Input batching is supported through text files listing paths, enabling high-throughput screening workflows. Molecular file format conversions throughout the docking pipeline are handled by Open Babel (O'Boyle et al., 2011) and RDKit (Landrum et al., 2026).

For the physics-based docking engines (AutoDock Vina and Gnina), TRILL integrates automated binding pocket detection via FPocket (Le Guilloux, Schmidtke, and Tuffery, 2009) as a preprocessing step. FPocket identifies candidate binding sites on the protein surface by computing Voronoi tessellations and detecting clusters of alpha spheres, which are spheres that contact four atoms and indicate cavities or clefts on the protein surface. TRILL's integration proceeds in three stages through the BioExcel Building Blocks (biobb) library. First, `fpocket_run` scans the entire protein surface for pockets, configurable through the minimum alpha sphere radius (`-min_radius`,

default 3.0 Å), maximum alpha sphere radius (`-max_radius`, default 6.0 Å), and the minimum number of alpha spheres required to define a pocket (`-min_alpha_spheres`, default 35). Pockets are ranked by an internal druggability score. Second, `fpocket_filter` retains only pockets with scores between 0.2 and 1.0, eliminating low-confidence predictions. Third, `fpocket_select` extracts each filtered pocket's atoms and vertices, from which TRILL computes the box center and dimensions used to define the search space for Vina or Gnina. TRILL then docks the ligand into each detected pocket independently and reports results per pocket. Users who prefer to skip pocket detection and dock against the entire protein surface can use the `-blind` flag to perform blind docking instead.

DiffDock-L DiffDock-L (Corso et al., 2022) reframes molecular docking as a generative modeling problem rather than a search-and-score optimization. The key insight is that the standard docking evaluation metric (the fraction of predictions with RMSD below a threshold) is equivalent to maximizing the likelihood of the true pose under the model's output distribution. DiffDock therefore learns a model over the non-Euclidean product manifold of the degrees of freedom involved in docking, specifically the ligand's position relative to the protein (translation in $\mathbb{R}^3$), its orientation (rotation in SO(3)), and its internal conformation (torsion angles in the torus $\mathbb{T}^m$ for m rotatable bonds). The forward diffusion process progressively adds noise to each of these components, and the reverse process learns to denoise random ligand poses into binding configurations conditioned on the protein structure. After sampling, a separate confidence model ranks the generated poses and estimates the probability that each is within a given RMSD tolerance of the true binding mode. In TRILL, users can control the number of candidate poses via `-samples_per_complex` (default 10), the number of denoising steps via `-inference_steps` (default 20), and whether to inject noise in the final denoising step via `-final_step_noise`. The `-save_visualisation` flag saves a PDB file showing all intermediate steps of the reverse diffusion trajectory. DiffDock-L does not use FPocket because it performs its own learned pocket identification as part of the diffusion process.

AutoDock Vina AutoDock Vina (Eberhardt et al., 2021) is a widely used physics-based docking engine that combines a parameterized empirical scoring function with a stochastic global optimization algorithm based on iterated local search. The scoring function evaluates protein-ligand binding poses by summing contributions from

steric interactions, hydrophobic contacts, hydrogen bonding, and torsional penalties. Vina searches over the ligand's translational, rotational, and torsional degrees of freedom within a defined 3D search box on the protein surface. In TRILL, users can dock multiple ligands against a single receptor by providing them sequentially as arguments or via a text file with one ligand path per line. When FPocket pocket detection is enabled (the default), the search box for each docking run is automatically centered and sized based on the detected pocket geometry. The computational effort of the search can be adjusted via `-exhaustiveness` (default 8), which controls the number of independent search runs performed. Higher exhaustiveness values increase the probability of finding the global optimum at the cost of longer runtime.

Gnina Gnina (McNutt et al., 2025) augments the AutoDock Vina framework with convolutional neural network (CNN) scoring models that evaluate protein-ligand poses on a discretized 3D voxel grid. The CNN takes as input a voxelized representation of the protein-ligand complex, with separate channels encoding different atom types, and outputs both an affinity prediction and a pose quality score. During docking, Gnina uses Vina's iterated local search algorithm to generate candidate poses and then rescores them with the CNN, producing a CNN score, a CNN affinity estimate, and the traditional Vina score for each pose. This hybrid approach combines the sampling efficiency and physical grounding of Vina with the pattern recognition capabilities of the learned CNN model. In TRILL, Gnina accepts the same inputs, FPocket integration, and exhaustiveness parameter as Vina.

LightDock LightDock (Jiménez-García et al., 2018) is a macromolecular docking framework designed for protein-protein docking. LightDock represents the docking problem using glowworm swarm optimization (GSO), a nature-inspired metaheuristic where multiple agents ("glowworms") explore the binding pose landscape in parallel by communicating local fitness information. Each glowworm encodes a rigid-body transformation of one binding partner relative to the other, parameterized as a translation vector and a rotation quaternion. The swarm size, number of swarms, and number of simulation steps are configurable in TRILL via `-glowworms` (default 200), `-swarms` (default 25), and `-sim_steps` (default 100). Residue-level restraints can be provided via the `-restraints` flag to bias the search toward known or predicted interface regions. The input to TRILL is two PDB files representing the two protein partners.

GeoDock GeoDock (Chu et al., 2024) is a deep-learning-based protein-protein docking method built on a multi-track iterative transformer architecture inspired by AlphaFold2. The two input docking partners are represented as fully-connected graphs, with nodes carrying ESM-2 (650M) sequence embeddings and edges encoding inter-residue distances and relative positional orientations adapted from trRosetta and AlphaFold-Multimer. The architecture consists of two alternating modules that are recycled iteratively. The graph module updates node and edge representations through self-attention with pair bias, outer product difference operations, and triangular updates. The structure module then predicts backbone rotations and translations for each residue using Invariant Point Attention (IPA), similar to the structure module in AlphaFold2. The model was trained on the Database of Interacting Protein Structures (DIPS), comprising over 36,000 experimentally resolved binary protein complexes. Unlike LightDock's sampling-based approach, GeoDock directly predicts the docked complex structure in a single forward pass through the network, enabling sub-second inference on a single GPU. GeoDock is also flexible at the residue level, allowing the prediction of conformational changes upon binding. The input to TRILL is two PDB files representing the two protein chains.

After docking is complete, TRILL automatically generates interaction fingerprints for each predicted pose using ProLIF (Bouysset and Fiorucci, 2021). ProLIF encodes the protein-ligand contacts as binary fingerprints capturing specific interaction types, including hydrophobic interactions, hydrogen bonds (donor and acceptor), van der Waals contacts, pi-stacking, pi-cation interactions, salt bridges, and water bridges.

Molecular dynamics simulation

The `simulate` command performs molecular dynamics (MD) simulations using OpenMM (Eastman et al., 2017) with automated pre-processing, simulation execution, and post-simulation analysis.

System Setup TRILL constructs the simulation system using OpenMM's ForceField and Modeller classes. The default force field is Amber14 (`amber14-all.xml`), configurable via the `-forcefield` flag. Three solvent models are available via `-solvent`. The default is the Generalized Born implicit solvent model HCT (`implicit/hct.xml`), which approximates the effect of water without explicitly modeling solvent molecules, greatly reducing computational cost. For explicit solvent simulations, users can select TIP3P (`amber14/tip3p.xml`) or TIP3P-FB (`amber14/tip3pfb.xml`) water models, in which case TRILL uses the Modeller

to solvate the system within a periodic box of configurable size (`-periodic_box`, default 10 nm). When a small-molecule ligand is present, TRILL parameterizes it using the SMIRNOFF force field through the Open Force Field toolkit and registers the resulting parameters with OpenMM's ForceField via a template generator, ensuring that both the protein (Amber14) and ligand (SMIRNOFF/OpenFF) are parameterized within a single consistent system. Users can configure bond constraints (`-constraints`, with options None, HBonds, AllBonds, or HAngles), rigid water (`-rigidWater`), and the nonbonded interaction method (`-nonbonded_method`, with options including NoCutoff, CutoffNonPeriodic, CutoffPeriodic, Ewald, PME, and LJPME).

Simulation Execution The production simulation is configured with a user-specified step size (`-step_size`, default 2 femtoseconds) and number of steps (`-num_steps`, default 5,000). Trajectory frames are saved as PDB files at a configurable reporting interval (`-reporter_interval`, default 1,000 steps). TRILL provides NVT and NPT equilibration prior to production dynamics, with configurable equilibration steps (`-equilibration_steps`, default 300). For simulations that encounter numerical instabilities (manifesting as "Energy is NaN" errors from OpenMM), the `-rerun` flag specifies the number of automatic retry attempts before aborting.

Structure Relaxation For users who need only energy minimization without a full dynamics simulation, the `-just_relax` flag performs structure relaxation using the Amber14 force field and outputs the fixed and minimized structure. This is particularly useful for preparing predicted structures (from ESMFold or other methods) for downstream docking, as predicted structures often contain steric clashes or suboptimal bond geometries that should be resolved before use. The relaxation mode supports batch processing via a text file listing multiple PDB paths.

Post-Simulation Analysis After each simulation completes, TRILL automatically performs several analyses on the resulting trajectory.

RMSD and RMSF. MDAnalysis (Gowers et al., 2016) is used to compute structural dynamics metrics from the trajectory. The trajectory is first aligned to a reference frame using all atoms. RMSD (root-mean-square deviation) is then computed for all non-ion atoms (excluding sodium and chloride) across every frame, measuring how far the system has deviated from its starting conformation over time. A stable RMSD

plateau suggests the simulation has reached equilibrium, while a steadily rising RMSD may indicate unfolding or large-scale conformational changes. In addition to the whole-system RMSD, TRILL decomposes the RMSD calculation by protein chain, computing separate RMSD traces for the receptor and ligand (or for each chain in a multi-chain complex). This per-chain decomposition prevents the dynamics of one component from being washed out by the other, allowing users to assess, for example, whether a small-molecule ligand remains stably bound even if the receptor undergoes large conformational fluctuations, or vice versa. RMSF (root-mean-square fluctuation) is computed per backbone atom, quantifying the average positional variability of each residue over the trajectory. High-RMSF residues correspond to flexible loops, termini, or hinge regions, while low-RMSF residues indicate structurally rigid core regions.

Interaction Fingerprints. When a ligand is present, ProLIF (Bouysset and Fiorucci, 2021) generates per-frame interaction fingerprints from the trajectory. TRILL configures ProLIF to detect 15 interaction types that capture the major classes of non-covalent molecular recognition. These include anionic (negatively charged group near a positively charged partner), cationic (positively charged group near a negatively charged partner), cation-pi (cation interacting with an aromatic ring, detected in both directions), pi-stacking in both edge-to-face and face-to-face orientations, hydrogen bond donor and hydrogen bond acceptor interactions, hydrophobic contacts between nonpolar groups, metal coordination (metal ion acting as either acceptor or donor), van der Waals contacts, and halogen bond donor and acceptor interactions. The fingerprints are computed in count mode, recording how many interactions of each type are present between each protein residue and the ligand at every frame. The resulting dataframe provides a time-resolved contact map that reveals which protein-ligand interactions are persistent (present in most frames), transient (appearing and disappearing), or lost over the course of the simulation.

MM/GBSA Binding Free Energy. To estimate the thermodynamic favorability of each protein-ligand complex, TRILL performs MM/GBSA (Molecular Mechanics/Generalized Born Surface Area) calculations (Wang et al., 2019) using the AmberTools (Case et al., 2023) MMPBSA.py program. The calculation requires solvated and gas-phase topology files, which TRILL generates automatically using cpptraj to strip waters and ions and to remove periodic box information from the Amber parameter/topology files. The MM/GBSA calculation uses the igb=5 Generalized Born model (the OBC-II model) and decomposes the binding free energy

into contributions from van der Waals interactions, electrostatic interactions, polar solvation (GB), and nonpolar solvation (SA) terms. The resulting energy components (average, standard deviation, and standard error) are written to a CSV file for each complex.

Multi-Pose Batch Processing. When multiple ligand poses are provided (for example, from a prior docking run that outputs a multi-model PDBQT file), TRILL automatically splits the poses using Vina's split utility, converts each pose from PDBQT to MOL2 format, and simulates each one independently. All RMSD, RMSF, ProLIF, and MM/GBSA results are collected into consolidated master CSV files, enabling systematic comparison and ranking of docking poses by their simulated binding characteristics.

2.10 Classification, regression, and scoring

TRILL provides three commands for evaluating and predicting protein properties. The `classify` command supports both pre-trained property predictors from the literature and custom classifier training on protein embeddings. The `regress` command provides analogous functionality for continuous-valued predictions. The `score` command enables zero-shot evaluation of protein sequences and structures without any training data.

Pre-trained property predictors

The `classify` command provides direct access to eight pre-trained models from the literature, each targeting a specific protein property. The input for all pre-trained classifiers is a FASTA file of protein sequences. Table 2.8 provides a summary.

TemStaPro TemStaPro (Pudžiuvelytė et al., 2024) predicts protein thermostability from sequence alone using a multi-threshold ensemble approach. ProtT5-XL mean-pooled embeddings (1,024-dimensional) are fed into an ensemble of feed-forward MLP binary classifiers, each trained at a different temperature threshold ranging from 40°C to 80°C in 5-degree increments. For each threshold, five models trained with different random seeds are averaged, yielding a total of 45 model evaluations per query. The training data comprises over 2 million protein sequences from 21,498 organisms whose optimal growth temperatures are known. The output is a binary thermostability prediction at each temperature threshold along with a mean probability score, allowing users to assess not just whether a protein is thermostable but at what temperature range stability is predicted to break down. In TRILL, all

Table 2.8: Pre-trained property predictors available through the `classify` command.

Model	Predicted Property	Additional Input	Output
TemStaPro	Thermostability at 40–80°C thresholds	—	Binary stability prediction per threshold + probability scores
EpHod	Optimal enzyme pH	—	Continuous pH value (max seq. length 1,022)
CatPred	Enzyme kinetics (k_{cat}, K_M, K_i)	Substrate SMILES	$\log_{10}$ kinetic parameters + uncertainty estimates
CataPro	Enzyme kinetics (k_{cat}, K_M, k_{cat}/K_M)	Substrate SMILES	$\log_{10}$ kinetic parameters (single substrate)
ECPICK	EC number (enzyme function)	—	Hierarchical 4-level EC prediction + per-residue importance
M-Ionic	Metal ion binding sites	—	Per-residue binding predictions for 10 ion types
PSALM	Pfam domain annotation	—	Per-residue clan and family labels, domain boundaries
PSICHIC	Protein–small molecule interactions	Ligand SMILES	Affinity, interaction class, functional effect, binding fingerprint

model checkpoints are automatically downloaded and cached on first use.

EpHod EpHod (Gado et al., 2023) predicts the optimal pH for enzyme activity using a two-stage semi-supervised architecture. The first stage generates per-residue embeddings using the ESM-1v language model (650M parameters, 1,280-dimensional embeddings). The second stage processes these embeddings through a Residual Light Attention (RLAT) top model, which combines 1D convolution with attention-weighted pooling and max pooling to produce a 2,560-dimensional representation, followed by batch normalization, dropout, and four residual dense blocks terminating in a single continuous pH output. EpHod was trained on 9,855 enzymes from BRENDA with experimentally measured pH optima and pre-trained on 1.9 million secreted bacterial proteins mapped to environmental pH values. The model uses label-distribution-aware reweighting to handle the non-uniform pH distribution in the training data, where approximately 75% of pH optima fall between 6 and 8. Sequences longer than 1,022 residues are automatically filtered out due to the ESM-1v context length limit.

CatPred CatPred (Boorla and Maranas, 2025) predicts in vitro enzyme kinetic parameters (k_{cat}, K_M, and K_i) from enzyme sequence and substrate identity. The architecture is an ensemble of probabilistic deep learning regression models that use three feature learning modules of increasing complexity. Enzyme sequences are processed through sequence attention, pretrained pLM embeddings, and equivariant graph neural networks (E-GNNs) operating on AlphaFold2-predicted structures. Substrates are represented via directed message passing neural networks (D-MPNN) operating on molecular graphs. The model outputs Gaussian distributions (mean and variance) for each prediction, providing built-in uncertainty quantification decomposed into aleatoric and epistemic components. CatPred was trained on a curated database of approximately 23,000 k_{cat}, 41,000 K_M, and 12,000 K_i data points from BRENDA and SABIO-RK. In TRILL, users provide protein sequences via a FASTA file and substrate SMILES strings via a `-smiles` file, and CatPred returns $\log_{10}$ kinetic parameter predictions with per-query uncertainty estimates.

CataPro CataPro (Wang et al., 2025) also predicts enzyme catalytic parameters (k_{cat} and K_M) but uses a different architecture that combines ProtT5-XL embeddings (1,024-dimensional) for enzyme sequences with MolT5 embeddings (768-dimensional) and RDKit MACCS fingerprints (167-dimensional) for substrates. These features are concatenated into a 1,959-dimensional vector, compressed to 256 dimensions via a dense layer, and passed through a feature-wise attention layer before the final linear predictor. For catalytic efficiency (k_{cat}/K_M) prediction, a correction term network adjusts the ratio of independently pre-trained k_{cat} and K_M models to mitigate error propagation. CataPro was trained on 27,658 k_{cat} and 42,018 K_M entries from BRENDA and SABIO-RK, with training and test sets clustered at 40% sequence similarity by CD-HIT to prevent data leakage. As with CatPred, users provide both protein sequences and substrate SMILES, though CataPro accepts only single substrates.

ECPICK ECPICK (Han et al., 2023) predicts Enzyme Commission (EC) numbers from protein sequences using a biologically interpretable convolutional neural network with evidential deep learning. The architecture encodes the first 1,000 amino acids of each sequence as a one-hot matrix ($1{,}000 \times 21$), which is processed by three parallel convolutional layers with kernel sizes 4, 8, and 16 (128 filters each) followed by 1-max pooling to produce a 384-dimensional feature vector. This vector is then passed through a hierarchical classification module with dual global and local

flow branches that model the four-level EC number hierarchy, with a tunable beta parameter controlling the balance between the two branches. ECPICK was trained on 20 million protein sequences from UniProt, Swiss-Prot, PDB, and KEGG.

M-Ionic M-Ionic (Shenoy et al., 2024) predicts metal ion binding sites from protein sequences at residue-level resolution. The architecture uses frozen ESM2-650M per-residue embeddings (1,280-dimensional) as input to a network consisting of an input projection layer, a hidden block with LayerNorm and LeakyReLU activation, four Transformer encoder layers with 4-head self-attention, and 10 ion-specific output heads (one MLP per ion type). M-Ionic simultaneously predicts binding for 10 metal ion species (Ca, Co, Cu, Fe^{2+}, Fe^{3+}, Mg, Mn, PO_4, SO_4, and Zn) at each residue position. The model was trained on approximately 29,400 metal-binding proteins from BioLip, clustered at 20% sequence similarity to reduce redundancy. In TRILL, M-Ionic outputs two CSV files, one with per-protein metal ion counts and one with per-residue binary binding predictions for each of the 10 ion types.

PSALM PSALM (Sarkar, Krishnan, and Eddy, 2024) performs residue-level protein domain annotation using a hierarchical two-stage architecture operating on frozen ESM2-650M embeddings. The first stage (Clan Model) uses a bidirectional LSTM with dense layers to predict Pfam clan labels for each residue. The second stage (Family Model) uses a similar architecture to predict Pfam family labels, conditioned on clan predictions through a softmax-masked mapping matrix that enforces hierarchical consistency between clan and family assignments. PSALM was trained on Pfam 35.0 annotations spanning 19,632 families grouped into 655 clans. The model identifies domain boundaries, annotates multi-domain proteins, and detects intrinsically disordered regions, providing a deep-learning replacement for traditional profile HMM methods that can capture long-range residue dependencies.

PSICHIC PSICHIC (Koh et al., 2024) predicts protein-small molecule interaction fingerprints from sequence data alone, without requiring three-dimensional structural information. The architecture constructs two 2D graphs from the inputs. The ligand molecular graph is built from the SMILES string via RDKit with atom-type and physicochemical node features. The protein residue graph is built from the amino acid sequence with residue-type embeddings, physicochemical properties, and an ESM2-predicted contact map defining the edges. Three iterative physicochemical graph convolutional layers perform intramolecular message passing, physicochemical

constraint grouping (junction tree decomposition for the ligand, minCUT clustering for the protein), and cross-attention-based intermolecular interaction between grouped ligand and protein regions. The output includes binding affinity predictions, binary interaction classification, functional-effect prediction (agonist, antagonist, or non-binder), and an interpretable fingerprint identifying key binding-site residues and ligand atoms. In TRILL, users provide protein sequences via a FASTA file and ligand SMILES via a `-smiles` file.

Custom embedding-based classifiers

For tasks where no pre-trained model exists, TRILL enables users to train custom classifiers on protein language model embeddings. The workflow proceeds in three steps. First, TRILL embeds the input protein sequences using any of the supported embedding models (selectable via `-emb_model`, default ESM2-35M). Users can also provide pre-computed embeddings via `-preComputed_Embs` or use Pool PaRTI embeddings via the `-poolparti` flag. Second, a user-provided CSV key file (`-key`) maps sequence labels to class assignments. Third, TRILL trains and evaluates the selected classifier.

XGBoost XGBoost (Chen and Guestrin, 2016) is a gradient boosted decision tree algorithm. TRILL exposes key hyperparameters including learning rate (`-lr`, default 0.2), maximum tree depth (`-max_depth`, default 8), number of boosting rounds (`-n_estimators`, default 115), gamma for minimum loss reduction on leaf partitions (`-xg_gamma`, default 0.4), L1 regularization (`-xg_reg_alpha`, default 0.8), and L2 regularization (`-xg_reg_lambda`, default 0.1). The F1 score averaging method for evaluation is configurable via `-f1_avg_method` (with options macro, weighted, micro, or no averaging).

LightGBM LightGBM (Shi et al., 2026) is a gradient boosting framework that uses histogram-based algorithms for efficient tree construction. TRILL exposes hyperparameters including learning rate (`-lr`, default 0.2), maximum tree depth (`-max_depth`, default 8), maximum number of leaves per tree (`-num_leaves`, default 31), number of boosting rounds (`-n_estimators`, default 115), bagging frequency (`-bagging_freq`), bagging fraction (`-bagging_frac`), and feature sampling fraction (`-feature_frac`).

Isolation Forest Isolation Forest (Liu, Ting, and Zhou, 2008) is an unsupervised anomaly detection algorithm that identifies outliers by measuring how easily data points can be isolated through random partitioning. Unlike XGBoost and LightGBM, Isolation Forest requires only positive examples (or no labels at all), making it suitable for one-class classification tasks where "negative" examples are poorly defined or unavailable. The contamination parameter (`-if_contamination`) controls the expected proportion of outliers in the data.

MLP TRILL also supports training a standalone multilayer perceptron (MLP) classifier on pre-computed embeddings. Users can configure the hidden layer sizes (`-hidden_layers`, default [128, 64, 32]), dropout rate (`-dropout`, default 0.3), learning rate (`-lr`), batch size (`-batch_size_mlp`), and number of training epochs (`-epochs`, default 3).

ESM2+MLP For end-to-end fine-tuning, the ESM2+MLP option trains an ESM2 backbone jointly with a classification head, allowing the language model's representations to be adapted specifically for the classification task. This approach can be more powerful than training on frozen embeddings when sufficient labeled data is available, as the embedding space is optimized for the target task. Users configure the number of training epochs (`-epochs`, default 3) and learning rate (`-lr`).

Hyperparameter Optimization For XGBoost and LightGBM, TRILL integrates with the Optuna optimization library (Akiba et al., 2019) to perform automated hyperparameter sweeps. Enabling the `-sweep` flag triggers Bayesian optimization over the hyperparameter space for a configurable number of iterations (`-sweep_iters`, default 10). This is particularly valuable because optimal hyperparameters can vary significantly depending on the embedding model, dataset size, and classification task.

Trained classifiers are saved as serialized model files and can be reloaded for inference on new data via the `-preTrained` flag, decoupling the training and prediction steps.

Regression

The `regress` command provides regression functionality for predicting continuous-valued protein properties from embeddings. The workflow mirrors that of the custom classifiers. Users provide a FASTA file of sequences, a CSV key file mapping labels

to continuous values, and select an embedding model. Two regression algorithms are available.

Linear Regression Scikit-Learn (Pedregosa et al., 2011) linear regression fits a linear model to the embedding features. This provides a simple, interpretable baseline that is useful for assessing the predictive information content of different embedding models.

LightGBM Regressor LightGBM (Shi et al., 2026) is also available in regression mode, with hyperparameters including learning rate (`-lr`, default 0.2), maximum tree depth (`-max_depth`, default -1 for no limit), maximum leaves (`-num_leaves`, default 31), and number of boosting rounds (`-n_estimators`, default 115). Hyperparameter sweeps via Optuna are supported through the `-sweep` flag with cross-validation (`-sweep_cv`, default 3 folds).

Zero-shot scoring

The `score` command evaluates protein sequences or structures without any training data, using model-derived likelihoods or geometric complementarity as proxies for quality or fitness.

ESM1v and ESM2 Pseudo-Log-Likelihoods ESM1v and ESM2 (150M and 650M variants) can be used to compute pseudo-log-likelihood scores for protein sequences. For each position in the sequence, the model predicts the probability of the native amino acid given all other positions, and the scores are summed across the sequence. Sequences that are more "protein-like" according to the model's learned distribution receive higher scores. These scores serve as zero-shot fitness predictors and computational filters for protein design, as sequences with high pseudo-log-likelihoods are more likely to fold and function correctly.

ProteinMPNN and LigandMPNN Scoring ProteinMPNN (Dauparas et al., 2022) and LigandMPNN (Dauparas et al., 2025) score protein structures by computing the conditional log-likelihood of each amino acid given the backbone structure (and ligand context for LigandMPNN). The input is a PDB file (or a text file listing multiple PDB paths for batch scoring). Users can select the model variant via `-mpnn_model` (ProteinMPNN, LigandMPNN, Soluble, Global_Membrane, or Local_Membrane) and noise level via `-lig_mpnn_noise`. For membrane proteins, global transmembrane labels (`-global_transmembrane_label`) and per-residue transmembrane

annotations (`-transmembrane_buried`, `-transmembrane_interface`) can be provided, with batch specification supported via `-batch_transmembrane_csv`. For LigandMPNN scoring, the `-ligand_mpnn_cutoff_for_score` parameter (default 8.0 Å) controls the distance cutoff between protein and context atoms for selecting which residues are included in the reported score.

Shape Complementarity (SC) The SC scorer (Lawrence and Colman, 1993) evaluates protein-protein complexes by computing a shape complementarity statistic that measures how well the molecular surfaces of two binding partners fit together. The SC score ranges from 0 (poor fit) to 1 (perfect complementarity). The input is a PDB file containing a multi-chain protein complex. For complexes with more than two chains, users can specify which chains belong to each binding partner via `-chain_group_A` (for example, passing "ABC" for a six-chain complex ABCDEF would score the complementarity of chains ABC against chains DEF). The interface distance parameter (`-interface_distance`, default 4.0 Å) controls which surface atoms are included in the complementarity calculation.

2.11 Composability and declarative bioengineering

What distinguishes TRILL from individual model implementations is the ability to compose multi-step pipelines from independent commands, combining models and methods from different paradigms into workflows whose analytical power exceeds what any single tool could provide. Each command in TRILL produces standardized output files (CSV for embeddings and classifications, PDB for structures, FASTA for generated sequences) that can be directly consumed by subsequent commands. This standardized I/O contract means that any combination of commands can be chained together without format conversion, and that new models can be added to any command without disrupting existing workflows.

This composability enables a wide range of multi-step protein engineering campaigns. For example, a user could embed a set of known functional proteins with ESM2, train an XGBoost classifier to distinguish them from negative examples, fine-tune ProtGPT2 on the positive sequences to learn the family-specific sequence distribution, generate candidate proteins from the fine-tuned model, classify the generated sequences with the pretrained XGBoost model to filter for likely functional hits, and visualize both natural and generated proteins in an interactive UMAP embedding to assess coverage of the functional landscape. Each of these steps is a single TRILL command, and the entire workflow requires no Python programming. A

different user working on enzyme engineering might instead combine RFDiffusion to generate novel scaffolds around a catalytic motif, LigandMPNN to design sequences for the generated backbones conditioned on a bound substrate, ESMFold to predict the structures of the designed sequences, and the docking and simulation pipeline to evaluate binding affinity and stability. The key point is that users can freely substitute models within any step of these pipelines. If one embedding model performs poorly for a given classification task, it can be swapped for another without changing the rest of the workflow.

This composability is the practical foundation of the *declarative bioengineering* vision introduced in Section 2.2. While users currently specify each step explicitly, the modular command structure and standardized I/O lay the groundwork for future automation in which users could specify goals like "design thermostable variants of this enzyme that retain catalytic activity" and have the platform select and chain the appropriate methods.

Foldtuning as a case study in composability

The most compelling validation of TRILL's composable design came from an external user. Arjuna Subramanian independently developed a novel protein design strategy called Foldtuning (Subramanian et al., 2023) by chaining together TRILL's existing commands, without any modification to TRILL's source code. Foldtuning is an iterative pipeline that combines language model fine-tuning, sequence generation, structure prediction, and structural search into a closed loop. In each round, a generative protein language model (such as ProtGPT2) is fine-tuned on a set of training sequences, then used to generate a large batch of candidate sequences. The candidates are folded with ESMFold, and the predicted structures are compared to the target fold using Foldseek. Sequences whose predicted structures match the target fold but have low sequence identity to natural proteins are selected as the training set for the next round. Over successive rounds, this procedure drives the generative model to explore progressively more distant regions of sequence space while maintaining structural fidelity to the target fold.

Foldtuning was validated through the design of novel Barstar variants capable of inhibiting Barnase, an RNase, despite having no detectable sequence homology to any natural Barstar protein. This result demonstrates that TRILL's modular commands can be composed into genuinely novel design methodologies that produce experimentally relevant outcomes. The fact that this methodology was conceived and

executed entirely by an external user, using only TRILL's documented command-line interface, is a strong endorsement of the platform's accessibility and composability. As of TRILL v1.8.3, Foldtuning has been integrated into the `workflow` command, allowing users to run the full iterative pipeline with a single invocation rather than scripting each step manually.

2.12 Related work

The landscape of computational protein engineering tools has evolved considerably since TRILL's initial development. Several platforms exist that address portions of the problem space, and understanding their respective strengths and design choices helps clarify TRILL's position and contributions.

`bio_embeddings` (Dallago et al., 2021b) provides a pipeline for generating protein embeddings from multiple PLMs and visualizing them. It was among the earliest tools to democratize access to protein language model representations. However, it does not support fine-tuning, protein generation, structure prediction, docking, or molecular simulation, limiting its utility to the embedding and visualization stages of a protein engineering workflow.

ColabFold/Design (Mirdita et al., 2022) focuses on structure prediction and makes it highly accessible through Google Colab notebooks. ColabFold's key contribution is making AlphaFold2 and related structure prediction methods usable without local GPU infrastructure by leveraging cloud compute. However, it does not provide embedding, classification, generative, or simulation capabilities, and its cloud-dependent design means it cannot be used in air-gapped or proprietary environments.

Ovo is a more recent open-source ecosystem for de novo protein design, released in November 2025. Ovo addresses the fragmented tool landscape by consolidating models, workflows, data management, and interactive visualization into a scalable platform. Its architecture uses Nextflow-based workflow orchestration, an SQL database for storing designs and metadata, and both a web application and command-line interface. Ovo focuses primarily on the de novo structure-based design pipeline (scaffold design, binder design, diversification, and validation). However, Ovo does not include protein language model embedding, fine-tuning, language model-based sequence generation, or molecular dynamics simulation.

evedesign, released by the Marks lab at Harvard in March 2026, takes a method-agnostic approach to biosequence design by defining three composable operations (Generate, Score, and Transform) that work across any model type, including

MSA-based methods, language models, inverse folding models, and de novo 3D generators. evedesign introduces a standardized multi-level instance representation that simultaneously captures sequence, embedding, and 3D structure information, allowing outputs from one model to feed directly into another without reformatting. The platform includes a web interface at evedesign.bio that takes users from a target protein to orderable DNA. evedesign's emphasis on multi-objective optimization and lab-in-the-loop iteration makes it particularly suited to iterative experimental campaigns. However, evedesign focuses on the design-score-iterate loop and does not include molecular dynamics simulation, docking, or the distributed training infrastructure for fine-tuning large language models that TRILL provides.

ProteinMCP, also released in March 2026, represents an emerging paradigm in which an AI agent autonomously plans and executes protein engineering workflows. ProteinMCP integrates 38 protein design tools through the Model-Context-Protocol (MCP) and uses a large language model to interpret high-level scientific goals, select appropriate tools, and chain them into multi-step pipelines.

Individual model repositories (e.g., the ESM, RFDiffusion, or LigandMPNN codebases) provide full access to their respective models but often require significant technical expertise to install, configure, and run, and are not designed for interoperability with other tools. Each repository has its own input format conventions, dependency requirements, and execution patterns, making it difficult for non-specialists to chain multiple models into coherent pipelines.

TRILL occupies a distinctive position within this landscape. It is the broadest platform in terms of the paradigms it integrates, spanning protein language model embedding and fine-tuning, autoregressive and masked language model-based sequence generation, inverse folding, diffusion-based backbone generation, structure prediction, molecular docking, molecular dynamics simulation, property classification, regression, and zero-shot scoring. It is the only platform that combines deep-learning models with physics-based simulation and scoring within a single command-line interface. Its multi-modal embedding capability, spanning proteins, nucleic acids, and small molecules (Section 2.5), further distinguishes it from existing tools.

2.13 Software engineering and community

TRILL is released as open-source software under an open license and is available on GitHub (`https://github.com/martinez-zacharya/TRILL`). Documentation is hosted on ReadTheDocs (`https://trill.readthedocs.io`). As of this writing,

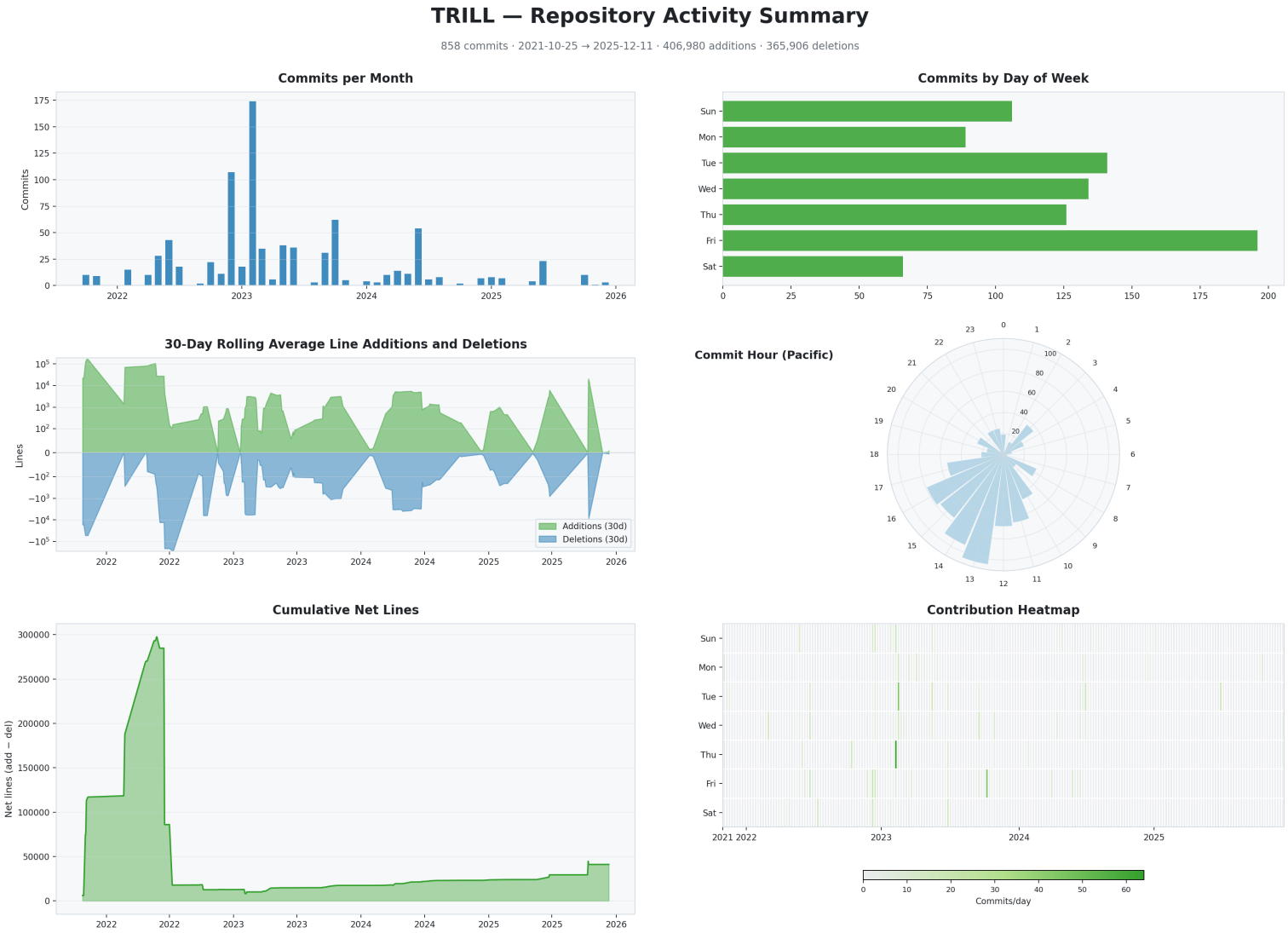


Figure 2.3: Summary of Work done on TRILL's GitHub Repository

the repository has accumulated 119 GitHub stars, and the preprint has been cited 9 times. The repository includes a SLURM example script for supercomputer deployment and comprehensive command-line documentation via `--help` flags on all commands and subcommands.

TRILL has only been made possible through building upon the efforts of open-source software development, highlighting the importance of open science and collaboration. As the field of deep-learning-based bioengineering matures, the real-world impact won't be fully realized without the sharing of methods and widespread adoption.

2.14 Future directions

Looking ahead, my goal is to continue actively refining and expanding TRILL, ensuring it remains current with emerging methods while pruning approaches that prove unreliable or are superseded through future benchmarking. The Foldtuning workflow command (Section 2.11) represents a first step toward the declarative paradigm described in Section 2.2, and several additional directions are planned.

1. **Continued model integration.** The field of protein engineering is advancing rapidly, with new models published regularly. TRILL will continue to incorporate the most impactful methods as they emerge, while also evaluating and potentially retiring models that no longer represent the state of the art.

2. **Automated, declarative workflows.** Expanding the `workflow` command to support more end-to-end declarative pipelines, where users specify goals and constraints rather than individual computational steps.

3. **Benchmarking.** Systematic comparison of the models and methods available within TRILL to provide users with evidence-based guidance on method selection for specific tasks.

4. **Testing.** Redesigning TRILL with a test-driven development strategy for easier maintenance.

5. **Community contributions.** Lowering the barrier for external contributors to add new models and workflows, building on the precedent set by the Foldtuning integration.

Perhaps the most compelling long-term possibility is the use of TRILL as a tool-use backend for autonomous AI agents. Large language models such as GPT-4 and Claude have demonstrated the ability to plan multi-step tasks, interpret scientific results, and invoke command-line tools through function calling or tool-use interfaces. Because TRILL exposes its entire functionality through a simple, text-based command-line interface with well-documented arguments and standardized output formats, it is naturally suited to be wielded by an AI agent rather than a human operator. An agent could, for example, accept a high-level objective like "design a thermostable variant of this enzyme that retains activity on substrate X," then autonomously plan and execute a multi-round campaign by embedding the wild-type sequence, scanning the literature for homologs, fine-tuning a generative model, producing candidate

sequences, predicting their structures, docking the substrate, running short molecular dynamics simulations, scoring the results, and iterating on failures. The agent could inspect intermediate outputs, decide which candidates to carry forward, swap models that underperform, and adjust hyperparameters based on observed outcomes. This vision represents the natural convergence of TRILL's declarative design philosophy with the planning and reasoning capabilities of frontier language models, and could change how computational protein engineering is practiced by transforming it from a human-operated pipeline into a goal-directed autonomous loop.

Chapter 3

APPLICATIONS OF TRILL TO PROTEIN DISCOVERY AND DESIGN

3.1 Demonstrating levels of workflow complexity achievable with TRILL

The preceding chapter described TRILL as a platform, including its design philosophy, command-line surface, integrated models, and supporting infrastructure. This chapter is its companion, where several examples of what is achievable with TRILL are exemplified.

The applications presented here are not exhaustive, and they were not chosen to maximize benchmark numbers. They are the case studies that, when stitched together, demonstrate the design claims of Chapter 2. Protein language model embeddings can support sensitive functional and homology-based classification, fine-tuning a generative protein language model can substantially increase the rate at which it is predicted to generate functional sequences, the same command-line surface scales from a few thousand training sequences to hundreds of millions of inference targets, and workflows composed from TRILL's primitives can be modified and recomposed by external users to address problems I never anticipated.

The chapter opens with a systematic benchmark, then turns to six application case studies. Section 3.2 uses TRILL to compare eight pretrained protein language models on two biophysical regression tasks, meltome-atlas melting point prediction and stability prediction, under a joint sweep of bit-quantization and PCA compression. Section 3.3 focuses on remote homology detection for the microvirus major capsid protein VP1, an early motivating problem for TRILL, with refreshed classifiers and a refined training-size and dimensionality sweep. Section 3.4 uses TRILL's `classify` command to train LightGBM classifiers for cellulase and antimicrobial activity, and applies them at the scale of NCBI's non-redundant database. Section 3.5 adapts the family-based generation workflow, originally developed to design cell-penetrating peptides and anti-CRISPR proteins, and documents both the legacy results from that effort and the principles that emerged. Section 3.7 describes "fast foldtuning", an extension to the Foldtuning algorithm introduced by Subramanian and colleagues (Subramanian et al., 2023) that swaps an expensive ESMFold step for a much cheaper ProstT5 translation, and benchmarks the substitution on the three-finger

protein (3FTx) fold. Section 3.6 presents a complementary cross-cutting comparison of ten protein language models on three classification tasks, providing pragmatic guidance on model selection for users of the `classify` command. Section 3.8 ties these threads together with the longest end-to-end TRILL pipeline demonstrated in this thesis, a defensive workflow that trains a toxin/superantigen/prion classifier on UniProt SwissProt, applies it to over 200 million proteins in NCBI's non-redundant database, and then takes seventeen predicted threats of hypothetical function from the order Araneae forward through ESMFold structure prediction, surface-patch identification, RFdiffusion binder design, LigandMPNN sequence design, ESMFold re-folding, GeoDock complex prediction, and OpenMM molecular dynamics simulation, with 892 designed threat–anti-threat complexes reaching the simulation stage.

Section 3.9 describes BioSentinel, an agentic cascade for detecting AI-designed toxin mimetics, a more recent, deliberately adversarial threat surface that classical sequence-alignment screening cannot address alone, combining a curated 15-keyword toxin training set, an end-to-end fine-tuned ESM2 classifier on a strictly homology-aware split, foldtuned-mimetic evaluation, and a ProstT5+Foldseek structural-alignment backstop. Section 3.10 steps back to identify the recurring patterns that emerged across all of these applications.

3.2 Benchmarking protein language models on biophysical regression tasks

Motivation and workflow

Performing benchmarks, a seemingly-easy task, can be deceptively challenging, in part because integrating multiple models, each with its own tokenizer, embedding API, and dependency stack, can be a substantial engineering effort. TRILL is well-suited to closing this gap. Every supported PLM is exposed through the same `embed` command, every embedding is written to the same on-disk format, and every downstream regressor or classifier consumes that format identically. A benchmarking experiment is therefore a loop over command-line arguments rather than a per-model rewrite.

To make this concrete, this section presents a case study comparing eight pretrained PLMs across two biophysical regression tasks, with the majority of the workflow, embedding extraction, regressor training, and evaluation, orchestrated through TRILL's command-line interface. One step, bit-level quantization of the embeddings, is performed outside TRILL. TRILL does not itself implement quantization but

accepts the resulting quantized vectors through its precomputed-embedding interface (`-preComputed_Embs`) and trains and evaluates regressors on them in exactly the same way as on raw embeddings.

Tasks

Melting point prediction (MPP) A sequence-level regression task in which the model must predict a per-protein melting temperature, drawn from the Meltome Atlas (Jarzab et al., 2020), a mass-spectrometry proteomics resource that reports the thermal stability of approximately 48,000 proteins across 13 species. We use the FLIP "mixed" splits (Dallago et al., 2021a), which cluster sequences at 20% identity and assign 80% of clusters to train and 20% to test, avoiding over-emphasis of large clusters. This setup tests whether embeddings capture global thermal tolerance signals driven by amino acid composition, hydrophobicity patterns, and overall fold stability across diverse proteins (12,144 total samples).

Stability prediction A second sequence-level regression task drawn from the dataset of Rocklin and colleagues (Rocklin et al., 2017), in which both de novo computationally designed minibinder and natural proteins were synthesized in yeast and exposed to proteolytic gradients. Stability is defined as the difference between a protein's measured EC_{50} (the protease concentration at which half of folded molecules survive) and the predicted EC_{50} of the same sequence in its unfolded state, on a $\log_{10}$ scale. The training and validation sets span a broad selection of designed topologies, while the test set consists of single-codon mutation neighborhoods around the most stable candidates (69,063 total samples). The split structure is therefore intentionally asymmetric, with train and validation probing sensitivity to fold class, whereas test probes sensitivity to single-residue substitutions on stable scaffolds.

Sweep, quantization, and evaluation

For each (model, task) pair, the workflow sweeps 45–55 compression configurations spanning raw embeddings at five bit precisions (2, 4, 8, 16, and float32), PCA compression to powers-of-2 component counts ranging from 2 up to the model's native dimensionality, and the full Cartesian product of (PCA-components, bit-precision) combinations. Bit-quantization is implemented as a uniform mapping of each feature's training-set range to 2^b uniform levels and is fitted on the training set only. The quantized embeddings are passed to TRILL via `-preComputed_Embs`, and the `regress` command trains a LightGBM regressor (Shi et al., 2026). All

sweeps use random seed 713. The result is a comprehensive map, for every PLM and task, of how embedding quality degrades under compression and where the Pareto frontier of accuracy versus storage cost lies.

The eight models tested were ESM2-8M, ESM2-35M, ESM2-650M, ProtT5-XL, ProstT5, Ankh, the codon-level model CaLM (Outeiral and Deane, 2024), and the messenger-RNA model mRNA-FM (Chen et al., 2022). Including two nucleotide-level models alongside six protein-level models is a deliberate design choice that lets the benchmark probe whether codon-usage and mRNA-level information carries signal that residue-level PLMs cannot recover.

Melting point results

Model comparison At full float32 precision, CaLM leads with $R^2 = 0.835$ and Spearman $\rho = 0.775$, followed by mRNA-FM ($R^2 = 0.779$, $\rho = 0.727$) and ProtT5-XL ($R^2 = 0.750$, $\rho = 0.699$). The two nucleotide-level models out-perform every protein-level PLM on a global biophysical property. The most plausible interpretation is that melting temperature is dominated by aggregate sequence-composition signals (amino-acid frequencies, hydrophobicity, codon usage, expression-level proxies) that codon-level representations capture directly, while residue-level masked language models recover them only indirectly through downstream pooling. Conversely, structure-conditioning hurts on this task. ProstT5 reaches RMSE 9.74 versus 6.74 for its parent ProtT5-XL, suggesting that 3Di-derived features overwrite the sequence-level signals relevant to thermostability.

Quantization Moderate bit-quantization actually improves test RMSE for six of the eight models. At 4-bit precision (16 uniform levels per feature), CaLM improves from RMSE 5.477 to 5.458, ProtT5-XL from 6.740 to 6.692, and ESM2-650M from 6.885 to 6.831. The improvements are small but consistent and suggest that quantization is acting as a mild regularizer by removing high-precision noise. The 2-bit level (four levels per feature) marks a consistent degradation threshold across all models (3–10% RMSE increase), indicating that while many embedding dimensions carry redundant precision, the underlying signal still requires at least four distinct levels per feature.

Dimensionality reduction PCA compression is remarkably effective on this task. CaLM retains 98.6% of its R^2 at just 16 PCA components (a reduction from 768 to 16 dimensions). ESM2-650M retains 96.7% at 16 components, and mRNA-FM retains

98.2%. Most models plateau by 32–64 components, indicating that the melting-point-relevant signal occupies a low-dimensional subspace consistent with the task being driven by aggregate physicochemical properties rather than residue-level features.

Joint compression Combining PCA with bit-quantization yields extreme storage reductions while remaining within 5% of peak RMSE. mRNA-FM achieves 1,280× compression (PCA-8 + 4-bit), CaLM achieves 384× (PCA-16 + 4-bit), and ESM2-650M achieves 320× (PCA-32 + 4-bit). These ratios matter operationally. A dataset that would otherwise require gigabytes of float32 storage fits in megabytes without meaningfully sacrificing predictive accuracy.

Stability results

Train–test distribution shift The Rocklin stability task exhibits a systematic distribution shift that is worth flagging before reading the numbers. The train and validation sets span diverse de novo–designed topologies, while the test set consists of single-codon mutation neighborhoods around high-stability candidates. This asymmetry manifests empirically. For most models, test RMSE is *lower* than validation RMSE (for example, ProtT5-XL records validation RMSE 0.413 and test RMSE 0.292), reflecting the compressed target variance of the local-neighborhood test regime. For CaLM, however, validation $R^2 = 0.487$ but test $R^2 = -0.687$, and mRNA-FM records validation $R^2 = 0.484$ but test $R^2 = -1.315$. A negative test R^2 does not mean the predictions are uninformative, they remain rank-correlated with the target according to ρ, but it does mean the predictions are on the wrong absolute scale and offset relative to the test distribution. Spearman ρ, which depends only on rank order and is invariant to strictly monotone transforms, is the reliable metric here. All stability comparisons below prioritize ρ.

Model comparison The rankings invert almost completely relative to melting point. ProtT5-XL leads ($\rho = 0.811$), followed by ProstT5 ($\rho = 0.764$), ESM2-35M ($\rho = 0.727$), ESM2-650M ($\rho = 0.723$), and Ankh ($\rho = 0.720$). CaLM falls to last ($\rho = 0.481$) and mRNA-FM to second-to-last ($\rho = 0.663$). The inversion is interpretable in terms of what each task demands. Stability prediction on the Rocklin test set requires sensitivity to single-residue substitutions in the context of a stable fold, precisely the regime that residue-level masked language modeling is trained to capture. ProtT5-XL and ESM2, both trained directly on protein sequences with residue-level objectives, encode the structural grammar needed to rank point

mutations. CaLM is in fact the only model whose test ρ drops below its validation value (0.607 to 0.481), suggesting that it generalizes poorly specifically to the local-neighborhood regime. ProstT5's structure conditioning helps here relative to melting point (rank seven on MPP to rank two on stability) but it still underperforms its parent ProtT5-XL, suggesting that 3Di tokens are useful for stability.

Compression on stability is harder PCA is far more destructive on stability than on melting point. At 16 components, ProtT5-XL retains 93.1% of its Spearman ρ (versus 95.5% R^2 retention on melting point), but CaLM retains only 46.6% (versus 98.6%) and ESM2-35M only 74.5% (versus 97.5%). Many models continue improving beyond 512 PCA components on stability, whereas on melting point most plateau by 32–64. Maximum compression within 5% of best RMSE drops dramatically, with 4–8× for most protein-level models on stability versus 100–1280× on melting point. Only CaLM (96×) and mRNA-FM (40×) tolerate substantial compression on stability, and only because their already-poor baseline performance leaves less accuracy to lose. The intuition is straightforward. Stability is encoded in high-dimensional patterns of residue interactions that cannot be flattened into a handful of principal components without losing the local sensitivity the task requires.

Cross-task discussion

Figure 3.1 summarizes the storage–accuracy Pareto frontiers for the eight models across both tasks. Two observations cut across both tasks. First, the right model depends on the task in a way that is not captured by parameter count or model family alone. CaLM, the smallest model in the sweep at 86M parameters, leads on melting point and ranks last on stability. The principal axis distinguishing the two tasks is whether the relevant signal is aggregate (composition, hydrophobicity, codon usage) or local (residue-level). Second, quantization is almost always free and can even be beneficial. 4-bit precision is comparable float32 for nearly every (model, task) combination, and joint quantization with PCA opens up compression ratios in the hundreds without meaningful accuracy loss on tasks dominated by aggregate signal. Third, the same workflow trivially supports adding a new model or a new task.

3.3 Remote homology detection for microviral major capsid proteins

Background

Microviridae are abundant single-stranded DNA bacteriophages, and yet their taxonomy was, until somewhat recently, treated as a single flat family containing tens

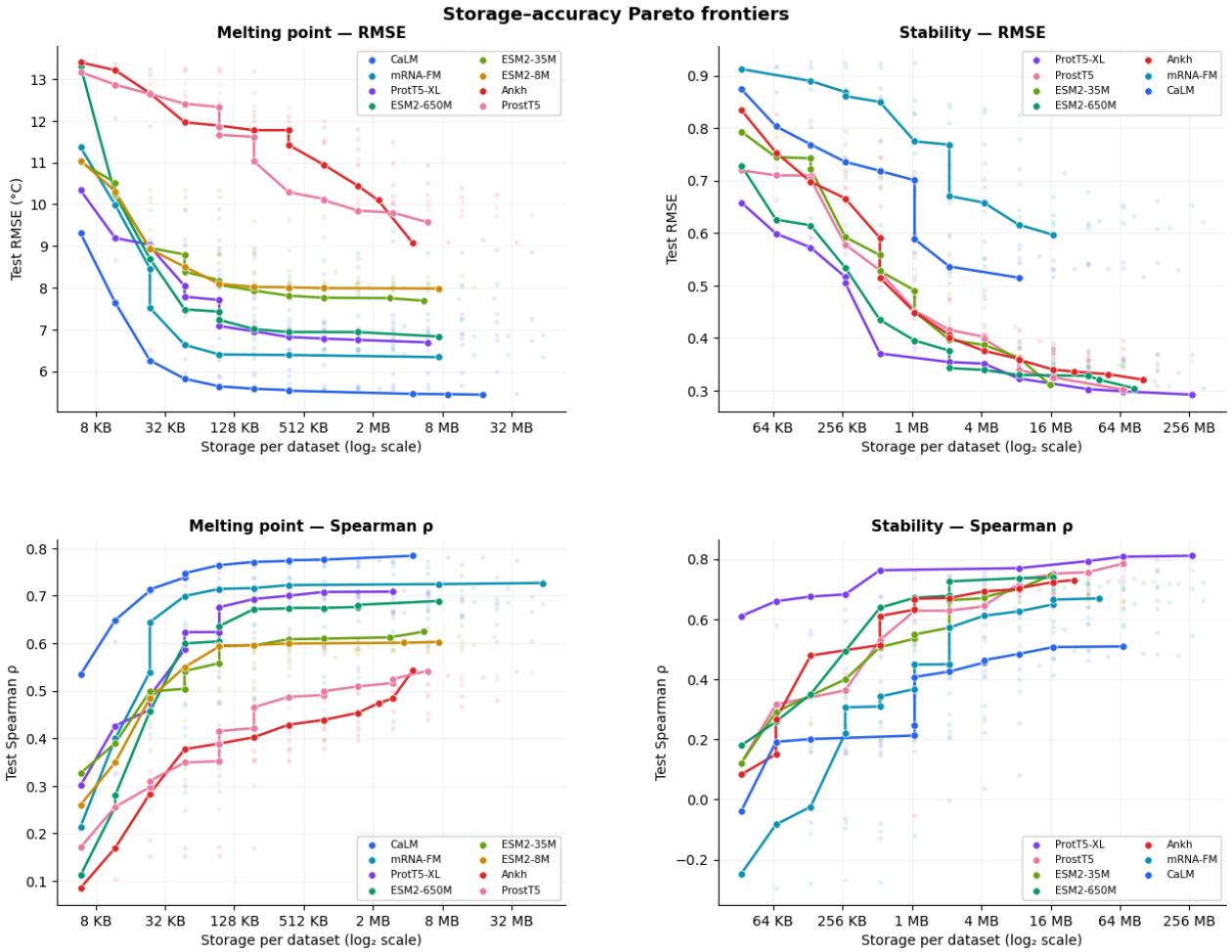


Figure 3.1: Storage–accuracy Pareto frontiers for eight protein language models across the melting-point (left) and stability (right) regression tasks. **Top row**. Test RMSE versus storage per dataset on a $\log_2$ scale. **Bottom row**. Test Spearman ρ versus storage per dataset. Each colored line traces the Pareto-optimal configurations for one model across the 45–55 (PCA, bit-precision) compression configurations tested. Faint background points show all sweep configurations.

of thousands of poorly characterized genomes (Kirchberger, Martinez, and Ochman, 2022). Reorganizing this diversity required identifying the major capsid protein, VP1, in candidate genomes, a problem that is non-trivial because microviral sequences mutate and diverge rapidly, and frequently undergo horizontal exchange. Traditional sequence-similarity tools such as BLAST (Altschul et al., 1990) miss many VP1s because their pairwise identity to known examples falls below detectable thresholds, and profile-based methods built on hidden Markov models (Eddy, 2011) require iterative searches that, in our earlier work(Kirchberger, Martinez, and Ochman, 2022), consumed months of CPU time.

The question I set out to answer with TRILL was whether protein language model embeddings could replace this iterative HMM machinery with a single forward pass through an encoder followed by a classifier trained on a small held-out fraction of known VP1s. The original results, published with the preprint (Martinez, Murray, and Thomson, 2023), used ESM2-3B embeddings together with XGBoost and Isolation Forest classifiers and reported F1 scores of 0.999 and 0.950, respectively. Here, I revisit the problem with a refreshed pipeline that (i) substitutes LightGBM for XGBoost, in line with TRILL's current default for new classification work, (ii) sweeps both classifier choice and embedding dimensionality through PCA compression, and (iii) varies the number of positive training sequences across nearly three orders of magnitude to characterize the data efficiency of the approach.

Data curation and experimental design

The positive set comprised 4,007 microviral VP1 sequences, drawn from the curated collection assembled in (Kirchberger, Martinez, and Ochman, 2022). The negative set was constructed by downloading all viral proteins from NCBI RefSeq, filtering out any sequence with detectable VP1 homology by HMM search, and clustering the remainder with MMseqs2 `linclust` (Steinegger and Söding, 2017) to remove near-redundant proteins. This procedure yielded a non-redundant pool of 130,279 viral proteins that are not VP1s. All 4,007 + 130,279 sequences were embedded with TRILL's `embed` command using ESM2-650M (Lin et al., 2023) and averaged across residue positions to obtain a single 1,280-dimensional vector per sequence.

The VP1 sequences were partitioned into three pools, namely a training pool (30%, n = 1,202), a held-out test set (20%, n = 801), and a supplementary test pool (50%, n = 2,004). Training-pool sequences that were not drawn into a given training subsample were merged into the test set, yielding roughly 2,800–4,000 test positives

at each training size depending on how many positives the subsample consumed. A matched 20% holdout of the non-VP1 sequences (n = 26,055) served as the negative test set. All results below report the mean and 5th–95th percentile interval across 50 independent random subsamples drawn at each training size, so that the reported uncertainty reflects sampling variability in the training partition rather than test-set noise alone.

Classifiers and embedding compression

Two TRILL classifiers were evaluated, reflecting the two qualitatively different regimes that arise in protein-classification practice. The first, an Isolation Forest (Liu, Ting, and Zhou, 2008), is an unsupervised anomaly detector trained only on positive (VP1) examples. It can be a appropriate choice when "negative" is poorly defined or when the user wishes to avoid making strong assumptions about what counts as non-VP1. The second, a LightGBM gradient-boosted decision tree classifier (Shi et al., 2026), is a supervised binary classifier trained jointly on VP1 and non-VP1 examples and represents the choice when high-quality negatives are available. Both classifiers are exposed directly through the `classify` command and require no Python beyond the user's choice of embedding model.

Each classifier was evaluated on the full 1,280-dimensional ESM2 embeddings and on a PCA-reduced representation retaining 99% of the explained variance (125 dimensions, with PCA fitted on the inlier pool for the Isolation Forest and on the combined training pool for LightGBM). The number of training VP1s was swept across 5, 8, 10, 15, 20, 30, 50, 100, 200, and 300 examples, with LightGBM trained on an equal number of randomly drawn non-VP1 negatives at each setting. A complementary lower-resolution sweep over more aggressive PCA levels (PCA95% through PCA25%, dropping to as few as one principal component) is summarized at the end of the section.

Sensitivity, specificity, and reliability scaling with training size

Figure 3.2 compares the two classifiers on four axes that together describe their behavior under varying labeled-data budgets, namely VP1 recall (sensitivity), non-VP1 rejection rate (specificity), overall accuracy, and the width of the 90% bootstrap confidence interval on VP1 recall (reliability).

LightGBM supervised classifier The supervised classifier is strikingly data-efficient. With only five labeled examples per class, VP1 recall is already 91.5%,

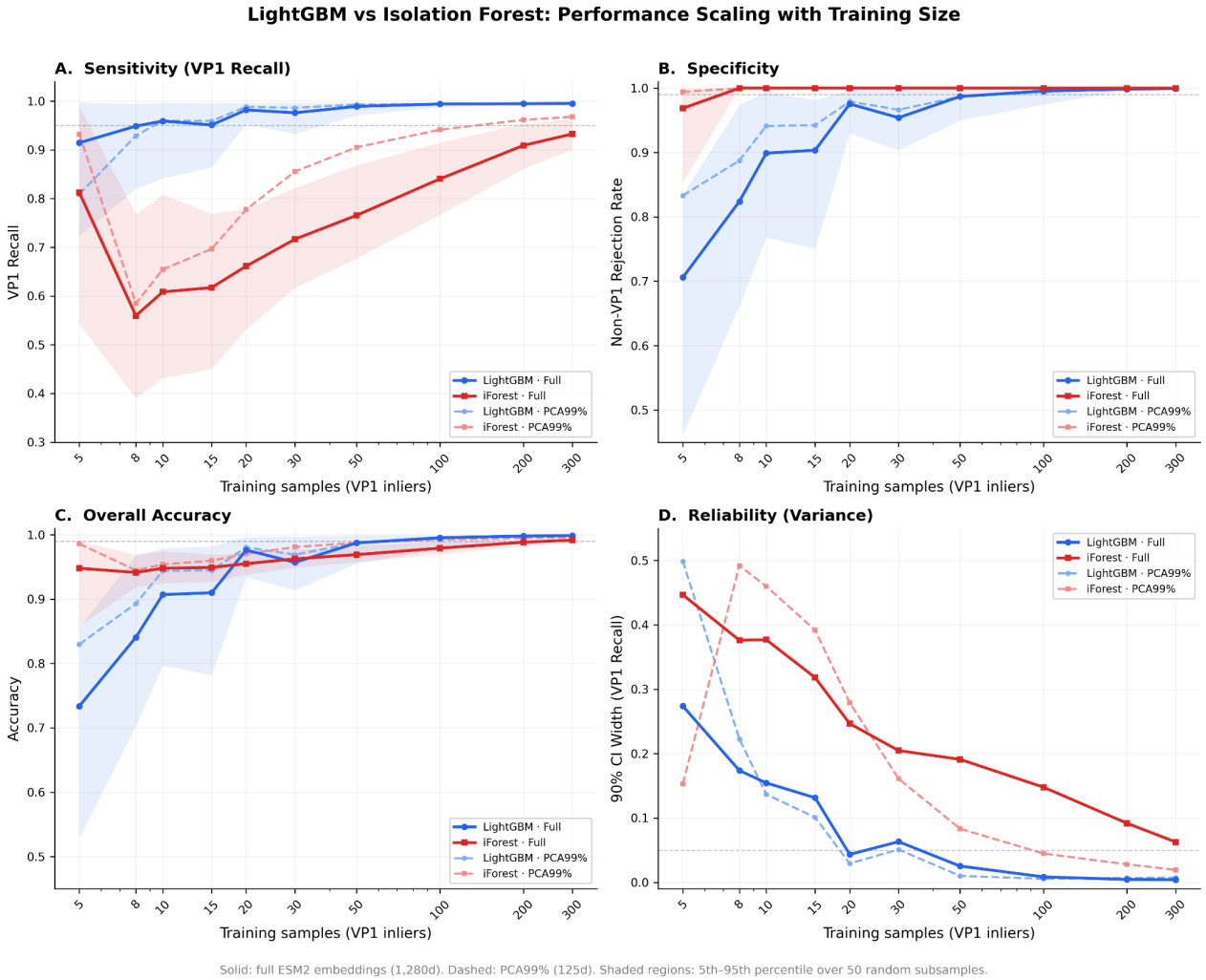


Figure 3.2: LightGBM (supervised, blue) vs. Isolation Forest (one-class, red) performance on VP1 detection as a function of training-set size. Solid lines, full 1,280-dimensional ESM2-650M embeddings. Dashed lines, PCA99% (125 dimensions). **Panel A**. VP1 recall (sensitivity). **Panel B**. Non-VP1 rejection rate (specificity). **Panel C**. Overall accuracy. **Panel D**. 90% confidence-interval width on VP1 recall, computed across 50 random training subsamples per setting. Shaded regions in panels A–C show the 5th–95th percentile interval.

crossing 95% by 10 labeled examples and 99% by 100. AUC-ROC exceeds 0.95 at five training examples and reaches 0.999 by 50. Specificity starts at 70.6% with five examples, where the classifier casts a seemingly wide net to capture all positives, and reaches 97.5% by 20 examples and 99.5% by 100. At the largest training size tested (300 per class), LightGBM achieves 99.5% VP1 recall, 99.9% specificity, 99.9% accuracy, and AUC 0.999, with a 90% CI width on recall of only 0.4 percentage points across 50 random training subsamples. PCA99% reduction yields nearly identical performance (99.4% recall, 99.9% specificity at $n = 300$), with slightly different calibration at small training sizes but comparable AUC throughout.

Isolation Forest anomaly detector The Isolation Forest exhibits a qualitatively different scaling profile, and one of the most informative features of the figure is its non-monotonic recall curve. With fewer than five training VP1s, the detector fails outright, classifying every sequence as an inlier (zero outlier detections across all 50 subsamples). At five training examples, recall jumps to a mean of 81.3% but with enormous variance (5th–95th percentile spanning 54.3%–98.9%). At eight training examples, recall *drops* to 56.0%, a consequence of the forest learning an overly tight inlier boundary around the few training points and rejecting most genuine VP1s that differ from those specific examples, and only gradually recovers as more training data widens the boundary. Recall reaches 76.6% at 50 examples, 84.1% at 100, and 93.3% at 300 with full embeddings. Specificity, by contrast, is near-perfect at all functional training sizes. From eight examples onward the Isolation Forest rejects essentially 100% of non-VP1 proteins.

PCA99% compression substantially improves the Isolation Forest. With 125-dimensional input, VP1 recall at 300 training examples increases from 93.3% to 96.8% (5th–95th percentile 95.8%–97.7%), with specificity unchanged above 99.9%. The improvement is consistent across all training sizes. At 50 examples, PCA99% delivers 90.5% recall versus 76.6% for the full embeddings. This contrast with LightGBM, where PCA had little effect, is informative. LightGBM performs implicit feature selection through its gradient-based split criterion and largely ignores uninformative dimensions, while the Isolation Forest selects split features uniformly at random. In 1,280 dimensions, the majority of its splits fall along uninformative axes, diluting the anomaly score. PCA pre-filtering concentrates the forest's splitting capacity on dimensions that actually describe the VP1 manifold.

Reliability Panel D quantifies how reliably each classifier produces a given recall under repeated training-set draws. LightGBM's 90% CI width on VP1 recall is 27 percentage points at $n = 5$, narrows to under 5 points by $n = 15$, and falls below 2 points by $n = 50$. The Isolation Forest's CI is 45 points at $n = 5$, remains at 38 points at $n = 8$ (the local-minimum point in the sensitivity dip), and is still 6 points at $n = 300$ with full embeddings. PCA99% brings the Isolation Forest CI down to 2 points at $n = 300$ but it remains wider than LightGBM at every training size below 100. This roughly 20-fold gap in data efficiency, where LightGBM hits 95% recall at $n = 10$ and the Isolation Forest needs $n \approx 200$ to do the same with PCA99%, reflects a fundamental architectural difference. The supervised classifier defines its decision boundary extrinsically using both positive and negative examples, while the Isolation Forest must infer the boundary from the inlier distribution alone.

Embedding-space compressibility

The training-size analysis above varied PCA only between FULL and PCA99%. A complementary lower-resolution sweep across more aggressive compression levels (PCA95%, PCA90%, PCA75%, PCA50%, and PCA25%, dropping to as few as one principal component) is summarized in Figure 3.3.

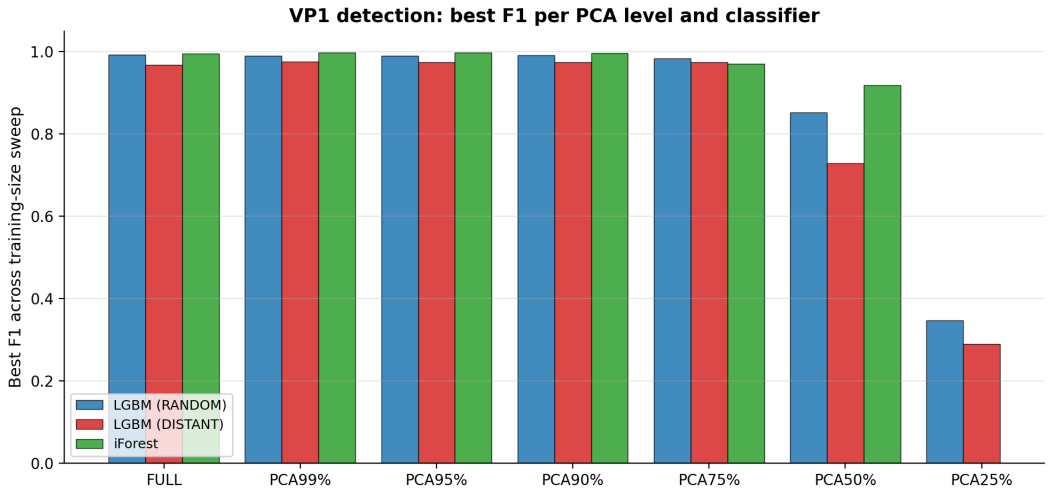


Figure 3.3: Best F1 achieved across the training-size sweep for each (classifier, PCA-level) pair.

The picture confirms and extends the PCA99% finding. Retaining 90% of the explained variance reduces the representation to roughly 16 dimensions for the Isolation Forest while keeping F1 within 0.03 of the FULL representation. Even PCA75% (5–26 dimensions depending on the classifier) retains F1 above 0.96. Substantial degradation appears only at PCA50% (1–5 dimensions) and below. This

embedding-space compressibility matters operationally. Downstream pipelines that pass embeddings into clustering, visualization, or out-of-core nearest-neighbour infrastructure can use 50–100× smaller representations without measurably sacrificing classifier accuracy.

Discussion

The combination of mean-pooled ESM2 embeddings with a TRILL-trained classifier recovers an annotation problem that previously required iterative HMM searches over a structured database. Beyond reproducing this headline result from (Martinez, Murray, and Thomson, 2023), the rigorous sampling design (50 random subsamples per training size, full sensitivity/specificity/reliability breakdown) reveals practical lessons.

Choose a classifier by what kind of error costs more The two classifiers fail in opposite directions at small training sizes. LightGBM with five labeled examples per class achieves 91.5% sensitivity but only 70.6% specificity, casting a wide net that catches most VP1s but admits many false positives. The Isolation Forest with eight training examples achieves essentially 100% specificity but only 56.0% sensitivity, drawing a tight boundary that excludes all non-VP1 proteins but rejects most VP1s. As training data grows, both classifiers converge toward the same high-performance regime, but from opposite directions, with LightGBM refining its specificity and the Isolation Forest expanding its sensitivity. The practical consequence is that when false positives are more costly than false negatives, curating a high-confidence reference database for instance, the Isolation Forest's conservative behavior at small training sizes is advantageous. When false negatives are more costly, as in metagenomic discovery, LightGBM's early sensitivity might be preferable.

Supervised classification is roughly twenty times more data-efficient LightGBM reaches greater than 95% VP1 recall at only 10 labeled examples per class, while the Isolation Forest requires approximately 200–300 VP1 examples to approach 95% recall with PCA-reduced embeddings, and never crosses this threshold with full embeddings within the training-size range tested. When negative examples are even loosely available, the supervised path can deliver a usable classifier from a handful of curated positives.

3.4 Predicting convergent functions at genomic scales

Motivation

Two protein properties of broad biotechnological interest are cellulase activity (the ability to hydrolyze the β-1,4 glycosidic bonds of cellulose, central to biomass-to-biofuel pipelines) and antimicrobial activity (the ability of proteins to disrupt microbial membranes or essential processes, central to antibiotic discovery in the face of growing resistance). For both properties, the experimentally validated examples available in curated databases are vastly outnumbered by the much larger pool of uncharacterized natural sequences. The question for this section is whether a TRILL-trained classifier can, from a manageable training set, achieve high enough precision to make scanning a database of hundreds of millions of proteins useful as a discovery filter.

Cellulase training data

The cellulase positive set was constructed by downloading all reviewed proteins under 6,000 amino acids from UniProt annotated with Enzyme Commission number 3.2.1.4 (cellulase) and removing any entry whose annotation contained "partial" or "fragment", yielding 36,235 unique sequences. The negative set was chosen deliberately to make the task difficult. Rather than solely randomly sampling unrelated proteins, I assembled a roughly equal-size pool of non-cellulase enzymes drawn from the closely related EC neighbourhoods around cellulase, namely the broader hydrolase class (EC 3.-.-.-), the glycosylase subclass (EC 3.2.-.-), and the O- and S-glycosyl hydrolases (EC 3.2.1.-). The motivation is that a classifier trained against these near-neighbours must learn features specific to cellulose hydrolysis rather than coarse signals such as "is a glycosylase."

To prevent test-set leakage through homology, I ran an all-versus-all MMseqs2 search (`easy-search`; Steinegger and Söding, 2017) on the positive and negative pools separately and split the training and test partitions such that no training sequence shared more than 10% sequence identity with any test sequence.

The resulting training set is small, with 1,364 cellulase positives and 1,461 non-cellulase negatives. The held-out evaluation pool, reported in Table 3.1 and Figure 3.4, comprises 36,897 cellulases and 37,592 non-cellulase negatives, totalling 74,489 sequences. The TRILL `classify` command was used end-to-end. ESM2-650M (Lin et al., 2023) embeddings were generated via `embed`, and a LightGBM (Shi et al., 2026) binary classifier was trained.

Antimicrobial training data

For antimicrobial prediction, positives were aggregated from three complementary sources, namely the Antimicrobial Peptide Database (APD6), the Database of Antimicrobial Activity and Structure of Peptides (DBAASP), and UniProt SwissProt entries carrying the antimicrobial-activity keyword KW-0929. After deduplication across these sources, the positive set contained 11,445 unique antimicrobial sequences. Negatives were drawn from UniProt SwissProt by retaining every reviewed protein under 5,000 amino acids that did *not* carry KW-0929, an initial pool of 113,414 sequences, and then reducing this pool to 38,128 representatives via MMseqs2 `easy-cluster` to remove near-duplicates. As with the cellulase pipeline, ESM2-650M embeddings were generated with TRILL's `embed` command and a LightGBM classifier was trained via `classify`.

Held-out performance

On the held-out test set (n = 74,489 sequences, with 36,897 cellulases and 37,592 closely-related non-cellulases), the cellulase classifier achieves ROC AUC of 0.989 and cellulase-class average precision of 0.987, with overall accuracy 0.954, balanced accuracy 0.954, cellulase-class F1 0.954, and Matthews correlation coefficient 0.909 at a probability threshold of 0.50. Table 3.1 reports the per-class breakdown, and Figure 3.4 shows the corresponding confusion matrix, ROC, precision-recall, and score-distribution panels.

Table 3.1: Per-class held-out classification report for the LightGBM cellulase classifier at probability threshold 0.50. Class 0 includes related non-cellulase pool (EC 3.-.-.- / 3.2.-.- / 3.2.1.-), and class 1 is cellulase (EC 3.2.1.4).

Class	Precision	Recall	F1	Support
0 (non-cellulase)	0.9520	0.9582	0.9551	37,592
1 (cellulase)	0.9571	0.9508	0.9540	36,897
Macro avg	0.9546	0.9545	0.9545	74,489
Weighted avg	0.9547	0.9545	0.9545	74,489

The two true-class score distributions are well separated, with the bulk of cellulases scoring near $P(\text{Cellulase}) = 1$ and the bulk of non-cellulases scoring near $P(\text{Cellulase}) = 0$. The principal failure mode is a long-tailed overlap near the decision threshold that is difficult to eliminate without sacrificing one of the two recall rates. The fact that the classifier reaches MCC > 0.9 despite being trained on fewer than 3,000 sequences, and validated against the most challenging conceivable

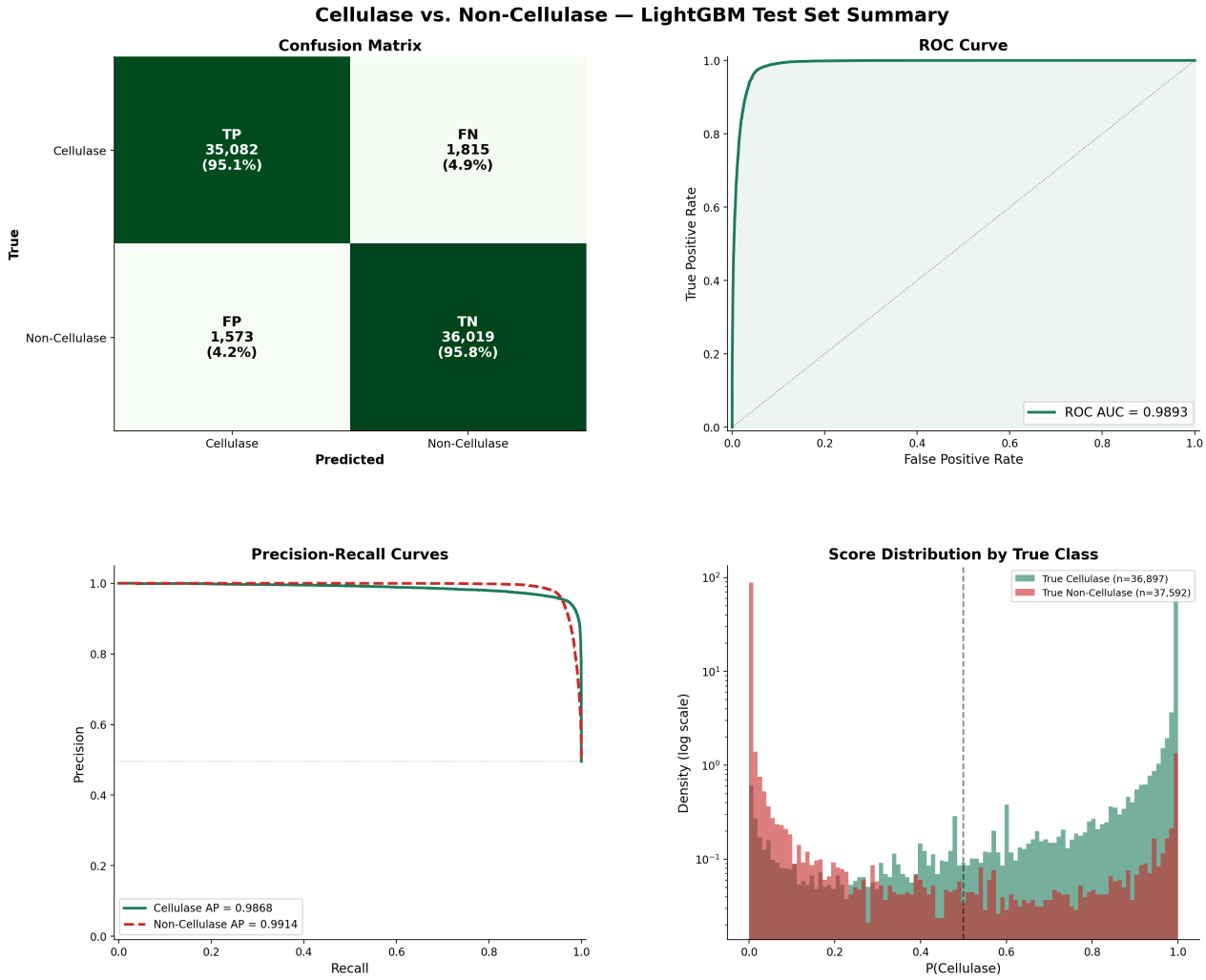


Figure 3.4: LightGBM cellulase-vs-non-cellulase classifier on the held-out test set. **Top**. Confusion matrix and ROC curve. **Bottom**. Precision-recall curves for both classes and score distribution by true label, plotted on a logarithmic density scale. Test set n = 74,489 sequences.

negative set, namely closely related glycosyl hydrolases, is consistent with the broader VP1 finding that ESM2 embeddings concentrate function-specific signal into a tractable representation.

The antimicrobial classifier achieves ROC AUC of 0.9904 and PR AUC of 0.9987 on a merged held-out test set of 18,633 sequences (2,289 antimicrobials and 16,344 non-antimicrobials), with overall accuracy 0.9455, balanced accuracy 0.9543, and Matthews correlation coefficient 0.7960 (Figure 3.5). Table 3.2 reports the per-class breakdown. The classifier correctly identifies 96.6% of held-out antimicrobials (2,211 of 2,289) and 94.3% of held-out non-antimicrobials (15,407 of 16,344).

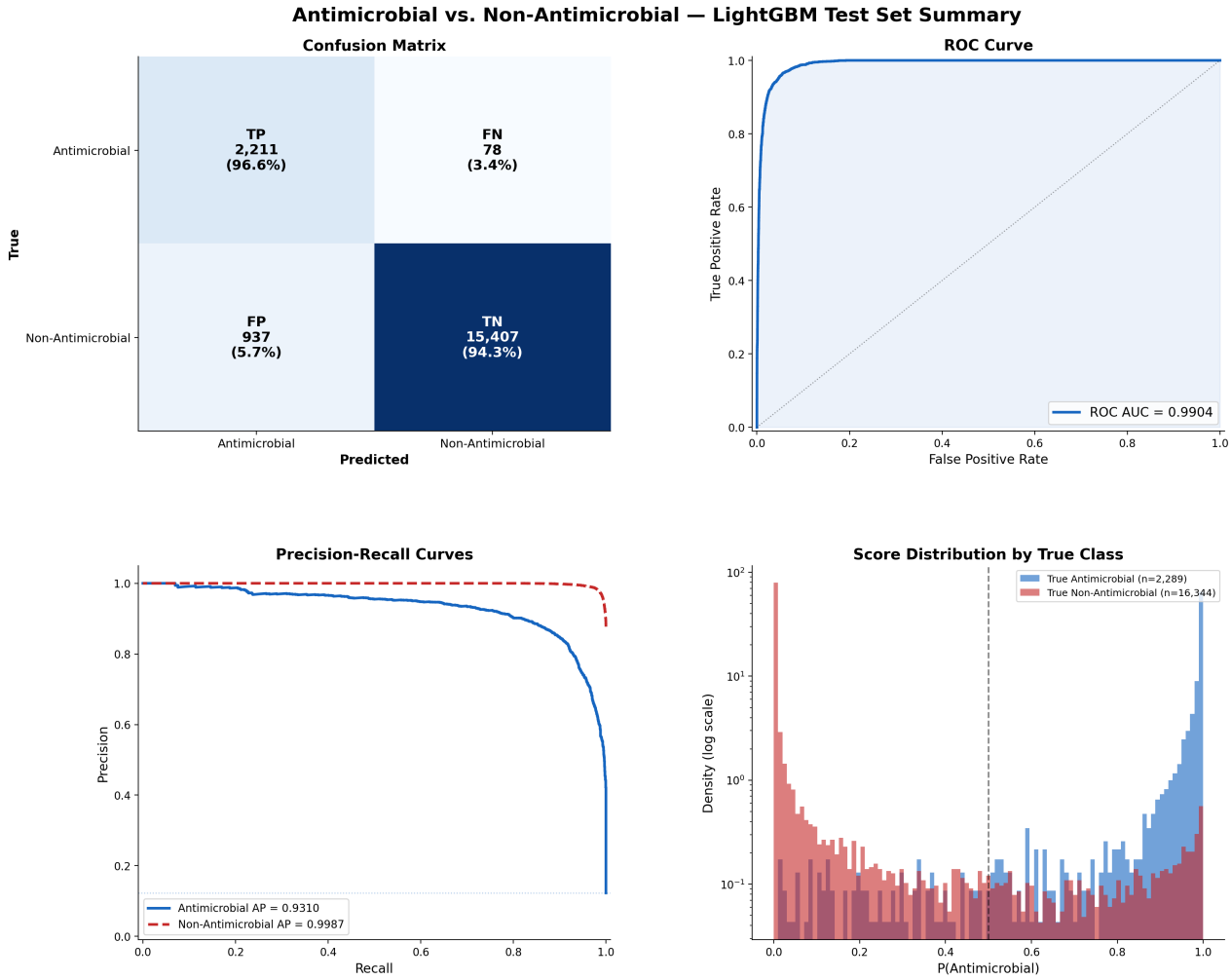


Figure 3.5: LightGBM antimicrobial-vs-non-antimicrobial classifier on the held-out test set. Panels as in Figure 3.4. The non-antimicrobial class dominates the test set (n = 16,344 vs. n = 2,289 antimicrobials), reflecting the relative rarity of validated antimicrobial peptides, and the precision-recall curves report this asymmetry honestly.

Table 3.2: Per-class held-out classification report for the LightGBM antimicrobial classifier at probability threshold 0.50. Note that the non-antimicrobial class is roughly seven times larger than the antimicrobial class in the held-out evaluation, reflecting the relative rarity of validated antimicrobials in the underlying databases.

Class	Precision	Recall	F1	Support
0 (non-antimicrobial)	0.9950	0.9427	0.9681	16,344
1 (antimicrobial)	0.7024	0.9659	0.8133	2,289
Macro avg	0.8487	0.9543	0.8907	18,633
Weighted avg	0.9590	0.9455	0.9491	18,633

The class imbalance is operationally important. Antimicrobials are recovered at high recall (96.6%), but the per-class precision is only 0.70 because the much larger non-antimicrobial pool contributes 937 false positives at the default threshold. The 0.7960 MCC, substantially lower than the cellulase classifier's 0.9032, is a more honest single-number summary than accuracy or F1 alone when classes are this skewed. For downstream use, the practical lesson is that a probability threshold tuned upward will purchase higher antimicrobial precision at modest cost to recall, and the precision-recall curve in Figure 3.5 is the appropriate object for choosing that threshold.

Cellulase generation with ProtGPT2

A classifier good enough to filter natural sequences could also good enough to score the output of a generative model. To test whether TRILL's classify–finetune–generate–classify loop can produce novel candidate cellulases, I clustered the held-out cellulase test set with MMseqs2 `easy-cluster` to 2,613 non-redundant representatives, then used the TRILL `finetune` command to adapt ProtGPT2 (Ferruz, Schmidt, and Höcker, 2022) on those representatives for 10 epochs (at least 1,500 sequences generated per epoch), yielding 10,003 unique fine-tuned ProtGPT2 sequences. I chose the test-set to prevent data-leakage occuring between training the classifier and fine-tuning the generator on the same sequences. As a generative-baseline control I also produced 36,286 sequences from the base ProtGPT2.

Random-sequence controls A classifier that has been trained to discriminate cellulases from near-neighbour glycosyl hydrolases will not necessarily respond predictably to inputs that lie outside the training distribution entirely. To probe this, I produced a structured family of synthetic random-sequence controls. For each of six Poisson length parameters $\lambda \in \{508, 1000, 1500, 2000, 2500, 3000\}$, with $\lambda = 508$ chosen to approximate the mean length of the clustered cellulase training set and the larger values covering long-protein regimes, I generated 9,000 sequences whose lengths were drawn from a Poisson(λ) distribution, with all positions after the first sampled independently and uniformly from the 20 canonical amino acids. The three families differ only in the distribution from which the N-terminal residue is drawn. The *uniform* family samples the first residue uniformly over the 20 amino acids, providing a fully naive null. The *clustered-bias* family samples the first residue from the empirical N-terminal frequency of the 2,613 clustered cellulase representatives, in which 99.35% of sequences begin with methionine and the remainder begin with one

of S, H, V, A, R, T, G, K, P, or Q. The *UniProt-bias* family samples the first residue from the empirical N-terminal frequency of the full 36,235-sequence cellulase set described above, in which roughly 99.7% of sequences begin with methionine. In total, 162,000 synthetic control sequences were assayed.

Scoring The fine-tuned ProtGPT2 sequences, base ProtGPT2 sequences, and all eighteen random-control sets were scored by the cellulase LightGBM classifier described above. Figure 3.6 reports the resulting predicted-positive rates.

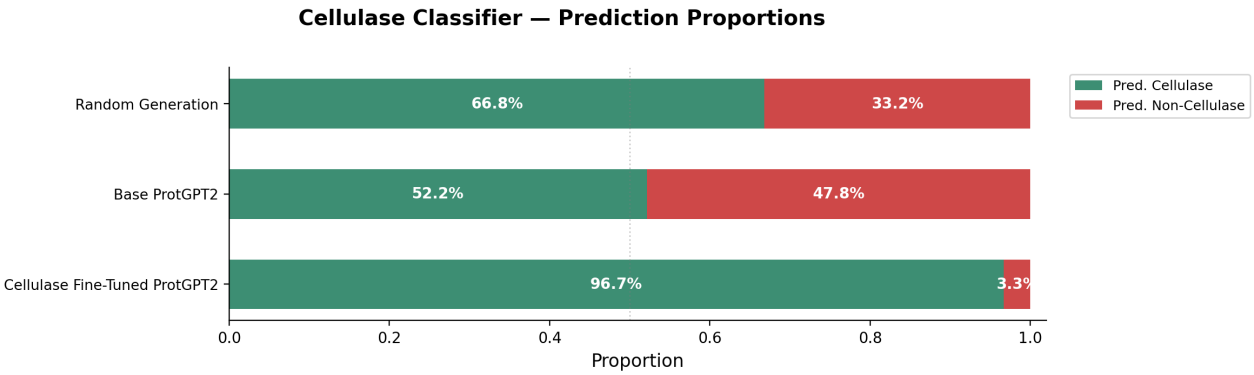


Figure 3.6: Predicted-cellulase rates assigned by the LightGBM classifier to sequences from three generation regimes. **Random Generation** aggregates the 18 synthetic null controls (six Poisson lengths × three N-terminal-residue distributions). **Base ProtGPT2** is the un-fine-tuned baseline. **Cellulase Fine-Tuned ProtGPT2** is the model fine-tuned on 2,613 clustered cellulase representatives for 10 epochs.

Fine-tuned ProtGPT2 produces sequences that the classifier predicts to be cellulases 96.7% of the time, compared with 52.2% for the base ProtGPT2 and roughly 66.8% for the aggregate random controls. Two findings are worth flagging. First, fine-tuning produces a dramatic increase in classifier-validated hit rate (52.2% to 96.7%), consistent with the qualitative pattern observed for anti-CRISPR proteins and the published CPP result (Martinez, Murray, and Thomson, 2023), which are both later discussed in this chapter. Second, the surprisingly high apparent success rate of random controls, approximately two thirds of uniformly random sequences are classified as cellulases, might be a feature of training a classifier against a hard, function-specific negative set rather than against arbitrary proteins. The classifier has been forced to learn "not a glycosyl hydrolase" rather than "not a protein at all", so its predictions on inputs that look unlike any real protein are unreliable. This is not a defect of the workflow but a reminder that classifier-validated generation rates should always be interpreted alongside out-of-distribution null controls. The 30-percentage-point margin between the fine-tuned ProtGPT2 and the random controls

is the quantity that should be trusted as evidence of genuine cellulase-like sequence features in the fine-tuned outputs.

Antimicrobial generation with ProtGPT2

The same classify–finetune–generate–classify loop applied to cellulases above was applied to antimicrobial peptides, using the antimicrobial-positive test set as the fine-tuning corpus. TRILL's `finetune` command was used to further train ProtGPT2 on the antimicrobial test sequences, and the resulting model was used to generate a large pool of candidate antimicrobials with `lang_gen`. As with the cellulase pipeline, a base ProtGPT2 baseline and a structured family of synthetic random-sequence controls were produced for comparison.

Random-sequence controls The control design followed the same three-tier N-terminal-residue scheme used for cellulases, but with two adaptations tailored to the antimicrobial setting. First, the length distribution was extended downward to span the typical antimicrobial peptide regime. For each of nine Poisson length parameters $\lambda \in \{39, 52, 100, 500, 1000, 1500, 2000, 2500, 3000\}$, 9,000 sequences were generated per family, with $\lambda \in \{39, 52, 100\}$ covering short AMPs and the larger values extending into longer-protein regimes to cover the upper tail of the training data. As before, sequence lengths were drawn from a Poisson(λ) distribution, with all positions after the first sampled independently and uniformly from the 20 canonical amino acids. For the *uniform* family the first residue was sampled uniformly over the 20 amino acids, providing a fully naive null. For the *clustered-bias* family the first residue was sampled from the empirical N-terminal frequency of the 2,289 representative sequences obtained by clustering the antimicrobial training set with MMseqs2 `easy-cluster`. Unlike the methionine-dominated distribution observed for cellulases (where roughly 99% of sequences began with methionine), this antimicrobial distribution is substantially more diverse, with glycine the most common starting residue (32.1%), followed by methionine (17.3%), phenylalanine (17.1%), and arginine (9.7%). Together these four residues account for 76% of N-termini. For the *full-set bias* family the first residue was sampled from the empirical N-terminal frequency of the full 11,445-sequence antimicrobial reference set, in which methionine (21.9%) and glycine (19.2%) were the two dominant residues, followed by lysine, alanine, arginine, and phenylalanine at 7–8% each. In total, we generated 243,000 synthetic control sequences, about 50% more controls than the cellulase null pool, owing to the extra length parameters needed to cover the AMP

regime.

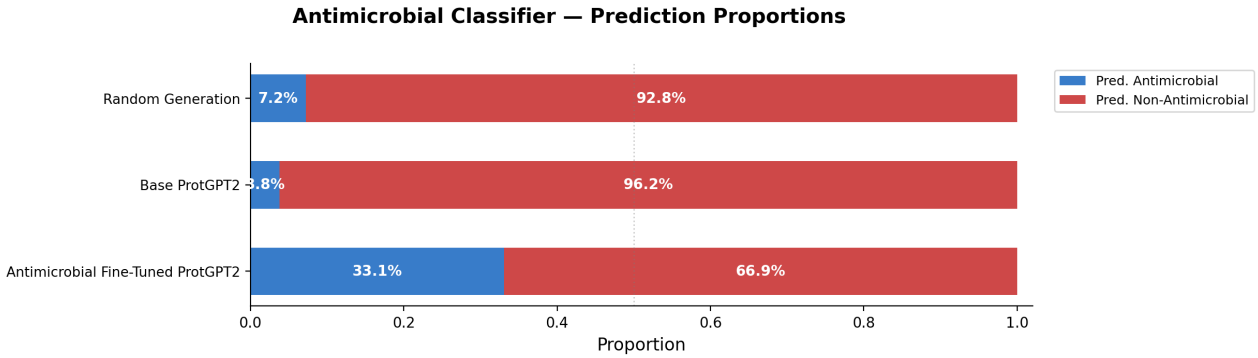


Figure 3.7: Predicted-antimicrobial rates assigned by the LightGBM classifier to sequences from three generation regimes. **Random Generation** aggregates the synthetic null controls. **Base ProtGPT2** is the un-fine-tuned baseline. **Antimicrobial Fine-Tuned ProtGPT2** is the model adapted on the antimicrobial training set.

Figure 3.7 reports the resulting predicted-antimicrobial rates across the three generation regimes. Fine-tuning ProtGPT2 on the antimicrobial training set raises the classifier-validated antimicrobial rate from 3.8% (base ProtGPT2) to 33.1% (fine-tuned). The aggregate random controls sit at 7.2%, providing the relevant null floor. The 26-percentage-point margin of the fine-tuned model over the random baseline is the figure that should be trusted as evidence of genuine antimicrobial-like sequence features in the fine-tuned outputs.

The contrast with the cellulase generation result (96.7% fine-tuned vs. 52.2% base vs. 66.8% random) is informative. For cellulases, even the random and base-ProtGPT2 baselines score very high, potentially a consequence stemming from training the cellulase classifier against a function-specific near-neighbour negative set, so that out-of-distribution random sequences end up classified as cellulases by default. For antimicrobials, the negative set was drawn from the broad UniProt SwissProt pool of non-AMP proteins, which is itself a much larger and more heterogeneous distribution. The classifier therefore might learn "not a UniProt protein in the non-AMP majority" rather than "not a glycosyl hydrolase", and its predictions on random sequences fall close to zero. The same workflow can therefore look very different across functional classes purely because of how the negative set is constructed, which is a useful reminder that the absolute predicted-positive rate is hard to interpret without the corresponding null controls. The signal that travels across both case studies is the *margin* between fine-tuned and baseline rates, at 30 percentage points for cellulases, 26 percentage points for antimicrobials, and consistent in sign with the qualitative

pattern from CPP and ACr generation in Section 3.5.

Another observation worth recording is the lower ceiling for antimicrobial generation (33%) compared with cellulase generation (97%). Fine-tuning ProtGPT2 on antimicrobials therefore moves the generated distribution in the right direction but does not concentrate it as tightly as fine-tuning on cellulases does, and the achievable hit rate is correspondingly more modest. This is consistent with the broader theme that emerged from the legacy CPP and ACr workflows in Section 3.5, namely that family-based generation is most effective when the target class is ancestrally or functionally related in PLM latent space.

Generation hit rate across fine-tuning epochs

The numbers reported above came from the single fine-tuned ProtGPT2 checkpoint that produced the largest pool of generated sequences for each family. The TRILL `finetune` command supports saving a checkpoint after every epoch (`-save_on_epoch`), and exercising this option lets us trace how the predicted-positive rate evolves over the course of fine-tuning rather than collapsing it to a single number. Figure 3.8 shows the result for both families. At each epoch from 0 (the un-fine-tuned base model) through 10, sequences were generated and scored by the corresponding LightGBM classifier, and the predicted-positive rate is plotted.

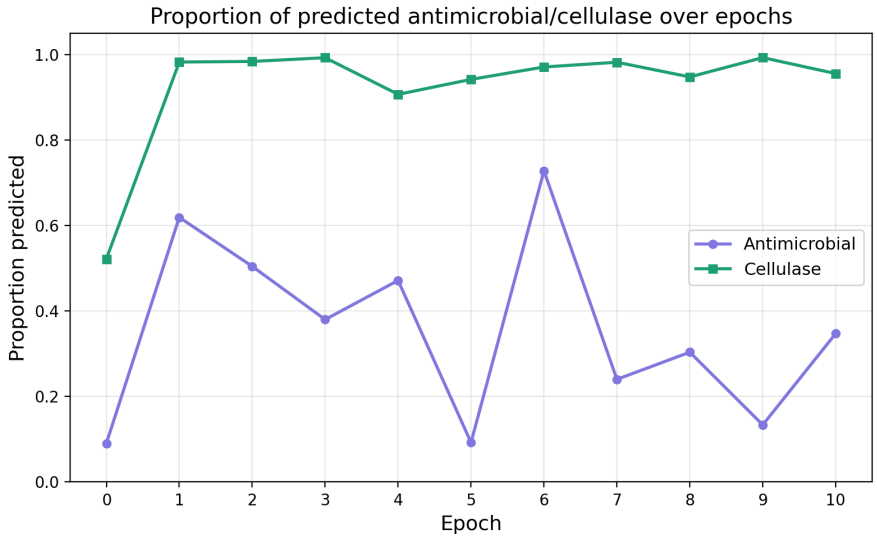


Figure 3.8: Proportion of ProtGPT2-generated sequences classified as positive by the matching LightGBM classifier, as a function of fine-tuning epoch. Epoch 0 corresponds to the un-fine-tuned base ProtGPT2.

In both functional classes, the largest or second largest gain occurs after a *single*

epoch of fine-tuning, where cellulase rises from ~52% to ~98%, and antimicrobial likewise jumps sharply from the un-fine-tuned baseline to a single-epoch peak around ~62%. Additional epochs do not monotonically improve the hit rate. For cellulases, they keep the classifier-validated rate fluctuating in a narrow high band (~91–99%), while for antimicrobials they produce dramatic swings, with the per-epoch evaluation reaching ~73% at epoch 6 and dropping back to ~13% at epoch 9. Second, the magnitude of this epoch-to-epoch variability differs sharply between the two families. Cellulase generation is stable across the entire schedule, whereas antimicrobial generation is essentially noise-dominated after epoch 1.

The implication for practice is the same one I drew from the legacy CPP and anti-CRISPR workflows in Section 3.5. Saving a checkpoint at every epoch when fine-tuning ProtGPT2 is not optional. The contrast between the cellulase (stable) and antimicrobial (volatile) curves provides a useful early diagnostic for users. A stable plateau across epochs suggests that the training family is structurally well-defined in PLM latent space and that the generated distribution has converged onto a tight functional region. A volatile curve signals that the family is heterogeneous enough that the generative model is overfitting to specific subsets of the training set on each pass, and that any single epoch's checkpoint should be treated as just one sample from a noisy distribution.

Detecting cellulases and antimicrobials in the NCBI non-redundant database
Both classifiers were applied to a unified scan of NCBI's non-redundant (NR)database under 5,000 amino acids, totalling 202,654,366 protein records spanning all domains of life. Of these, 62,255,157 (30.7%) carry a domain-of-unknown-function (DUF) or hypothetical-protein annotation in NCBI, the unannotated long tail that conventional keyword and BLAST searches (Altschul et al., 1990) cannot resolve, and the population on which an embedding-based classifier is most likely to recover function that other methods miss. ESM2-650M embeddings for the full pool were computed in a single TRILL `embed` run distributed over multiple GPUs using PyTorch Lightning's distributed data parallel strategy. The same unified embedding set was reused for the threat-classifier scan in Section 3.8, so the three NR-scale property scans in this chapter share a single pool and are directly comparable.

Taxonomic breadth and depth of the NR Figure 3.9 characterises the taxonomic structure of the NR pool used for the cross-classifier statistical analyses in this section and in Section 3.8. The analysis pool used for the per-classifier scans is the unified

202.6M-protein NR. The figure is computed on the slightly smaller 182,354,366-sequence four-way intersection in which every record has predictions from all three TRILL trained-on-all classifiers and an ECPICK EC-number assignment (the universe used by the per-EC enrichment and Jaccard analyses elsewhere in this chapter). Panel A shows that the kingdom-level composition is dominated by bacteria (Pseudomonadati ~51M and Bacillati ~32M, together ~45% of the pool), followed by animals (Metazoa ~13M), fungi (~10M), and plants (~6M), with archaea and viruses each contributing < 2%, and roughly 35.8% of records (65.2M) lack a kingdom-level assignment. Panel B reports the number of distinct assigned taxa at each rank, namely 22 kingdoms, 273 phyla, 495 classes, 1,728 orders, and 8,485 families. Panel C quantifies how concentrated the taxonomic diversity is. At every rank, fewer than 20 taxa cover more than half of the assigned NR records, while the long tail extends across thousands of families.

For each of cellulase and antimicrobial activity, two versions of the classifier were applied to this pool. The first is the held-out-validated classifier reported in Section 3.4 above, trained on the training partition only and characterized on the disjoint held-out test set. The second is a production-retrained classifier, fit on the union of the training and test sets to maximize generalizability at deployment time. This version has no held-out evaluation by construction, but it draws on more labeled data and is the version whose predictions should be carried forward into downstream analyses. Reporting both lets the calibration confidence of the held-out-validated classifier be read alongside the production yield of the retrained classifier. The held-out version anchors the precision and recall numbers, while the retrained version is the canonical NR-scale call set.

Cellulase The held-out-validated cellulase classifier flagged 29,454,478 NR sequences as predicted cellulases (14.53% of the pool), of which 16,952,240 (57.6% of predicted positives) carry a hypothetical or DUF annotation. The production-retrained classifier flagged 13,453,908 sequences (6.64%), of which 7,999,166 (59.5%) are hypothetical. The factor-of-two reduction in predicted-positive rate when the training corpus is expanded is consistent with the larger labeled pool tightening the classifier's decision boundary around the cellulase manifold. The hypothetical fraction of the surviving predictions *increases* slightly from 57.6% to 59.5%, indicating that the tighter boundary preserves the hypothetical-protein hits that motivated the scan in the first place rather than collapsing onto already-annotated sequences.

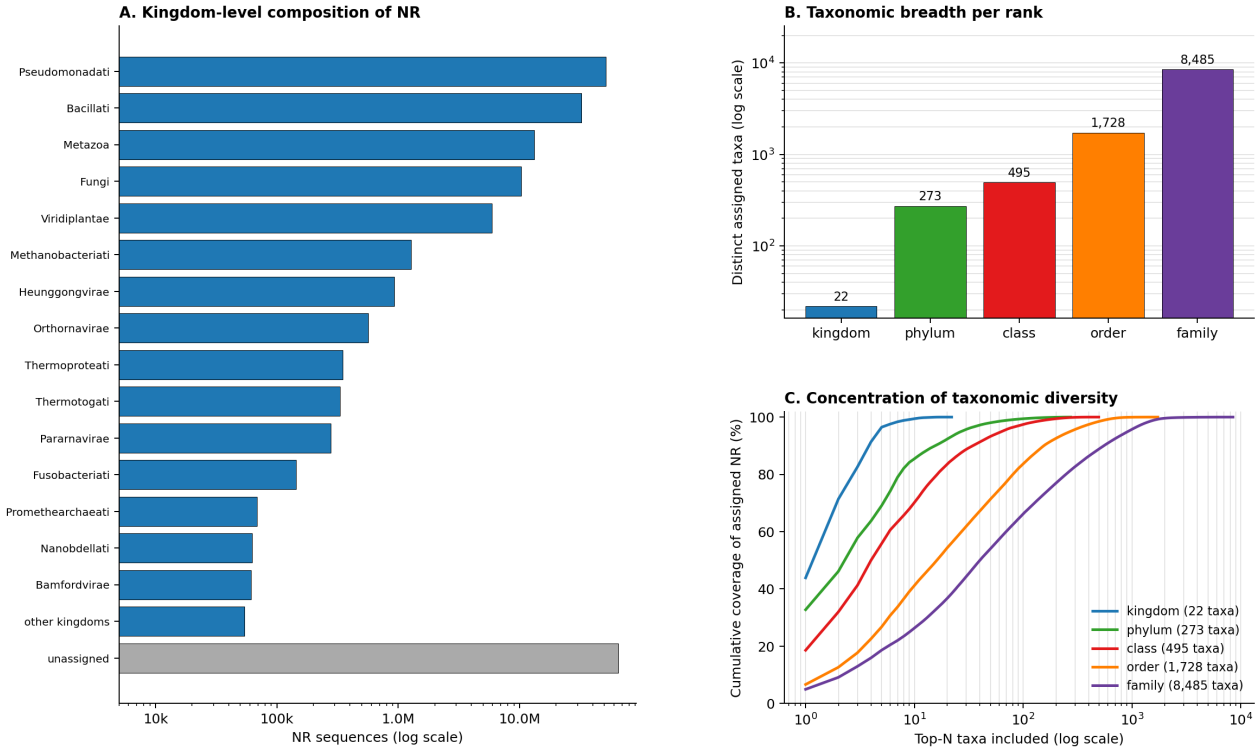


Figure 3.9: Taxonomic breadth and depth of the NR analysis pool. **A**. Kingdom-level composition. NR is dominated by Gram-negative and Gram-positive bacteria, followed by animals, fungi, plants, archaea, and several viral kingdoms, and 35.8% of records (65.2M) lack a kingdom-level assignment. **B**. Distinct assigned taxa per rank on a log scale, with 22 kingdoms, 273 phyla, 495 classes, 1,728 orders, 8,485 families. **C**. Cumulative coverage of assigned-NR records by the top-N taxa at each rank. At every rank a small head of taxa accounts for most records, with a long tail extending across the remaining diversity.

Antimicrobial The held-out-validated antimicrobial classifier flagged 1,991,167 NR sequences (0.98% of the pool), of which 1,132,341 (56.9%) are hypothetical. The production-retrained classifier flagged 877,655 sequences (0.43%), of which 406,603 (46.3%) are hypothetical. As with cellulase, retraining on the full labeled pool more than halves the predicted-positive rate, again consistent with a tightened decision boundary. Unlike cellulase, the hypothetical fraction *drops* from 56.9% to 46.3%.

Discovery yield over the unannotated dark matter Taken together, the production-retrained cellulase and antimicrobial classifiers flag 7,999,166+406,603 = 8,405,769 hypothetical NR proteins as candidate function-bearers across these two properties

alone. Read against the 62,255,157-strong hypothetical-protein background, that figure corresponds to roughly 12.85% of the unannotated NR pool being flagged for cellulase activity and roughly 0.65% for antimicrobial activity by the trained-on-all classifiers. The absolute predicted-positive rates should be interpreted cautiously. The cellulase rate of 6.64% across all of NR might reflect the fact that the classifier was deliberately trained against the closely related EC 3.-.-.-, EC 3.2.-.-, and EC 3.2.1.- neighbourhoods rather than solely against arbitrary proteins, so any glycosyl hydrolase that the classifier cannot cleanly separate from cellulase will appear in this count. But the operational point of the scan is unchanged. A LightGBM classifier trained on a few thousand examples through TRILL's `classify` command can expand the candidate pool for a given functional property.

Joint behavior of the three production classifiers Because the cellulase, antimicrobial, and threat trained-on-all classifiers were applied to the same NR pool, the predicted-positive sets can be intersected directly. Figure 3.10 shows the resulting UpSet plot. The dominant intersection is the trivially expected one, with 185,999,167 proteins (91.8% of the NR pool) flagged by none of the three classifiers. Among the predicted positives, the single-classifier-only intersections account for the bulk of the calls, with 12,080,573 cellulase-only, 2,545,150 threat-only, and 331,272 antimicrobial-only, consistent with the three properties being substantially distinct functional categories. The two-way overlaps are smaller but informative. 1,151,821 proteins are predicted to be both cellulase and threat, 324,869 are predicted to be both antimicrobial and threat, and 82,913 are predicted to be both antimicrobial and cellulase. The three-way intersection, proteins flagged simultaneously as cellulase, antimicrobial, and threat, contains 138,601 sequences, slightly larger than the antimicrobial-with-threat-only intersection of 324,869 would suggest. The asymmetry of these overlaps is interpretable. The cellulase-with-threat overlap (1.15M) is by far the largest because the cellulase classifier was trained against the closely related EC 3.2.-.- glycosyl hydrolase neighbourhoods rather than against arbitrary proteins, and the resulting decision boundary fires broadly on CAZyme architectures including non-catalytic carbohydrate-binding modules (CBMs), lectin-like domains, and expansin-family proteins that share fold-level features with cellulase enzymes without sharing their substrate specificity. That overlap surface intersects with the threat classifier's pickup on the chymotrypsin-fold S1 serine protease family discussed in Section 3.8, not because cellulases and toxins share compositional features, but because both classifiers' decision boundaries pass through architec-

turally similar regions of PLM latent space. The antimicrobial-with-threat overlap (324,869), by contrast, is relatively depleted because the antimicrobial training set was explicitly stripped of `KW-0800` threats, so the residual overlap reflects sequences whose embedding signatures resemble both classes despite the label-level separation enforced during training. The practical consequence for users running multiple TRILL classifiers in parallel is that overlap inspection is a meaningful diagnostic. A property pair whose predicted-positive sets overlap more than expected from independent base rates is signaling either a genuine functional connection, a shared compositional bias, or a shared decision-boundary surface that should be inspected before either prediction is trusted in isolation.

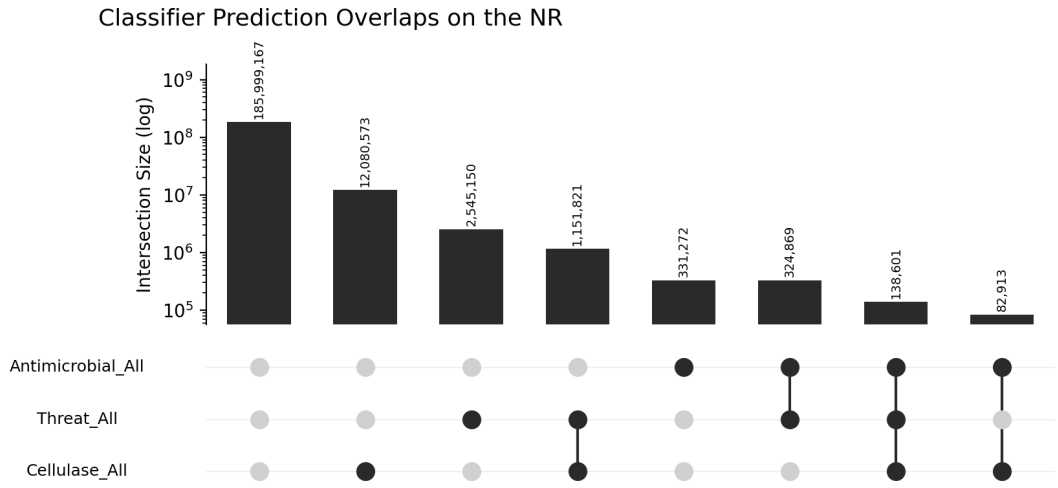


Figure 3.10: UpSet plot of the predicted-positive sets returned by the three trained-on-all classifiers (cellulase, antimicrobial, and threat) on the unified 202,654,366-protein NR pool. Bars give intersection sizes on a $\log_{10}$ scale, with dots beneath identifying which classifier(s) call each intersection a positive.

Orthogonal cross-validation with ECPICK The NR-scale predictions can be cross-checked against ECPICK (Han et al., 2023), a deep-learning EC-number predictor that does not rely on protein language models and therefore ostensibly provides an independent functional reference. Restricting attention to the subset of NR proteins for which ECPICK assigns the EC 3.2.1.4 (cellulase) label, 7,198 proteins in total, the cellulase trained-on-all classifier returns a positive call on 6,950 of them, a recall of 96.55%. The antimicrobial and threat classifiers return positives on only 14 (0.19%) and 7 (0.10%) of the same ECPICK-cellulase pool, respectively, with all three classifiers simultaneously positive on just 2 sequences (0.03%). The three classifiers are therefore well separated on an independently labeled cellulase

reference. The property-matched classifier recovers the bulk of the targets, and the two off-property classifiers fire below 0.2%. Broadening the reference to all ECPICK-predicted EC 3.2.1.∗ glycoside hydrolases (n = 476,123) shows the inverse behavior expected from a selective rather than a family-wide classifier. Cellulase calls 15.88% of the broader family as positive, antimicrobial 10.13%, and threat 18.00%, with all three simultaneously positive on only 0.42% of the subset and all three simultaneously negative on 65.00%. The cellulase classifier therefore does not over-fire on nearby glycoside hydrolases that share its catalytic family but not its substrate, which is the principal failure mode that the EC-neighbourhood-controlled training set in Section 3.4 was designed to suppress.

Where the predictions fall on the tree-of-life The resulting classifications can be further stratified by taxonomy to identify the kingdoms and phyla with disproportionately high predicted-positive rates. Figure 3.11 reports the per-phylum predicted-positive rate for the cellulase and antimicrobial trained-on-all classifiers, restricted to phyla with at least 100,000 NR proteins and showing the top 25 phyla per panel by classifier-specific positive rate. The two distributions are markedly different and biologically interpretable. The cellulase classifier fires most aggressively on the eukaryotic clades that produce or degrade plant cell walls, including Chlorophyta (green algae, ∼22% positive rate), Streptophyta (land plants, ∼21%), Basidiomycota (∼18%) and Chytridiomycota (∼16%, both saprotrophic fungi), and the nematode and oomycete clades whose lifestyles are dominated by plant-cell-wall interactions (∼14–16%). The antimicrobial classifier instead concentrates on viral phyla (Uroviricota at ∼1.8%, Artverviricota at ∼1.0%), consistent with phage lysins and holins being among the most common antimicrobial peptides in NR. The two phylum-by-phylum profiles share little overlap at the top of the ranking, which is the expected behavior for two functionally distinct classifiers and an additional cross-check that the NR-wide signal is not dominated by a generic "small-protein" or compositional bias.

A complementary kingdom-resolved view that adds the threat classifier to the same picture is shown in Figure 3.12, aggregating per-genus positive rates (per 1000 proteins, isoforms removed, genera with at least 100 NR proteins) across the eleven NCBI kingdoms that pass the genus-coverage filter. The cellulase classifier's per-1000-protein rate peaks in Fungi (median ∼140), consistent with the well-characterised lignocellulolytic CAZyme repertoires of saprotrophic ascomycetes such as *Trichoderma reesei* (Martinez et al., 2008), with substantial signal also in Bacillati and

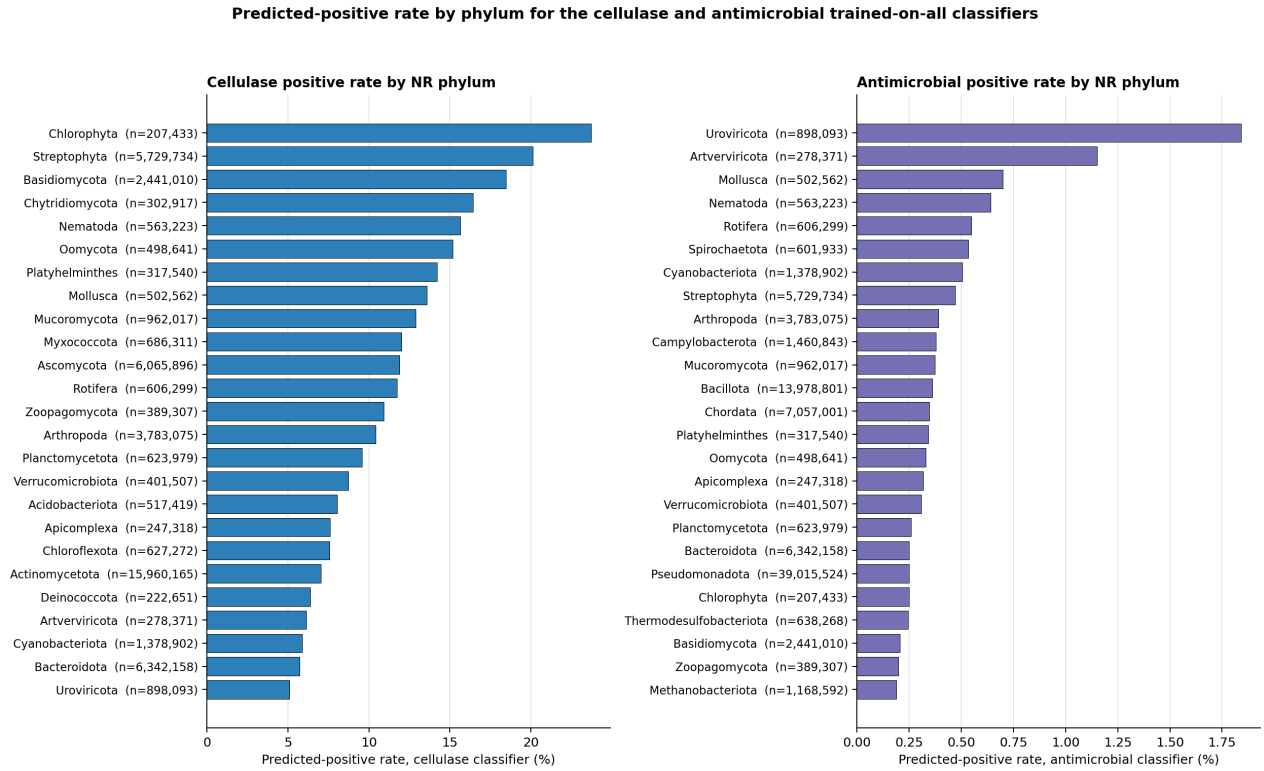


Figure 3.11: Predicted-positive rate by NR phylum for the cellulase (left, blue) and antimicrobial (right, purple) trained-on-all classifiers. Phyla with fewer than 100,000 NR proteins are excluded, and within each panel the top 25 phyla by classifier-specific positive rate are shown. Cellulase positives concentrate in the plant- and fungal-cell-wall-degrading clades, while antimicrobial positives concentrate in viral phyla dominated by phage lysins and holins. The two profiles share little overlap at the top of the ranking.

in Viridiplantae, the latter likely reflecting the endogenous endo-β-1,4-glucanase (GH9) repertoire that plants use for cell-wall biosynthesis and loosening rather than a pathogen-defence response (Glass et al., 2015). The antimicrobial and threat classifiers concentrate in the eukaryotic kingdoms. The antimicrobial signal in Metazoa and Viridiplantae aligns with the well-documented animal defensin/cathelicidin (Ganz, 2003) and plant defensin/thionin/cyclotide (Tam et al., 2015) repertoires, the Metazoa threat signal aligns with the chymotrypsin-fold S1 serine-protease-rich venom proteomes of arthropods and reptiles, and the Viridiplantae threat signal aligns with the ribosome-inactivating-protein and lectin-rich plant defence proteome (Zhu et al., 2018). Archaeal kingdoms (Methanobacteriati, Nanobdellati, Promethearchaeati, Thermoproteati) consistently show the lowest rates across all three classifiers, consistent with the small and structurally distinct archaeal antimicrobial repertoire (halocins, sulfolobicins) being underrepresented in SwissProt-derived training corpora (Besse

et al., 2015).

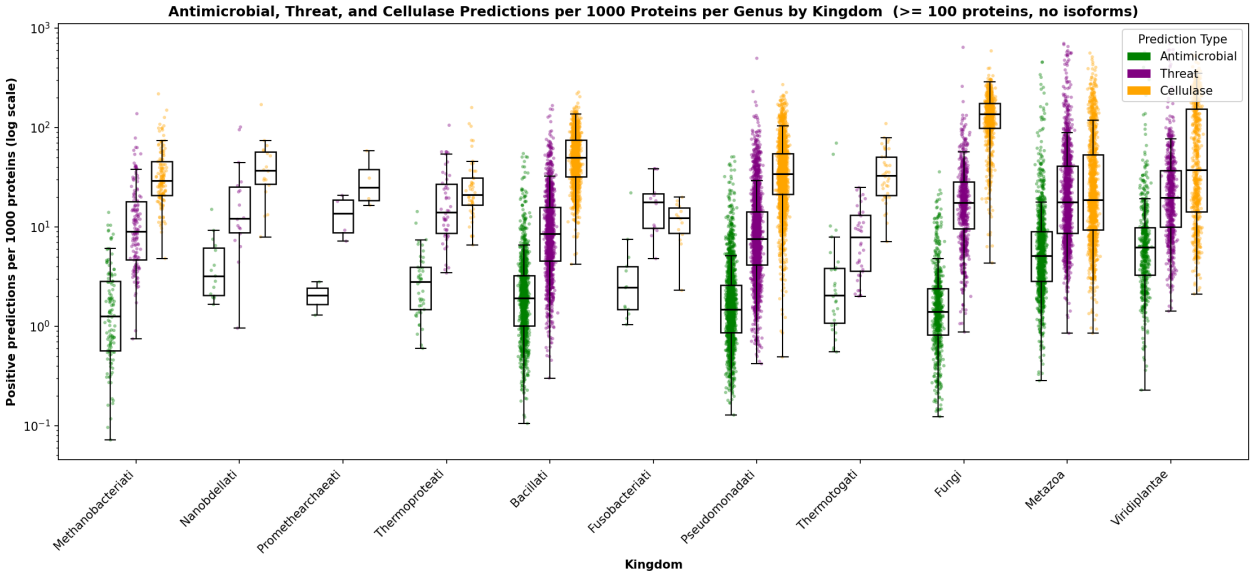


Figure 3.12: Per-genus predicted-positive rates per 1000 proteins for the antimicrobial (green), threat (purple), and cellulase (orange) trained-on-all classifiers, aggregated by NCBI kingdom and shown on a $\log_{10}$ scale. Each point is one genus with at least 100 NR proteins (isoforms removed), and box plots overlay the per-genus distribution within each kingdom. Cellulase rates peak in Fungi and are also high in Bacillati and Viridiplantae. Antimicrobial and threat rates concentrate in Metazoa and Viridiplantae. Archaeal kingdoms (left four) show the lowest rates across all three classifiers.

3.5 Family-based protein generation of cell-penetrating peptides and anti-CRISPR proteins

Previous findings

This entire pipeline involves no Python beyond the user's choice of input file. It is the workflow used to demonstrate TRILL's composability in (Martinez, Murray, and Thomson, 2023), and it has been applied to two protein families whose members are routinely missed by sequence-alignment methods, namely cell-penetrating peptides (CPPs) and anti-CRISPR proteins (ACrs).

Cell-penetrating peptides

Cell permeability is a function-defined rather than ancestry-defined property. Many cell-penetrating peptides arose independently and share no recognizable sequence homology with one another despite converging on the ability to translocate intact cell membranes (Yadahalli and Verma, 2020). This is precisely the regime where

alignment-based methods fail and embedding-based methods are expected to help.

Using Dataset E from Yadahalli and Verma (Yadahalli and Verma, 2020), I trained an XGBoost classifier on ESM2-150M embeddings of 75% of the labeled CPPs against an equal number of negative peptides, and evaluated it on the held-out 25%. The classifier achieved an F1 score of 0.876 (Figure 3.13). I then fine-tuned ProtGPT2 on the 955 known CPPs for 10 epochs with a learning rate of 10^{-5} and used the resulting model to generate 1,000 candidate sequences seeded with each of the nine most-frequent CPP-starting residues, for a total of 9,000 generated proteins.

Figure 3.13: Performance of XGBoost and Isolation Forest classifiers trained on ESM2-150M embeddings of cell-penetrating peptides from (Yadahalli and Verma, 2020). **Top**. PCA projection of CPP embeddings colored by ground-truth label, Isolation Forest predictions, and XGBoost predictions. **Bottom**. Confusion matrices for the two classifiers. Reproduced from (Martinez, Murray, and Thomson, 2023).

The classifier predicted that 22% of sequences generated by the base ProtGPT2 model were CPPs, compared to 57% of sequences generated by the CPP-fine-tuned model. These rates should be read with one methodological caveat. The legacy CPP and anti-CRISPR classifiers and generators in this section were trained and evaluated on random rather than homology-aware train/test splits, so the predicted-CPP rate on fine-tuned ProtGPT2 sequences may partly reflect classifier-to-generator data leakage, the classifier scoring its own training neighbourhood positively when the generator emits sequences drawn from that same neighbourhood, rather than genuine novel-CPP generation. The remedy in newer TRILL workflows is the MMseqs2-clustered split

used in the cellulase, antimicrobial, threat, and BioSentinel classifiers later in this chapter (Sections 3.4, 3.8, 3.9).

To verify that the 22% base-model rate was not an artifact of the classifier responding to superficial features such as starting residue composition or peptide length, I generated two control groups, namely 9,000 uniform-random peptides with lengths drawn from a Poisson distribution ($\lambda = 55$, matching the CPP set), and 9,000 peptides with the same length distribution but whose starting residue was sampled from the empirical CPP starting-residue distribution. Both control groups produced predicted-CPP rates well below the 22% base-ProtGPT2 rate, with the starting-residue-biased control scoring closer to the base-model rate than the uniform-random control. The bias control therefore confirmed that ProtGPT2's tendency to begin sequences with positively charged residues, arginine and lysine, also the dominant starting-residue motif in the CPP training set, accounts for a substantial portion of the base-model success rate, while the uniform-random control demonstrated that absent that starting-residue bias the classifier does not return predicted-CPP calls at meaningful rates on length-matched random sequences.

Anti-CRISPR proteins

Anti-CRISPR (ACr) proteins inhibit CRISPR-Cas systems and are of interest both for understanding phage-bacteria conflict and for engineering controllable CRISPR-based tools. Identifying them in sequence data is hard for the same reasons as CPPs. They have arisen multiple times in evolution and share little global sequence similarity even within a single inhibition target (Wang et al., 2020).

Using the validated ACr/non-ACr partition from PaCRISPR (Wang et al., 2020), I trained an XGBoost classifier on ESM2-150M embeddings that reached an F1 score of 0.886 on the held-out 25%. I then fine-tuned ProtGPT2 separately on three Anti-CRISPR families (I-D, I-F, and II-C) for 10 epochs at learning rate 10^{-5}, and generated 1,000 candidate sequences from each fine-tuned model plus the base model.

The base ProtGPT2 model produced predicted ACrs at a rate of 2.1%, while the three fine-tuned variants reached 23.5% (I-D), 28.7% (I-F), and 15.7% (II-C). Figure 3.14 summarizes the success rates for both CPP and ACr generation across base, fine-tuned, and random-protein controls. The same homology-aware-split caveat noted above for CPP applies here. The PaCRISPR partition is a random rather than MMseqs2-clustered split, so part of the fine-tuned models' gain may reflect

classifier-to-generator overlap rather than novel-ACr generation. The contrast with the CPP results is informative.

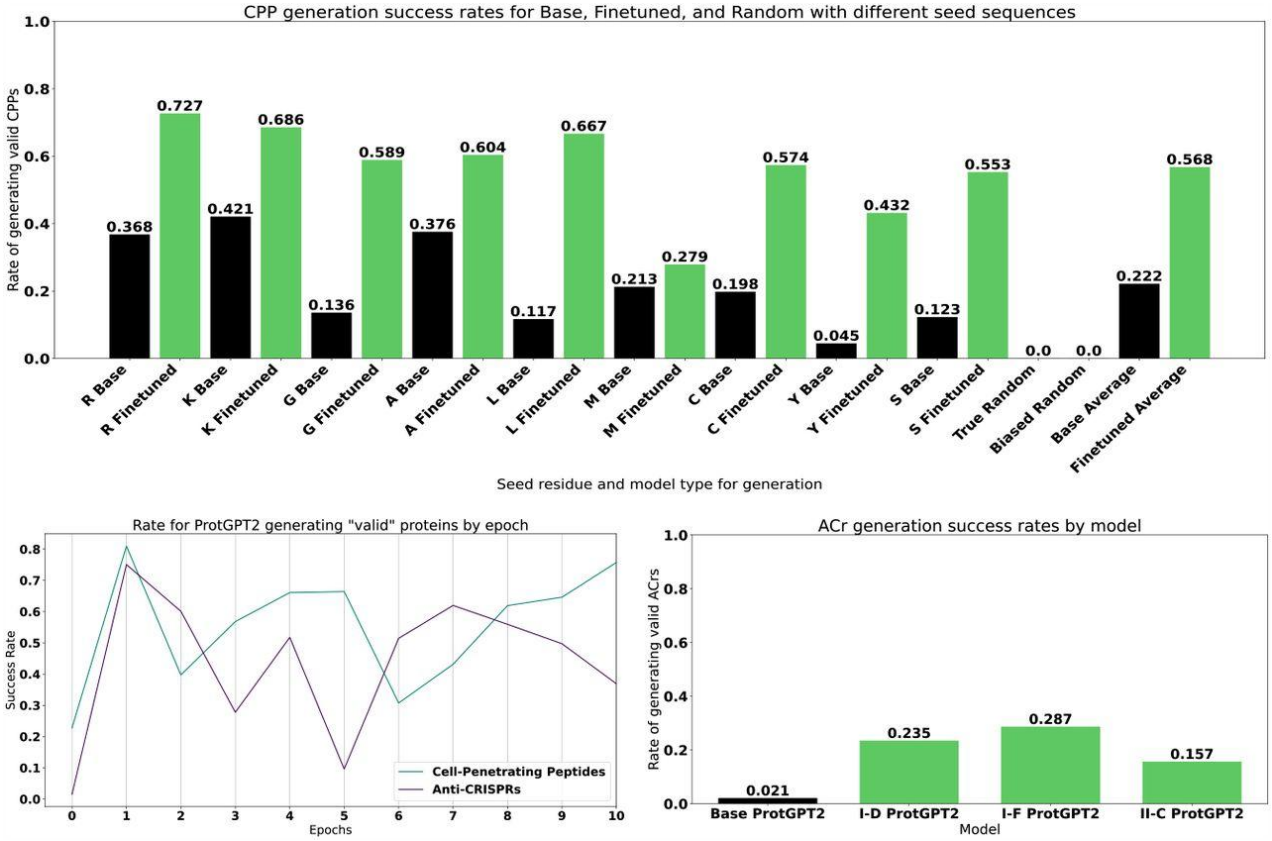


Figure 3.14: Predicted-positive rates for protein generation experiments. **Top**. Predicted-CPP rates for sequences generated either from a random distribution or from ProtGPT2. Black bars are the base ProtGPT2 model, green bars the CPP-fine-tuned ProtGPT2. **Bottom left**. Predicted-positive rate as a function of fine-tuning epoch, for both CPP and ACr-family-fine-tuned models. **Bottom right**. Predicted-ACr rates by ACr family (I-D, I-F, II-C) compared to the base ProtGPT2. Reproduced from (Martinez, Murray, and Thomson, 2023).

Sensitivity to fine-tuning length

For both functional classes, I tracked the predicted hit rate as a function of the number of fine-tuning epochs. After each epoch, 5,000 sequences were generated and scored. Both models reached their peak hit rate after a single epoch, 95% for ACrs and 90% for CPPs at the per-seed best generation, and then drifted downward over subsequent epochs by approximately 40%. The drift is consistent with the model gradually memorizing training-set features at the expense of the generalization that classifier predictions reward. The practical implication is that fine-tuning should not be treated as a monolithic step. Saving and evaluating intermediate checkpoints (the

`-save_on_epoch` flag in TRILL's `finetune` command) is essential, and the best checkpoint for downstream generation is often not the last one.

Discussion

These results, originally published in (Martinez, Murray, and Thomson, 2023), establish a qualitative lesson that hold up across the more recent applications described in the rest of this chapter. Modest fine-tuning of an off-the-shelf ProtGPT2 model achieves the bulk of the available gain. Furthermore, classifier predictions on generated sequences should always be interpreted alongside null controls (random sequences, base-model sequences, length- and composition-matched sequences) because protein language models inherit substantial compositional biases from their training data that can masquerade as functional success.

3.6 A comparison of protein language models for classification

The three classification tasks introduced above, VP1 detection, CPP prediction, and anti-CRISPR prediction, share a common structure of training a TRILL classifier on PLM embeddings and evaluating on a held-out test set. They differ in the nature of the positive class. VP1s are an ancestrally coherent family, CPPs are a function-defined class with extensive convergent evolution, and ACrs sit between the two extremes, with within-family ancestral signal but no signal across families.

This shared structure invites a direct comparison of the embedding models that TRILL exposes, asking which produces the most classification-relevant representation for each task. Table 3.3 reports F1 scores for XGBoost and Isolation Forest classifiers trained on embeddings from ten different PLMs spanning model sizes from 8M to 15B parameters, architectural families, and a six-class multi-class extension that combines the three positive sets with their respective negative pools.

The principal observations are unchanged from the previous analysis. First, returns on scale diminish quickly. ESM2-150M is within 1–2 F1 points of ESM2-15B on every task, and the smallest tested model (ESM2-8M, 320-dimensional) yields the single best ACr-XGBoost score in the entire table. Researchers who care about throughput more than the last few F1 points can confidently default to smaller models for classification work.

Second, the XGBoost-vs-Isolation-Forest gap depends on the class definition. For VP1 (ancestral homology), Isolation Forest closes most of the gap, with several models exceeding F1 of 0.95 in the one-class setting. For CPP (convergent function),

Table 3.3: F1 scores for XGBoost (XG) and Isolation Forest (iF) classifiers trained on embeddings from ten protein language models, evaluated on three binary tasks (CPP, ACr, VP1) and one six-class multi-class classification. The best score in each column is highlighted. Reproduced from (Martinez, Murray, and Thomson, 2023).

PLM	Params	Dim	CPP XG	CPP iF	ACr XG	ACr iF	VP1 XG	VP1 iF	Multi μF1
ESM2-8M	8M	320	0.854	0.546	**0.916**	0.733	0.978	0.833	0.830
ESM2-35M	35M	480	0.853	0.563	0.895	0.714	0.980	0.862	0.835
ESM2-150M	150M	640	0.876	0.553	0.886	0.745	0.984	0.906	0.842
Ankh	450M	768	0.839	**0.613**	0.888	0.751	0.993	0.956	0.826
ESM2-650M	650M	1,280	0.870	0.582	0.904	0.712	0.996	0.970	0.842
Ankh-Large	1.15B	1,536	0.833	0.601	0.882	0.773	0.986	0.443	0.813
ESM2-3B	3B	2,560	0.860	0.599	0.905	0.763	0.999	0.950	**0.850**
ProtT5-XL	3B	1,024	0.880	0.578	0.870	0.764	0.999	**0.972**	0.835
ProstT5	3B	1,024	**0.886**	0.609	0.889	**0.802**	0.998	0.689	0.826
ESM2-15B	15B	5,120	**0.886**	0.611	0.903	0.763	**1.000**	0.963	0.844

Isolation Forest performance plateaus near 0.6, well below XGBoost. This is the same pattern that motivated the choice to evaluate both classifiers separately in the VP1 sweep of Section 3.3.

Third, the multi-class results show that even ten dimensions of useful signal per class is enough to discriminate among six classes at F1 $\approx$ 0.85. The spread between the largest (μF1 0.850 for ESM2-3B) and smallest (μF1 0.830 for ESM2-8M) models is only 0.02. In practice, the cost of running the larger models against routine classification problems is rarely justified.

3.7 Fast foldtuning

Foldtuning and its bottleneck

Foldtuning, introduced by Subramanian and colleagues (Subramanian et al., 2023), is an algorithm that drives protein language models to generate sequences predicted to adopt a target fold or function while maximizing divergence from natural protein sequences. Each round of foldtuning involves four steps, namely generation of candidate sequences from the current PLM state, prediction of their structures with ESMFold (Lin et al., 2023), assignment of fold labels via Foldseek (Van Kempen et al., 2024) structural search against a reference database, and selection of the top candidates by "semantic change" (a measure of how far the generated sequence has departed from natural sequence space) for the next round of model fine-tuning. The original paper applied foldtuning to over 700 structural targets and validated a subset experimentally for expression, stability, and function.

Foldtuning illustrates the kind of multi-model workflow that TRILL was designed to enable. It composes language-model generation, structure prediction, and structural

search through TRILL's standardized file formats without any custom glue code, and is now exposed as a single `workflow` command (Section 2.11 of the preceding chapter). It also illustrates the cost. ESMFold inference is the dominant computational expense, scaling with sequence length and requiring substantial GPU memory. For campaigns involving thousands of candidates per round across many targets, this cost can dominate the entire pipeline.

Fast foldtuning

I implemented "fast foldtuning" within TRILL by replacing the ESMFold step with ProstT5 (Heinzinger et al., 2024), the bilingual T5-style protein language model described in Chapter 2. ProstT5 was trained to translate between amino-acid sequences and Foldseek's 3Di structural alphabet, which means that the per-sequence inference cost of obtaining a 3Di string is reduced to a single inference call to ProstT5 rather than a full structure prediction. Because Foldseek natively accepts 3Di tokens for structural search, the substitution preserves the rest of the foldtuning loop unchanged. Both variants are accessible from the same TRILL `workflow` command, with the fold-prediction backend selected by a single argument.

Benchmark on the three-finger protein fold

The fast-foldtuning substitution was benchmarked on the three-finger protein (3FTx) fold, a small (~60–80 residue), disulfide-rich topology with three β-stranded loops projecting from a hydrophobic core stabilized by four to five conserved disulfide bonds (Kessler et al., 2017). The 3FTx fold has been independently elaborated for venom toxicity in snake α-neurotoxins, immune-system signaling in the human Ly6/uPAR family (SLURP-1/2, Lynx1, Lypd6), and receptor modulation more broadly, making it an attractive target for the kind of fold-constrained sequence exploration that foldtuning was designed for.

Starting from 521 natural 3FTx sequences as training data, I ran five rounds of standard foldtuning and five rounds of fast foldtuning across three independent random seeds for each variant, generating approximately 14,500 sequences per method-seed combination (approximately 43,500 sequences per method in total). Each generated sequence was scored on three axes, namely predicted TM-score to the input 3FTx scaffold (structural fidelity), mean fraction of identical residues to the nearest training-set 3FTx (sequence novelty), and round of origin (to characterize round-over-round dynamics).

Figure 3.15 shows that both variants converge on a regime in which generated

Figure 3.15: Comparison of standard ("Base") and fast foldtuning on the three-finger protein (3FTx) fold over five rounds. **Top two rows**. Per-sequence scatter plots of predicted mean TM-score (structural fidelity) versus mean fraction of identical residues to the input training set (sequence novelty, where lower values are more novel). Density is color-coded on a logarithmic count scale. **Bottom**. Mean TM-score across rounds for foldtuned variants with no detectable sequence homology to the inputs, with each seed plotted independently.

sequences retain high TM-score to the 3FTx scaffold while drifting toward lower fractions of training-set sequence identity, consistent with the published foldtuning behavior. The fast-foldtuning trajectories are broadly similar to the base-foldtuning trajectories across all five rounds, and the round-five subset of generated sequences with no detectable sequence homology to the inputs shows comparable mean TM-scores between the two variants.

Discussion

The fast-foldtuning substitution preserves the most important behavior of standard foldtuning, namely round-over-round drift away from natural sequence space while maintaining structural fidelity, while removing the ESMFold bottleneck. In wall-clock terms on the 3FTx benchmark, a single 5-round foldtuning trajectory took

approximately 207 minutes for the base (ESMFold) variant versus approximately 33 minutes for the fast (ProstT5) variant on identical P100 GPUs, a ~ 6.3× end-to-end speedup at the workflow level. ProstT5's 3Di prediction is an approximation of a structural assignment that ESMFold's 3D output would yield via Foldseek's standard 3Di tokenization, and we should expect rare cases in which the two diverge meaningfully. The empirical observation on 3FTx is that, for the structural and sequence-novelty metrics that matter for downstream foldtuning, the divergence is modest enough that the cheaper variant is a practical default *for this fold*. Whether the substitution generalises is a separate question. A wider-fold sweep parallel to the 700+ targets used in the original foldtuning paper (Subramanian et al., 2023) is the future work that would be needed before promoting fast foldtuning to a general-purpose default.

3.8 End-to-end threat detection and neutralization

Motivation

The applications described so far each exercise one or two pieces of TRILL's command-line surface in isolation, namely `embed` and `classify` for VP1, cellulase, and antimicrobial detection, `finetune`, `lang_gen`, and `classify` for family-based generation, and `workflow` for fast foldtuning. This section presents the longest single composition in the chapter, in which ten chained TRILL commands take a defensive question, "what dangerous proteins are hiding in vast sea of unannotated NCBI proteins, and can we computationally design countermeasures against them?", from raw UniProt records all the way through molecular dynamics simulation of designed threat-antithreat complexes.

The biosecurity rationale is straightforward. Curated toxin databases such as the UniProt-tagged toxin and superantigen subsets cover only the proteins that have been deliberately collected and annotated as dangerous. The much larger pool of hypothetical and uncharacterized proteins in NCBI's non-redundant database is, by definition, not annotated, and conventional keyword and BLAST searches (Altschul et al., 1990) miss any threat whose closest characterized homolog falls below standard similarity thresholds. A classifier that operates on PLM embeddings rather than sequence-similarity scores can in principle recover such remote threats, and once a threat has been flagged, TRILL can then fold its structure, and design a candidate protein binder against it. The pipeline below executes that program end to end on the NR pool used for the cellulase and antimicrobial scans in Section 3.4.

Curating threats: keyword union and homology-controlled split

The positive class was defined as the union of three UniProt SwissProt keyword categories, namely toxin (`KW-0800`), superantigen (`KW-0766`), and prion (`KW-0640`). Any entry that additionally carried the antimicrobial keyword (`KW-0929`) was removed from the positive set, on the principle that antimicrobial peptides have legitimate biomedical applications and including them in the threat class would conflate "dangerous to humans" with "dangerous to bacteria". The negative set was every reviewed SwissProt entry that did *not* carry these keywords, save for antimicrobial.

To prevent homologs from leaking across the train/test split, I built a homology graph in which every sequence is a node and an edge is added between two sequences whenever they appear together in an MMseqs2 (Steinegger and Söding, 2017) `easy-search` alignment with positive bit-score *and* both sequences have a known origin (threat or non-threat) that agrees. The resulting graph yielded 5,457 connected components, each of which is single-origin by construction, with 9,982 non-threat and 5,138 threat sequences distributed across the clusters. The alignments that fed this graph were drawn from an all-versus-all MMseqs2 search of the combined positive–negative FASTA against itself in BLAST tabular format, and the per-sequence origin labels were assigned by majority vote across each query's alignment rows. With threats as the minority class, the target training count was set to $0.8 \times 5{,}138 = 4{,}110$ per class. Whole clusters were then shuffled and added greedily until the count was as close to 4,110 as possible, with a local-search pass swapping single clusters between the train and leftover pools whenever the swap reduced the gap. The resulting partition contains 4,110 threats and 4,110 non-threats in training (perfectly balanced, 8,220 sequences total) and 1,028 threats against 5,872 non-threats in testing (6,900 sequences, intentionally imbalanced as it absorbs all leftovers).

Held-out performance

A TRILL LightGBM (Shi et al., 2026) classifier trained on the 8,220-sequence balanced training set and evaluated on the 6,900-sequence test set produced the following performance at a probability threshold of 0.50, namely accuracy 0.981, balanced accuracy 0.979, precision 0.904, recall 0.977, F1 0.939, Matthews correlation coefficient 0.929, ROC AUC 0.996, and average-precision PR AUC 0.974. The per-class classification report is reproduced in Table 3.4.

The MCC of 0.929 is the highest of the three NR-scale classifiers in this chapter

Table 3.4: Per-class held-out classification report for the LightGBM threat classifier at probability threshold 0.50. Class 0 is the non-threat pool (any reviewed SwissProt entry without KW-0800), and class 1 is the threat union (KW-0800 ∪ KW-0766 ∪ KW-0640, excluding any entry also carrying KW-0929).

Class	Precision	Recall	F1	Support
0 (non-threat)	0.9959	0.9818	0.9888	5,872
1 (threat)	0.9037	0.9767	0.9388	1,028
Macro avg	0.9498	0.9792	0.9638	6,900
Weighted avg	0.9821	0.9810	0.9813	6,900

(cellulase 0.903, antimicrobial 0.796), despite the threat positive class being the least ancestrally coherent of the three. The most plausible interpretation is that the homology-controlled cluster split has, if anything, made the held-out test *harder* than the cellulase and antimicrobial splits (no test sequence shares a homology cluster with any training sequence) and that the classifier is recognizing pan-toxic features that span the toxin/superantigen/prion union rather than memorizing a specific family. This optimistic reading is partially deflated by the per-family analysis later in this section. The "Estimated false-positive contribution from known failure modes" paragraph in Section 3.8 identifies the chymotrypsin-fold S1 serine protease family as a likely source of host-enzyme false positives, indicating that part of what the classifier reads as "pan-toxic" is shared protease-fold architecture between snake-venom serine proteases and mammalian coagulation, fibrinolysis, and digestive enzymes. The class imbalance of the test set (~5.7:1 non-threat to threat) is also more representative of the real NR distribution than the balanced training set, and the precision of 0.904 on the threat class reflects the fact that a tiny minority can still be recovered at high quality from a long-tailed background.

Figure 3.16 reports the corresponding ROC, precision–recall, confusion matrix, and score-distribution panels in the same style as the cellulase and antimicrobial summaries (Figures 3.4 and 3.5).

NR-scale scan

After characterization on the held-out test set, the classifier was retrained on the full labeled pool (15,120 sequences) and applied to the same unified NR embedding set used for the cellulase and antimicrobial scans of Section 3.4. ESM2-650M embeddings for the full pool were computed with a single distributed TRILL `embed` run and passed to the retrained classifier through the `-preTrained` flag. The scan

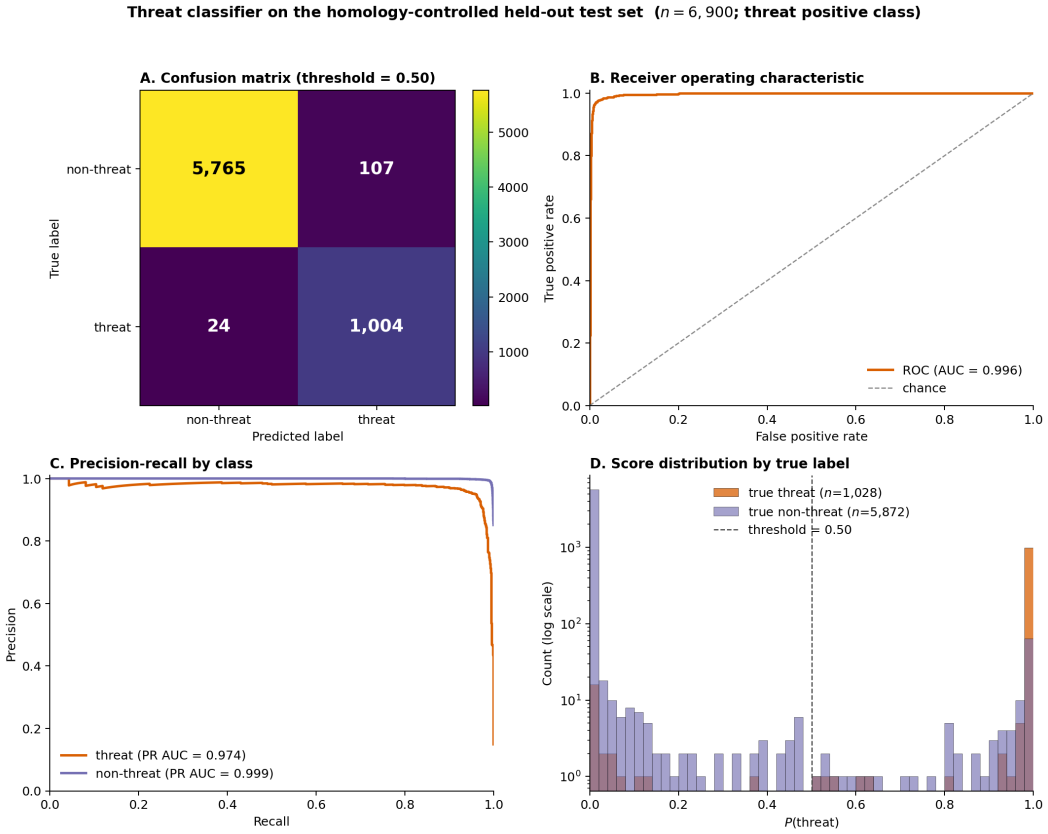


Figure 3.16: LightGBM threat-vs-non-threat classifier on the homology-controlled held-out test set (n = 6,900, with 1,028 threat sequences and 5,872 non-threats). **A**. Confusion matrix at probability threshold 0.50 (true positives 1,004, true negatives 5,765, false positives 107, false negatives 24). **B**. Receiver operating characteristic with AUC 0.996. **C**. Precision–recall curves for both classes, with PR AUC 0.974 for the threat class and 0.999 for the non-threat class. **D**. Score distribution by true label on a logarithmic count axis, with the threshold = 0.50 decision boundary marked. The two true-class score distributions are well separated and the long-tailed overlap near the decision threshold is small relative to the class-1 mass, consistent with the MCC of 0.929 reported in Table 3.4.

flagged 4,160,441 predicted threats, a predicted-positive rate of 2.05% on the NR pool. Of these, 2,603,331 (62.6% of predicted threats) carry a hypothetical or unknown function annotation in NCBI, which is the population that conventional keyword-based threat surveillance cannot see. The remaining 1,557,110 predicted threats overlap with sequences that carry some functional annotation and serve as a useful sanity check (they include, for example, well-known venom toxins and prion-related sequences that the classifier ought to recover). The 62.6% hypothetical fraction sits between the cellulase value (59.5%) and the antimicrobial value (46.3%) reported in Section 3.4, and the comparison is meaningful because the three classifiers were applied to the same NR pool. The threat classifier is the most prolific recoverer of hypothetical hits among the three property scans in this chapter.

Figure 3.17 reports the per-taxon predicted-threat rate for the threat trained-on-all classifier, stratified by NR phylum (left panel) and NR order (right panel). Taxa with fewer than 100,000 NR proteins are excluded and the top 25 taxa per panel by threat-classifier rate are shown. Both rankings are biologically interpretable. The phylum panel is led by viral phyla, Pisuviricota (~14% predicted-threat rate) and Uroviricota (~10%), reflecting that many viral protein families share the small, cysteine-rich, host-cell-disrupting signatures that the SwissProt KW-0800 training set was enriched for. Nematoda (~8%), Mollusca (~7%), Cnidaria (~5%), and Arthropoda (~4%) follow, consistent with the venom literature for cone snails (Mollusca), box jellyfish and sea anemones (Cnidaria), spiders and scorpions (Arthropoda), and the relatively under-explored parasitic-nematode toxin literature. The order panel surfaces *Araneae* (spiders, ~9%), the very order from which the spider-threat case study below was drawn, along with Hymenoptera (bees, wasps, ants), Lepidoptera, and Hemiptera among the highest-positive-rate animal orders. Ixodida (ticks) and Trichinellida (parasitic nematodes) also rank highly, reflecting their ectoparasitic and tissue-disruptive lifestyles. Among bacterial orders, Rickettsiales surfaces as a top hit, consistent with the obligate-intracellular and host-cell-manipulating biology of Rickettsia and its close relatives. However, these trends may simply be due to sequencing and collection biases rather than true biology.

Per-EC recall on the BRENDA toxin reference set A complementary independent validation uses the BRENDA enzyme database (Hauenstein et al., 2026), which curates a set of EC numbers explicitly flagged as toxin or toxin-related. The 29 BRENDA-toxin ECs with NR presence span snake-, spider-, and bee-venom enzymes, bacterial protein toxins, and plant ribosome-inactivating proteins. For each EC, the

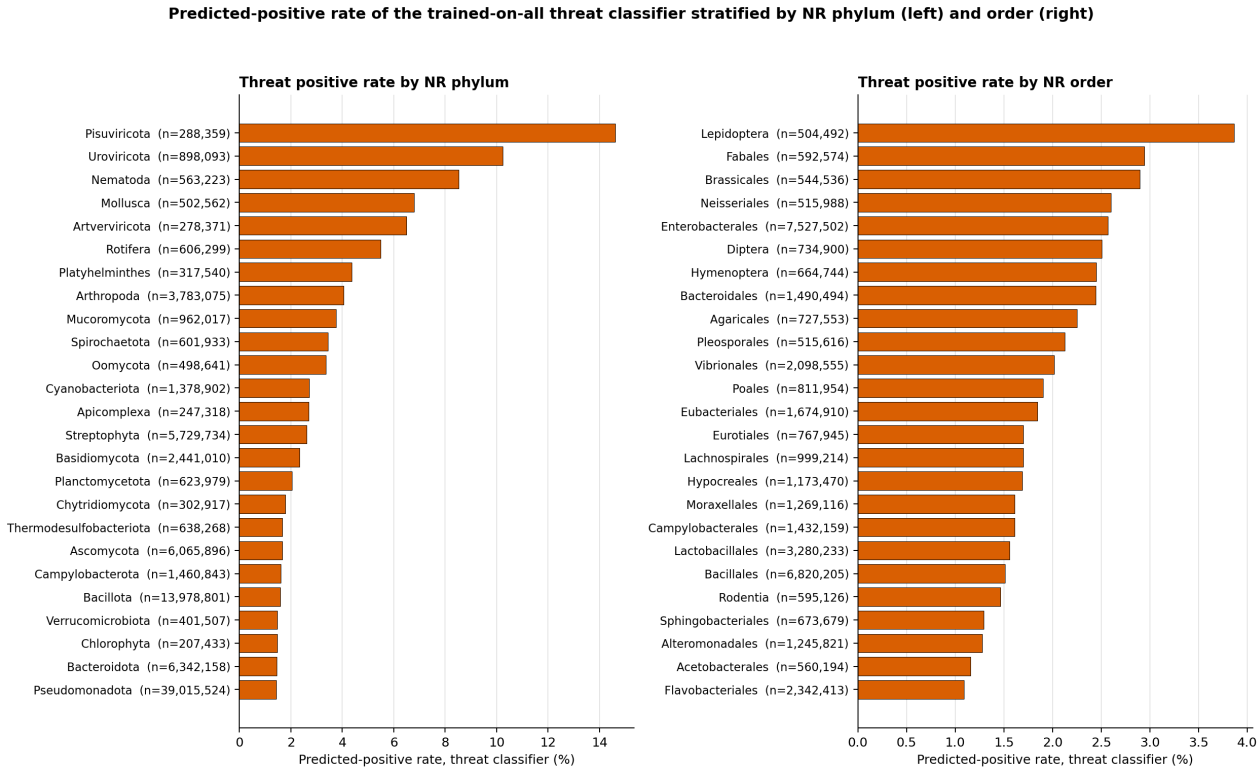


Figure 3.17: Predicted-positive rate of the trained-on-all threat classifier by NR phylum (left) and NR order (right). Taxa with fewer than 100,000 NR proteins are excluded, and within each panel the top 25 taxa by classifier-specific positive rate are shown. Viral phyla dominate the phylum-level ranking. *Araneae*, the order from which the spider-threat case study (Section 3.8) was drawn, leads the animal-order ranking, with Hymenoptera, Lepidoptera, Hemiptera, Ixodida, and Trichinellida also surfacing as venom- and toxin-rich taxa.

within-EC recall of the threat classifier, the fraction of NR proteins carrying that EC annotation that the classifier called positive, was computed alongside the same fractions for the cellulase and antimicrobial classifiers. Figure 3.18 reports the result, ordered by NR pool size per EC.

The threat classifier's per-EC recall spans the full range. Recall is highest for the enzymatic toxins most strongly enriched in the SwissProt KW-0800 training pool. Hyaluronoglucosaminidase (EC 3.2.1.35) is called positive on 83% of 21,956 NR proteins, the canonical venom "spreading factor" shared across snake, spider, and bee venoms (Kemparaju and Girish, 2006; Bordon et al., 2012). Sphingomyelin phosphodiesterase (EC 3.1.4.12) is recovered at 49% of 3,311, which includes the secreted bacterial neutral sphingomyelinases that drive the cytotoxic and hemolytic phenotypes of *Staphylococcus aureus* β-toxin and homologs in other pathogenic

Firmicutes. rRNA N-glycosylase (EC 3.2.2.22) is recovered at 39% of 580, the catalytic activity of ricin and the broader ribosome-inactivating-protein family (Stirpe, 2013). Phospholipase A2 (EC 3.1.1.4) is recovered at 17% of 3,187, the canonical viperid and elapid venom neurotoxin and myotoxin (Manjunatha Kini, 2003).

Recall falls essentially to zero on several other classic toxin ECs with non-trivial NR support. Phospholipase D (EC 3.1.4.4, 0% of 3,036) includes the *Loxosceles* brown-recluse-spider dermonecrotic toxin family (Chaim et al., 2011). Deoxyribonuclease I (EC 3.1.21.1, 0% of 1,865) is reported across viperid venoms as a contributor to envenomation pathology (Dhananjaya and D'souza, 2010). Cysteine-S-conjugate β-lyase (EC 4.4.1.13, 0% of 14,870) and tRNA-intron lyase (EC 4.6.1.16, 0% of 1,025) likewise have BRENDA-flagged toxin members that the classifier does not recover. The asymmetry is interpretable in light of how the threat training set was constructed. The SwissProt KW-0800 union is dominated by venom and bacterial protein toxins whose sequence signatures cluster densely in latent space, and EC families whose toxin members are sparsely represented relative to a much larger non-toxin homolog pool of the same EC (human DNase I, the many non-*Loxosceles* bacterial and plant PLDs, mitochondrial β-lyases) are under-represented in training and consequently under-recalled at deployment. A single EC value should therefore not be read as a single "toxicity" label, and per-EC or other function prediction agreement can be appropriate granularity for biosecurity-surveillance use. This EC-based screen also misses legitimate threats detected, since they do not have to be catalytic to be deadly.

A second observation is that the antimicrobial classifier also fires meaningfully on the BRENDA toxin reference set. It calls 42% of hyaluronoglucosaminidases positive and roughly 20–30% of pancreatic ribonucleases and EC 3.1.4.12 sphingomyelinases. This pattern is consistent with the Antimicrobial–threat two-way intersection of 324,869 sequences seen in the NR-wide UpSet plot of Figure 3.10, and it reflects the well-documented overlap between bacterial-pathogen-targeting antimicrobial activity and host-cell-damaging toxin activity. Many lytic AMPs (lysozymes, lysins, endolysins, defensins) share the small-protein, cysteine-rich, membrane-active character of several enzymatic venom and bacterial toxins. The practical implication for biosecurity-surveillance use of the threat classifier is that per-EC recall does not transfer uniformly from the SwissProt training distribution to the NR deployment distribution. A single probability threshold should be paired with an EC-stratified review before any predicted positive is acted on, and EC families with low historical

representation in SwissProt KW-0800 should be flagged as known blind spots.

The size of the hypothetical-protein threat pool is the operational point of this section. Manual inspection of 2.6 million predicted-threat hypothetical proteins is not viable, and the rest of the section demonstrates that the same TRILL framework that produced the prediction can also produce a computational countermeasure for any individual hit chosen for follow-up.

Spider-threat case study

To exercise the design half of the pipeline, I drew a case-study cohort from the predicted-threat hypothetical pool, seventeen sequences sampled from the subset of predicted threats that (i) are annotated as hypothetical or of unknown function, and (ii) come from the order *Araneae* (spiders). The order Araneae was chosen because spider venoms are an established source of cysteine-rich knottins and ion-channel-targeting peptides whose distant homologs are exactly the kind of remote threat the embedding-based classifier was designed to recover.

Structure prediction (`fold`) Each of the seventeen spider sequences was folded with ESMFold (Lin et al., 2023) via TRILL's `fold` command, producing one predicted PDB per sequence.

Surface-patch identification (custom script) For each predicted structure, per-residue solvent-accessible surface area (SASA) was computed with FreeSASA (Mitternacht, 2016), and a residue was labeled solvent-exposed if its SASA exceeded 30 $Å^2$. A sliding window of 50 residues was then scanned along the sequence, and a window was retained as a candidate binder-target patch if at least 70% of its residues were solvent-exposed and at least three of those residues were hydrophobic. For each threat, the single window with the largest hydrophobic-residue count was selected as the target patch for the subsequent design step. The hydrophobic-patch criterion biases the selection toward surfaces that are likely to support stable protein–protein interactions.

Binder backbone design (`diffusion_gen`) For each spider threat, TRILL's `diffusion_gen` command was invoked with RFDiffusion (Watson et al., 2023) in binder-design mode. The `-contigs` argument specified the chosen hydrophobic patch followed by a chain break (`/0`) and a binder length range of `100-200` residues, instructing RFDiffusion to diffuse a protein backbone of 100–200 amino acids whose

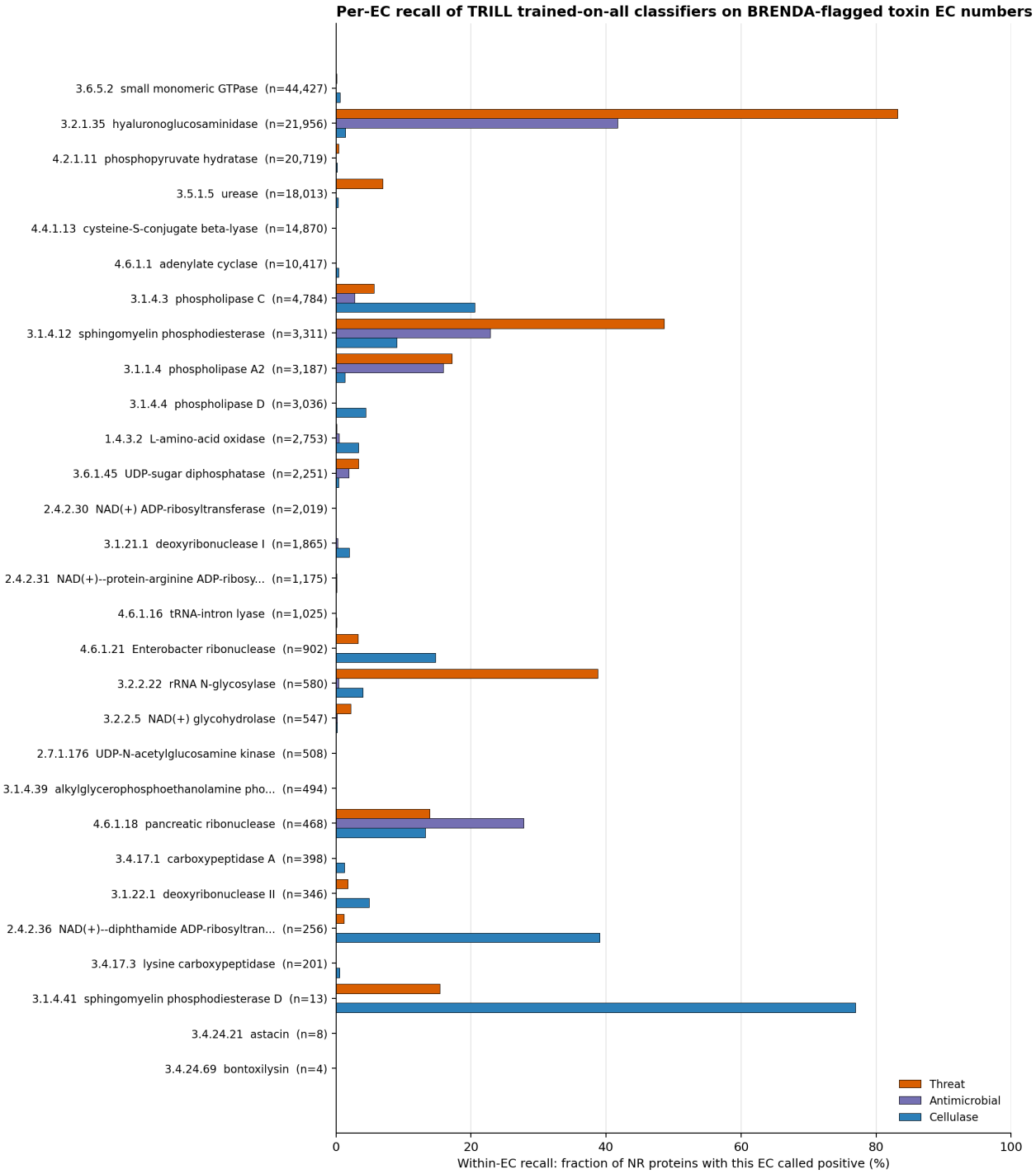


Figure 3.18: Per-EC recall of the three trained-on-all classifiers (threat, antimicrobial, cellulase) on the 29 EC numbers that the BRENDA enzyme database (Hauenstein et al., 2026) flags as toxin or toxin-related. Rows are ordered by the size of the NR pool for that EC (largest at top), and the within-EC recall is the fraction of NR proteins carrying that EC annotation that the classifier called positive. The threat classifier has high recall on hyaluronoglucosaminidases, sphingomyelin phosphodiesterases, ricin-family rRNA N-glycosylases, and phospholipase A2, and essentially zero recall on phospholipase D and deoxyribonuclease I despite each having thousands of NR-pool sequences.

interface contacts the patch. One hundred binder backbones were generated per threat.

Sequence design conditioned on the bound complex (`inv_fold_gen`) For each generated backbone, TRILL's `inv_fold_gen` command was invoked with LigandMPNN (Dauparas et al., 2025) using the full threat–anti-threat complex as input, so that the designed sequence is conditioned on the bound conformation rather than the free backbone alone. One thousand candidate sequences were sampled per backbone, and the top ten by LigandMPNN overall confidence were retained. Figure 3.19 shows the joint distribution of LigandMPNN overall confidence and sequence recovery across all sampled candidates. The two quantities are weakly correlated and span comparable ranges across the cohort, and the per-backbone top-ten cut-off is therefore acting as a top-confidence filter rather than a top-recovery filter, in line with the LigandMPNN authors' guidance to prioritize the model's own confidence metric over recovery when the backbones are non-native designs.

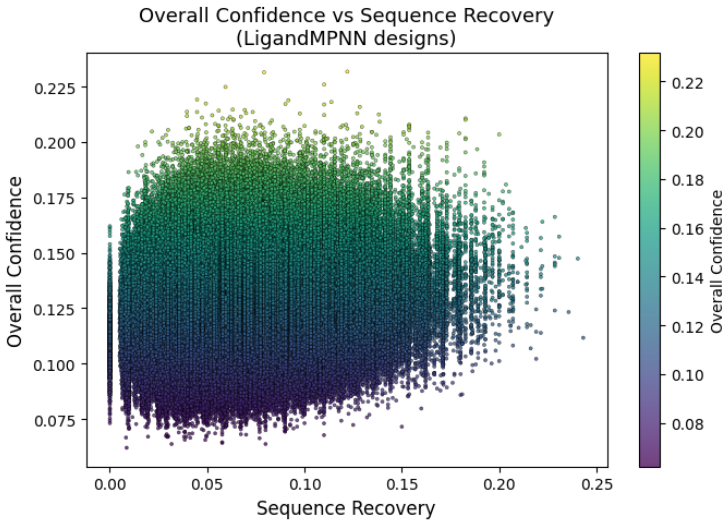


Figure 3.19: Joint distribution of LigandMPNN overall confidence and sequence recovery for the candidate sequences generated against each RFDiffusion binder backbone. Each point is one sampled sequence, with color encoding confidence. The two quantities are weakly correlated, and the top-ten-per-backbone filter used downstream selects by overall confidence rather than recovery.

Predicting the all-atom structures (`fold`) The top-10 LigandMPNN-derived sequences were folded with ESMFold to obtain a prediction of each monomeric structure complete with side-chains, providing a structural sanity check that the designed sequence agrees with the RFDiffusion backbone it was conditioned on.

Docking the threat-antithreat structures (`dock`) Each ESMFold-predicted anti-threat structure was docked against the original predicted threat with GeoDock (Chu et al., 2024) via TRILL's `dock` command, producing a top-ranked complex per pair.

Molecular dynamics (`simulate`) Each top-ranked threat–anti-threat complex that survived rfolding and docking was simulated with OpenMM (Eastman et al., 2017) through TRILL's `simulate` command using the Amber14 force field (`amber14-all.xml`) and the Hawkins–Cramer–Truhlar Generalized Born implicit solvent model (`implicit/hct.xml`, the TRILL default). Simulations were run for 1,000,000 timesteps of 2 fs each (2 ns per complex). After the pre-MD filtering, 892 candidate threat–anti-threat complexes reached the simulation stage, a >10× attrition from the LigandMPNN-derived pool, mainly from steric-clashes inherited from poorly-docked complexes. Per-frame interface RMSD (iRMSD), per-residue RMSF, and ProLIF (Bouysset and Fiorucci, 2021) interaction fingerprints were computed for each trajectory. The simulation step provides a physics-based check on whether each computational anti-threat candidate maintains interface contact with its target under thermal fluctuation, decoupled from the deep-learning models that produced it.

Results from the spider case study

The 892 simulated complexes were distributed unevenly across the seventeen-threat cohort, with binder counts per threat ranging from 1 to 154. To avoid drawing conclusions from per-target sample sizes too small to support a distributional read, the analysis below is restricted to the ten threats for which at least 40 binders survived to MD, namely `GFW20398.1` (n = 154), `GFU38284.1` (n = 133), `GIY00666.1` (n = 119), `GFS99196.1` (n = 75), `GFU59895.1` (n = 68), `GFU21108.1` (n = 67), `GFW50903.1` (n = 64), `GIY32514.1` (n = 49), `GFV32089.1` (n = 46), and `GFV16616.1` (n = 42), 817 complexes in total. The remaining seven targets, each with one to thirty-seven binders, are excluded from the per-target distributional analysis but are reported in the master cross-target table.

Choice of stability metric Standard CAPRI (Lensink, Méndez, and Wodak, 2007) classification of protein–protein docking solutions defines four quality tiers by interface RMSD (iRMSD), namely high quality (iRMSD $\leq$ 1 Å), medium ($\leq$ 2 Å), acceptable ($\leq$ 4 Å), and incorrect (> 4 Å), where iRMSD is computed on backbone atoms of interface residues after superposition. The CAPRI thresholds are calibrated against experimentally resolved native complexes. Here, since there is no

experimental reference, I use iRMSD as a stability metric instead, the time-average of frame-to-initial iRMSD over the post-equilibration window of the trajectory. A complex with low iRMSD has, by this measure, retained its starting interface geometry. The interface itself is defined here as the residues with any heavy atom within 5 Å of the partner chain at frame zero of the trajectory. Statistics are computed on the second half of each 1,000-frame trajectory, a more conservative window than the final-75% recommendation of Grossfield and Zuckerman (Grossfield and Zuckerman, 2009). Whole-complex RMSD is reported for context but is not used as the primary stability metric, because the toxin half-structures lack their native disulfide bonds in the simulation system and contribute substantial drift to the whole-complex signal regardless of binder quality.

Per-target stability profiles Figure 3.20 and Table 3.5 report, for each of the ten deeply-sampled threats, the distribution of post-equilibration iRMSD across all simulated binders and the fraction of binders falling into each CAPRI quality tier. The CAPRI medium-quality regime (iRMSD < 2 Å) is essentially empty across the cohort. Only one of the 817 simulated binders falls below the 2 Å threshold, and that binder is in `GFV32089.1` (1/46 = 2.2%). The more permissive 3 Å threshold spreads the targets out, with yields ranging from 0% (four targets, namely `GFW20398.1`, `GFU59895.1`, `GIY32514.1`, `GFV16616.1`, despite sampling 42 to 154 designs each) to 30.4% for `GFV32089.1` (14/46). `GFV32089.1` is also the only target whose CAPRI-acceptable yield (iRMSD < 4 Å) exceeds 50% (37/46 = 80.4%). Its top binder, `design_44_id=488`, is the only candidate in the entire cohort that comfortably clears the medium-quality threshold, with a post-equilibration iRMSD time-average of 1.82 Å and binder-only RMSD of 1.77 Å (Table 3.6). Figure 3.21 shows the decomposed RMSD trajectory for this binder over the full 2-ns simulation, decoupling the (substantial) toxin-only drift from the (small and stationary) binder-only and interface drift.

Table 3.6 reports the lowest-iRMSD binder for each deeply-sampled threat, selected purely by minimum iRMSD with binder-only RMSD as a tiebreaker. Six of the ten deeply-sampled threats yielded a candidate with mean iRMSD ≤ 3 Å (i.e., between CAPRI "medium quality" and "acceptable"), while the other four yielded only candidates in the "acceptable" or "incorrect" regime by the CAPRI scale.

Composite rankings disagree with iRMSD-based rankings The in-house analysis tooling for this pipeline produces a "composite rank" that averages five per-target

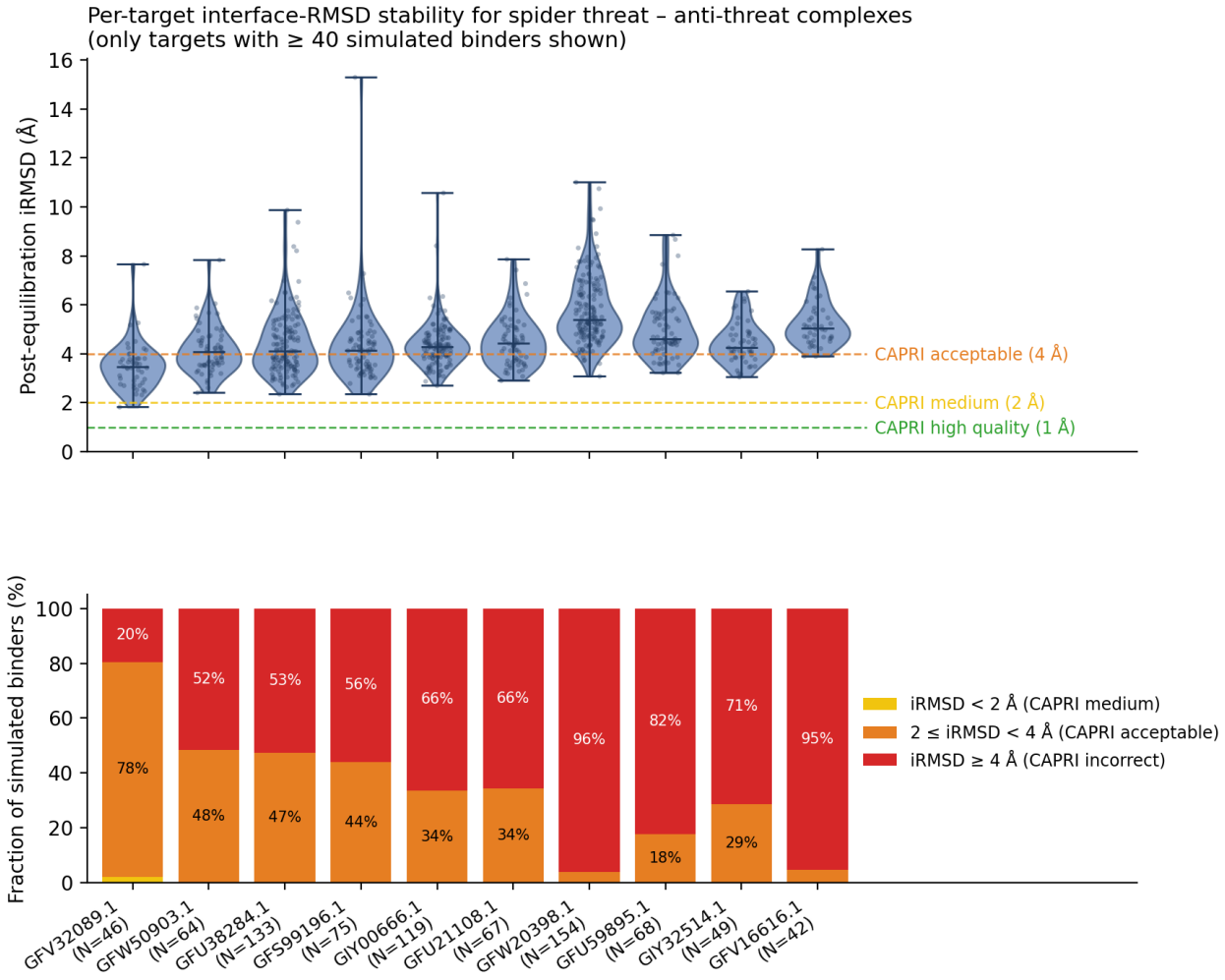


Figure 3.20: Per-target interface-RMSD stability across the ten spider threats with ≥ 40 simulated binders. **Top**. Violin plots of post-equilibration iRMSD per target (with all individual binders overlaid as points), annotated with the CAPRI high-quality (1 Å), medium (2 Å), and acceptable (4 Å) thresholds. **Bottom**. Stacked-bar fraction of each target's binders falling into the CAPRI medium (< 2 Å), acceptable (2–4 Å), and incorrect (≥ 4 Å) regimes. Targets are ordered by the medium-quality yield (left to right).

Table 3.5: Post-equilibration interface RMSD distribution and yield of stable binders for the ten threats with ≥ 40 simulated binders. Yields are the fraction of binders whose time-averaged iRMSD over the second half of the trajectory falls below the CAPRI "acceptable" threshold of 4 Å and the more stringent 3 Å threshold. `GFV32089.1` is the only target with a yield substantially above the noise floor.

Target	***N***	**iRMSD median (Å)**	**iRMSD min (Å)**	**Yield < 3 Å**	**Yield < 4 Å**
GFV32089.1	46	3.47	1.82	30.4%	80.4%
GFW50903.1	64	4.07	2.42	7.8%	48.4%
GFU38284.1	133	4.10	2.37	7.5%	47.4%
GFS99196.1	75	4.15	2.36	2.7%	44.0%
GIY00666.1	119	4.28	2.72	1.7%	33.6%
GFU21108.1	67	4.43	2.92	1.5%	34.3%
GFW20398.1	154	5.39	3.08	0.0%	3.9%
GFU59895.1	68	4.61	3.22	0.0%	17.6%
GIY32514.1	49	4.24	3.06	0.0%	28.6%
GFV16616.1	42	5.05	3.89	0.0%	4.8%

Table 3.6: Lowest-iRMSD binder per spider threat among the ten threats with ≥ 40 simulated binders. Selection is by minimum post-equilibration iRMSD, with binder-only RMSD as a tiebreaker. iRMSD, bRMSD, and tRMSD are post-equilibration time-averages of interface, binder-only, and toxin-only backbone RMSD, respectively, in Å. Contacts and H-bonds are persistent ProLIF interactions (≥ 0.5 and ≥ 0.3 frame occupancy, respectively).

Target	**Design**	**iRMSD**	**bRMSD**	**tRMSD**	**RMSF**	**Contacts**	**H-bonds**
GFV32089.1	design_44_id=488	1.82	1.77	3.69	0.99	43	18
GFU38284.1	design_10_id=562	2.37	2.86	3.28	1.28	39	14
GFS99196.1	design_37_id=274	2.36	2.41	3.56	1.20	34	9
GFW50903.1	design_1_id=858	2.42	2.73	4.42	1.41	47	19
GIY00666.1	design_1_id=681	2.72	4.55	5.12	1.60	60	24
GFU21108.1	design_8_id=57	2.92	4.32	6.37	1.34	34	15
GIY32514.1	design_86_id=217	3.06	3.99	4.70	1.61	46	14
GFW20398.1	design_49_id=486	3.08	2.51	8.38	1.48	61	29
GFU59895.1	design_12_id=901	3.22	2.37	5.74	1.82	46	18
GFV16616.1	design_26_id=603	3.89	3.73	8.10	1.97	35	18

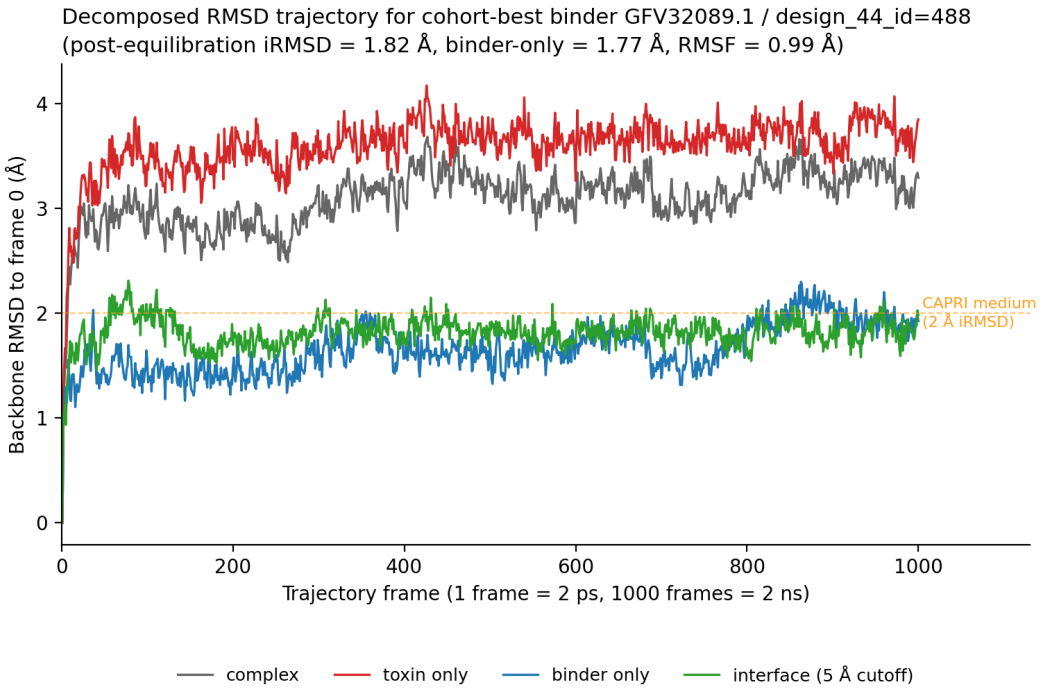


Figure 3.21: Decomposed RMSD trajectory for the cohort-best threat–anti-threat complex (`GFV32089.1` threat paired with `design_44_id=488` binder) over the 1,000-frame, 2-ns simulation. Four backbone-RMSD time series are shown, namely whole complex (gray), toxin chain only (red), binder chain only (blue), and the residues at the 5-Å-cutoff interface (green). The interface and binder-only signals plateau below 2 Å for the duration of the trajectory, while the toxin chain drifts steadily upward, the latter reflecting the loss of native disulfide bonds in the pre-MD preparation step rather than a defect of the binder. The 2-Å CAPRI medium-quality reference is shown for context.

rankings (iRMSD, binder-only RMSD, RMSF, and the counts of persistent ProLIF contacts and H-bonds). For every one of the ten deeply-sampled targets, the composite-rank-1 binder is a different design than the minimum-iRMSD binder reported in Table 3.6. The composite-rank-1 picks are systematically higher iRMSD (typical 3.1–4.9 Å) and higher contact density, reflecting a trade-off in which the composite metric rewards interface contact richness even when the interface is geometrically less stable. This is an instance of a broader caution. Averaging across orthogonal axes, namely a geometric-stability score and a contact-density score that respond to different physical signals, dilutes both. High-contact-density designs can rank well on the composite even when their interface drifts under thermal fluctuation, and tightly-held interfaces with sparse contact networks are penalized for properties they were not selected against. Contact density and interface stability are

also not the same quantity. Density measures how richly an interface is populated at the starting frame, whereas stability measures whether that population persists across the post-equilibration trajectory, and a binder can score highly on one without scoring highly on the other. Heuristic composite scores that conflate the two are not interpretable as a stability metric and should not be used as the sole basis for follow-up selection.

Persistent contacts and interaction fingerprints Figure 3.22 summarizes the persistent inter-chain ProLIF interactions across the cohort. The left panel shows that the relationship between interface stability (iRMSD) and contact density is weak. Binders with low iRMSD do not systematically have more persistent contacts than binders with high iRMSD, and the cohort spans roughly 30–110 persistent contacts across all stability tiers. The right panel decomposes the persistent-contact count by ProLIF interaction type and reports per-target means. Van der Waals contacts dominate (34–60 per binder on average, depending on target), followed by hydrophobic contacts (6–13), H-bonds (a sum of H-bond donor and acceptor counts, 13–24), salt bridges (anionic plus cationic, 4–8), and π interactions (cation–π, π–stacking, edge-to-face, and face-to-face combined, near zero across the cohort). The cohort-best binder `GFV32089.1/design_44_id=488` sits at 18 persistent H-bonds and 43 high-occupancy contacts, comfortably inside the natural-interface range.

Discussion

The end-to-end pipeline above passes through ten TRILL commands, namely `embed` (twice, once for training and once for the NR scan), `classify` (once for the training and then for evaluation), `fold` (for the seventeen spider threats and again for the surviving anti-threat sequences), `diffusion_gen`, `inv_fold_gen`, `dock`, and `simulate`.

Heterogeneous positive classes are not as fragile as the family-coherence framing suggests The recurring theme of Section 3.5 and Section 3.4 is that family-based generation hit rates are higher for ancestrally coherent classes than for function-defined classes. The classification side of the question is more forgiving. The threat classifier merges three mechanistically distinct categories (toxin, superantigen, prion) into a single positive class and still reaches MCC 0.929 on a homology-controlled held-out test, comfortably above both the cellulase (0.903) and antimicrobial (0.796) classifiers from the same chapter. The asymmetry is consistent with the broader

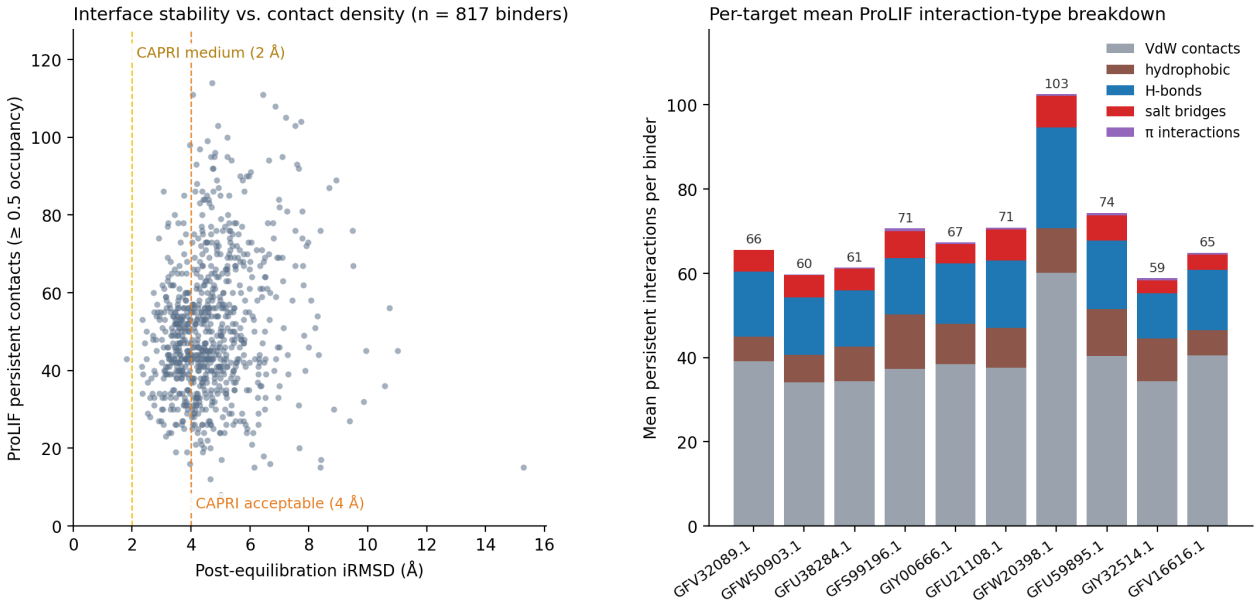


Figure 3.22: ProLIF persistent-interaction summary for the 817 threat–anti-threat complexes from targets with ≥ 40 simulated binders. **Left**. Per-binder count of persistent contacts (any ProLIF interaction with ≥ 0.5 frame occupancy) versus post-equilibration iRMSD. The CAPRI medium (2 Å) and acceptable (4 Å) thresholds are annotated. The relationship between interface stability and contact density is weak, motivating the analytical separation of the two quantities. **Right**. Per-target mean of persistent interactions decomposed by ProLIF interaction type. Counts are extracted directly from the per-frame ProLIF fingerprints (occupancy ≥ 0.5, grouped by the interaction-type field of each column). Bar-top numerals give the mean total persistent contacts per binder for each target. VdW contacts dominate, followed by hydrophobic contacts and H-bonds, and π-interaction counts are near zero across the cohort.

observation in this thesis that PLM embeddings carry enough functional signal to support classification of heterogeneous unions.

Limitations Several caveats should accompany this section. First, the "threats" that emerge from the NR scan are computational predictions. Biochemical confirmation requires expression and functional assays that are well outside the scope of a thesis chapter. Second, the hydrophobic-patch heuristic for selecting a binder-target surface is a hand-tuned filter rather than a principled binder-site prediction. Hot-spot prediction methods that incorporate evolutionary or interaction-energy information would likely produce different (and possibly better) target patches. Third, the molecular dynamics step uses implicit solvent and a 2-ns trajectory per complex. This is sufficient to identify candidates that drift apart rapidly but is not sufficient to

estimate binding free energies or to rule out slow conformational rearrangements. Fourth, the potentially missing disulfide connectivity from the predicted threat structures, which produces substantial drift in the toxin chain that is unrelated to binder quality. This is the reason iRMSD rather than whole-complex RMSD is used as the primary stability metric throughout this section. Fifth, generalized Born implicit-solvent models, including the Hawkins–Cramer–Truhlar variant (`igb=1`) used here, do not reliably reproduce the peptide secondary-structure equilibria recovered by explicit-solvent simulations, with the direction and magnitude of the deviation in α-helical content depending sensitively on the paired force field (Lang et al., 2022). For a cohort of conformationally unconstrained designed binders, absolute iRMSD values should therefore be read with this caveat in mind, and explicit-solvent confirmation of the highest-yield binders would be a natural next step. Each of these limitations is also a natural place where additional TRILL commands (for example, `score` for ProteinMPNN- or LigandMPNN-based binding scoring, or extended explicit-solvent simulation via `simulate` with TIP3P) can be inserted without disturbing the rest of the pipeline.

3.9 BioSentinel: an agentic cascade for detecting AI-designed toxin mimetics

Motivation and attribution

The threat-detection pipeline in Section 3.8 was designed against natural, hypothetical-function NR proteins. A different and more recently sharpened threat surface is the one created by generative protein design itself. AI-driven workflows that compose protein language models, structure predictors, and inverse-folding tools can generate *new-to-nature* sequences that adopt natural toxin folds while sharing no detectable sequence similarity with their natural parents. Such mimetics are by construction invisible to BLAST-based and HMM-based screening as used by commercial DNA-synthesis providers, and they constitute a dual-use biosecurity gap that classical homology tools cannot close. BioSentinel is an agentic, compute-aware cascade designed to close that gap.

Attribution BioSentinel is joint work with Matt Thomson (PI) and Arjuna Subramanian. Subramanian developed the FoldTuning algorithm that underpins the mimetic-generation step described in Section 3.7 above and contributed two prior foldtuned mimetic sets used here, 1,892 three-finger-toxin and 2,207 knottin sequences, together with the CLEAN-based enzyme-commission predictions used for functional validation in Section 3.9. The remaining elements of BioSentinel,

the curated 15-keyword toxin dataset, the homology-aware split, the fine-tuned ESM2 classifier, the new dermonecrotic / myotoxin / neurotoxin / hemotoxin / complement-impairing foldtuned mimetics, the classical-vs.-PLM benchmark, and the ProstT5+Foldseek structural-backstop cascade, are my contribution.

Curated 15-keyword toxin dataset

The threat classifier in Section 3.8 was trained on the broad SwissProt KW-0800 toxin keyword. For BioSentinel I curated a narrower training set explicitly focused on human-directed threats by selecting 15 specific UniProt keywords spanning functional categories of acute mammalian toxicity, namely complement-system-impairing toxins, dermonecrotic toxins, hemostasis-impairing toxins (with separate sub-keywords for fibrinogenolytic, fibrinolytic, blood-coagulation-cascade-activating, blood-coagulation-cascade-inhibiting, platelet-aggregation-activating, and platelet-aggregation-inhibiting activities), hemorrhagic toxins, myotoxins, presynaptic neurotoxins, postsynaptic neurotoxins, acetylcholine-receptor-inhibiting toxins, and ionotropic-glutamate-receptor-inhibiting toxins.

The resulting positive set contains 2,176 experimentally validated toxin sequences. The non-toxin negative pool was constructed by inverting the toxin-keyword query across the full SwissProt reviewed set, clustering with MMseqs2 `easy-cluster` (Steinegger and Söding, 2017), and randomly sub-sampling to a balanced, non-redundant negative training set.

Homology-aware train/test split

A central methodological choice was the construction of a strictly homology-aware train/test partition. Standard random splits on protein-classification datasets overstate generalization performance by allowing evolutionary ”cousins” of test sequences to appear in training, where they inflate accuracy without exercising the classifier's ability to recognize novel threats. To simulate the actual deployment regime, that is, detecting new-to-nature AI-designed sequences with no precedent in the training distribution, I ran all-versus-all MMseqs2 alignments across the toxin pool and enforced strict separation, so that no training toxin shares any detectable sequence similarity with any test toxin. Table 3.7 summarizes the resulting partition.

Fine-tuned ESM2 classifier

In contrast to the LightGBM-on-frozen-embedding approach used in Section 3.8, the BioSentinel classifier is an end-to-end fine-tuned ESM2 with a binary classification

Table 3.7: Homology-aware train/test split for the BioSentinel toxin classifier. The partition is constructed so that no training toxin shares detectable sequence similarity with any test toxin under MMseqs2 `easy-search`, mimicking the new-to-nature deployment regime.

Split	Toxin	Non-toxin	Total
Train	1,732	1,730	3,462
Test	444	1,269	1,713
Total	2,176	2,999	5,174

head. Fine-tuning was performed through TRILL's `finetune` command for 3 epochs at learning rate 10^{-4}. The end-to-end fine-tune is more expensive than embedding-plus-LightGBM but lets every layer of the encoder adapt to toxin/non-toxin discrimination, which proves to matter substantially on the new-to-nature test regime.

On the homology-aware test set, the classifier achieved accuracy 0.954, balanced accuracy 0.956, precision 0.875, recall 0.960, F1 0.915, and Matthews correlation coefficient 0.885, with the per-class breakdown shown in Figure 3.23. An MCC of 0.885 on a split that explicitly forbids homologous-sequence overlap is the metric that matters here, since the classifier is generalizing to genuinely novel toxin sequences rather than interpolating between seen examples.

Foldtuned mimetic generation

To probe the classifier on the threat surface it was actually built to defend against, namely AI-designed new-to-nature mimetics, I generated foldtuned mimetics using the algorithm of Subramanian et al. (Subramanian et al., 2023) as implemented in TRILL's foldtuning workflow (Section 3.7). The mimetic-generation step is intentionally adversarial in design. Foldtuning drives a generative protein language model to produce sequences that adopt a target fold or function while maximizing divergence from the natural sequence distribution, which is precisely the evasion strategy that a sophisticated adversary would employ.

Five rounds of foldtuning across five toxin classes yielded 6,027 new mimetics. Combined with the 1,892 three-finger-toxin and 2,207 knottin foldtuned mimetics contributed by Subramanian from prior work, the total foldtuned mimetic pool used for downstream benchmarking comprises 10,126 sequences. Table 3.8 reports the per-class breakdown.

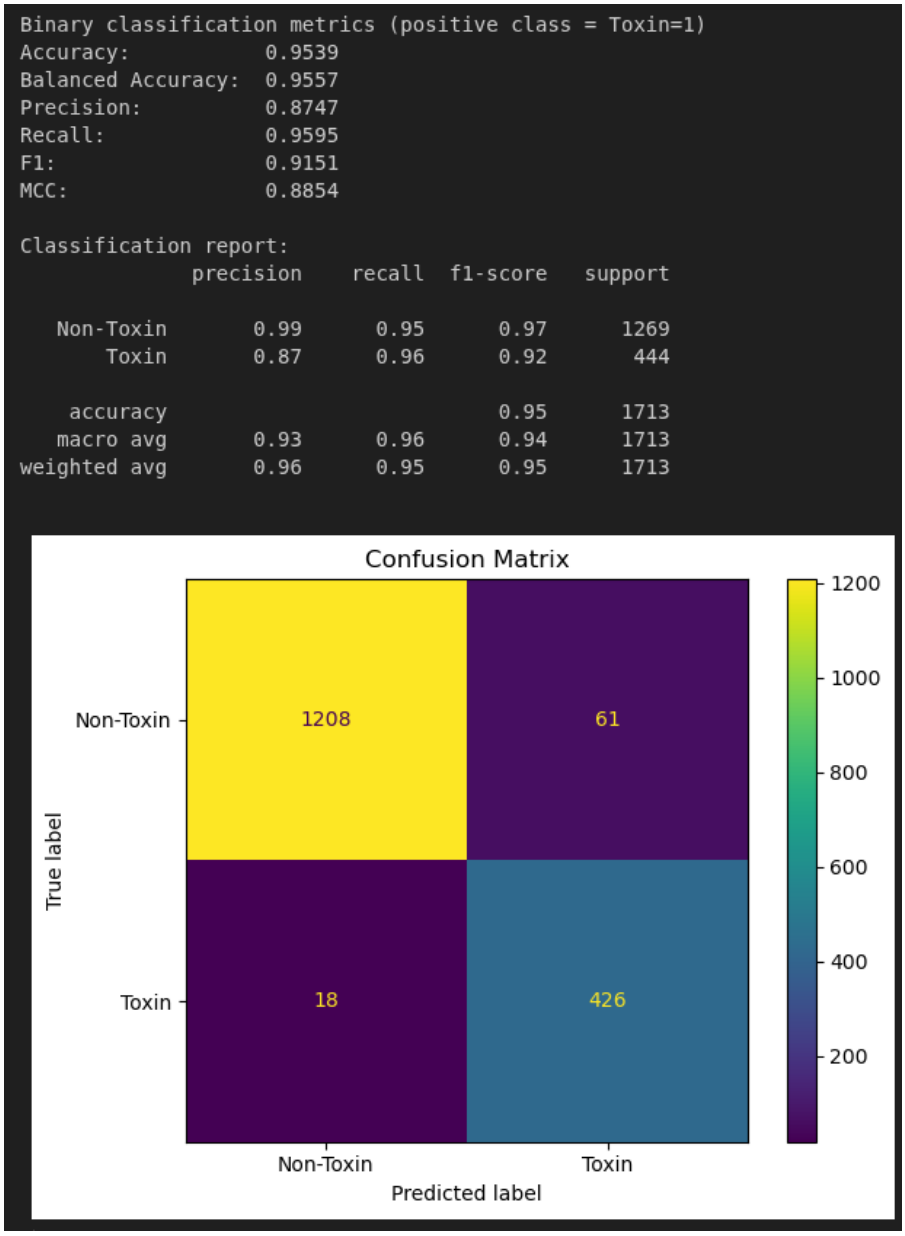


Figure 3.23: Performance of the BioSentinel fine-tuned ESM2-t12-35M binary toxin classifier on the homology-aware held-out test set (Table 3.7). Top, classification metrics and per-class breakdown. Bottom, confusion matrix showing 1,208 true negatives, 61 false positives, 18 false negatives, and 426 true positives.

Table 3.8: FoldTuned toxin mimetics used as a benchmark. The natural-sequence column counts SwissProt-curated toxins of that class in the BioSentinel positive set, and the mimetic column counts new-to-nature foldtuned sequences. Three-finger toxins and knottins were generated by Subramanian as part of prior foldtuning work, while the other five classes were generated for BioSentinel. The natural-sequence column sums to 2,297 rather than the 2,176-sequence positive set total of Table 3.7 because a single SwissProt entry can carry more than one of the 15 curated toxin keywords (e.g., a hemorrhagic and fibrinogenolytic snake-venom protease appears in both rows), and the 121-entry excess is the keyword-multiplicity artifact, not a counting error.

Toxin class	**Natural (SwissProt)**	**FoldTuned mimetics**
Complement-impairing	36	6
Dermonecrotic	216	2,480
Hemotoxin	1,266	588
Myotoxin	154	2,719
Neurotoxin	625	234
Three-finger toxin (prior)	—	1,892
Knottin (prior)	—	2,207
Total	2,297	10,126

Across-class variability in generation yield is striking. Myotoxins, with only 154 natural training exemplars, produced 2,719 mimetics, while neurotoxins, with 625 natural exemplars, produced only 234. This class-specific variability in foldtuning efficiency is not fully explained by training-set size and is consistent with the broader pattern (Sections 3.4 and 3.5) in which fold-defined classes are easier to generate from than function-defined classes whose members share little ancestral structure. A representative example of structural mimicry is shown in Figure 3.24, where the natural toxin backbone (green) and a foldtuned mimetic (blue) align closely in tertiary structure despite divergent sequence.

Functional validation via CLEAN (external tool)

A structurally faithful mimetic is not automatically a functional threat. To probe whether the foldtuned mimetics carry biologically meaningful toxin signal rather than only structural similarity, we applied CLEAN (Contrastive Learning-Enabled Enzyme Annotation) (Yu et al., 2023) to predict enzyme-commission numbers for each generated mimetic. Of the 10,126 mimetics, 7,910 (78.1%) received a CLEAN top-EC prediction that falls within the toxin-associated EC list compiled for BioSentinel.

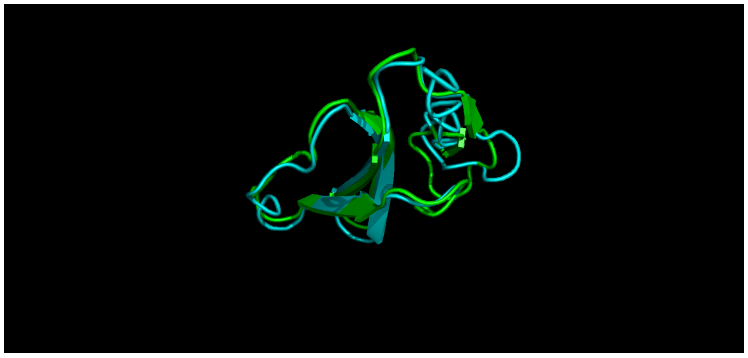


Figure 3.24: Backbone alignment of a natural toxin (green) and a FoldTuned mimetic (blue). The natural toxin is muscarinic toxin-like protein 3 from *Naja kaouthia* (monocled cobra). The structures superimpose closely despite the mimetic sharing no detectable sequence similarity with any known toxin under BLAST search (Figure 3.25).

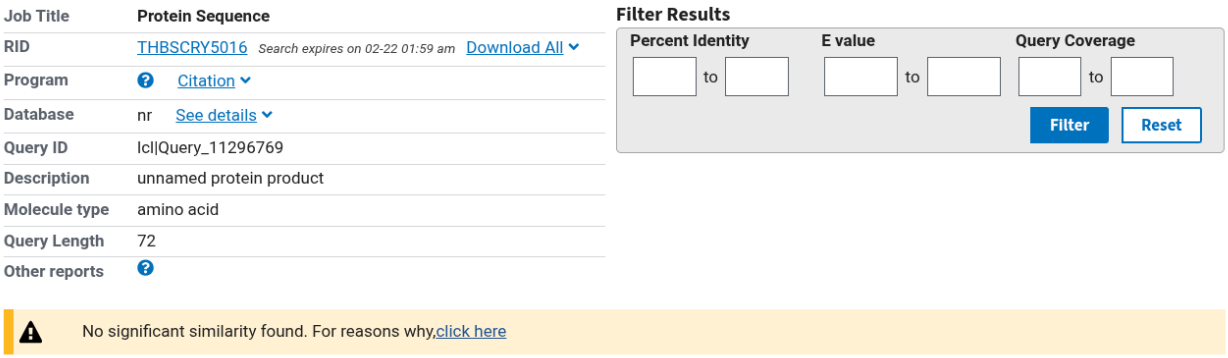


Figure 3.25: NCBI BLASTP search of a 72-residue FoldTuned toxin mimetic (the same mimetic shown in Figure 3.24) against the full NR database returns "No significant similarity found." This is representative of the BioSentinel mimetic pool, where classical sequence-alignment screening will miss this and structurally similar sequences entirely.

This number should be read as a coarse toxin-versus-non-toxin sorting signal rather than as functional validation. CLEAN was run in its forced-prediction mode without the confidence-gating model, so every input therefore receives a best-guess top-1 EC label regardless of how confidently CLEAN distinguishes its prediction from the next-best alternative. A matched non-toxin negative pool would also accumulate some CLEAN-predicted toxin-associated ECs at a baseline rate that we have not measured, so the 78.1% figure is best interpreted as "the foldtuned mimetics' top-1 CLEAN labels concentrate inside the toxin-associated EC neighbourhood" rather than as "78.1% of mimetics are functionally validated as toxins."

Note on tooling. CLEAN is run outside TRILL. The CLEAN predictions for

BioSentinel were produced by Subramanian using the upstream CLEAN distribution as an external step, and the resulting per-mimetic EC labels were imported into the BioSentinel analysis pipeline. The functional-validation step is therefore composed across two software environments rather than executed end-to-end inside TRILL.

Classical vs. PLM-based detection on Foldtuned mimetics

The headline empirical result of BioSentinel is the side-by-side benchmark of classical sequence-alignment detection against fine-tuned PLM-based detection on the foldtuned mimetic set. Four detection strategies were evaluated, in increasing order of training-data scale, as follows.

1. **MMseqs2 sequence alignment against the BioSentinel training-set toxins** (1,732 toxins, the most adversarial alignment baseline).
2. **ESM2-t12-35M fine-tuned on the training-set toxins only** (the strict-homology-aware version of the BioSentinel classifier in Section 3.9).
3. **MMseqs2 sequence alignment against the full BioSentinel toxin set** (train+test, 2,176 toxins).
4. **ESM2-650M fine-tuned on the full toxin set**.

Figure 3.26 reports the fraction of the 10,126 foldtuned mimetics that each method correctly flags as a toxin.

The biosecurity-relevant numbers are the gap between alignment-on-all (61%, 6,253/10,126) and PLM-on-all (98%, 9,951/10,126). Classical alignment, with full access to every toxin sequence in the curated set, would miss 3,873 of the 10,126 mimetics, a 38% miss rate. This is a direct, policy-relevant quantification of the AI-design biosecurity gap. A classical-only screening pipeline of the kind used today by commercial DNA-synthesis providers would fail to flag more than a third of foldtuned mimetics, regardless of how comprehensive their reference database is.

The strict-homology-aware variant of the BioSentinel classifier (ESM2-t12-35M trained on the train split only) captures only 22% of mimetics, a deliberately conservative number that establishes the floor. When the classifier sees only 1,732 training toxins with zero detectable similarity to any test sequence, it still flags better than chance on adversarial mimetics. The production-grade variant (ESM2-650M, all available toxins) is the operational classifier, and the 35M training-only number is reported as a methodological calibration.

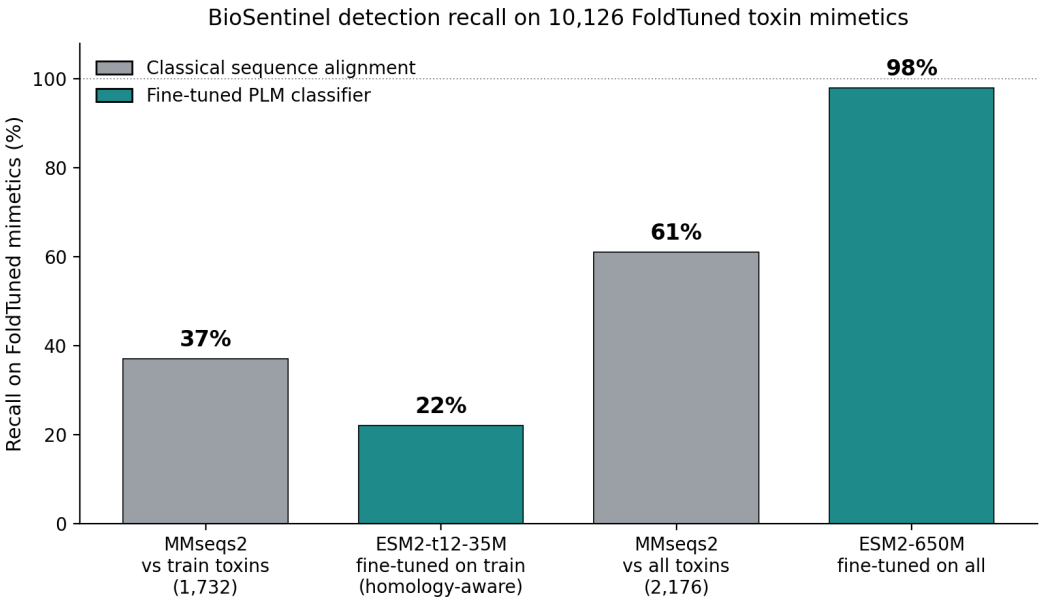


Figure 3.26: Percentage of 10,126 FoldTuned toxin mimetics correctly classified as toxins, by method. Left to right, MMseqs2 alignment against the training-set toxins only (37%), fine-tuned ESM2-t12-35M trained on the training set only (22%, the strict homology-aware variant), MMseqs2 alignment against the full toxin set (61%), and ESM2-650M fine-tuned on the full toxin set (98%). Classical alignment, even with access to the entire curated toxin pool, misses 3,873 of 10,126 AI-designed mimetics, and the fine-tuned PLM closes nearly all of that gap.

Closing the last gap: ProstT5+Foldseek as a structural backstop

Even the production-grade PLM classifier misses 175 of the 10,126 mimetics (∼1.7%). For a screening system intended to be the line of defense against AI-designed threats, 1.7% leakage is too high. To close the remaining gap, BioSentinel incorporates structural-alignment. Foldseek (Van Kempen et al., 2024) performs structural search using the 3Di structural alphabet, and is capable of identifying remote structural homologs that share no sequence similarity to known toxins. The conventional pipeline for Foldseek query construction requires predicting the query's 3D structure with a tool like ESMFold and then deriving 3Di tokens from the predicted coordinates, an expensive step that dominates the cost of structural screening.

BioSentinel sidesteps this cost by leveraging ProstT5 (Heinzinger et al., 2024), the bilingual amino-acid↔3Di T5 language model, to predict 3Di tokens directly from the input sequence in a single inference call. On a benchmark of 2,207 foldtuned knottins, ProstT5 generates 3Di tokens on an H200 GPU in 41 seconds, while folding the same sequences with ESMFold takes 34 minutes, a ∼50× speedup. The substitution is the same one validated in Section 3.7 (fast foldtuning), reused here as

a screening-pipeline component.

Figure 3.27 shows the resulting three-way overlap of detection across MMseqs2, the fine-tuned ESM2 classifier, and the ProstT5+Foldseek backstop. The three methods together capture all 10,126 mimetics. Each method covers cases the others miss. Foldseek-only captures 40 mimetics that neither alignment nor PLM detects, and ESM-only captures 22 that neither alignment nor Foldseek detects. The all-three intersection (6,115 mimetics) is the easy regime where any single method suffices.

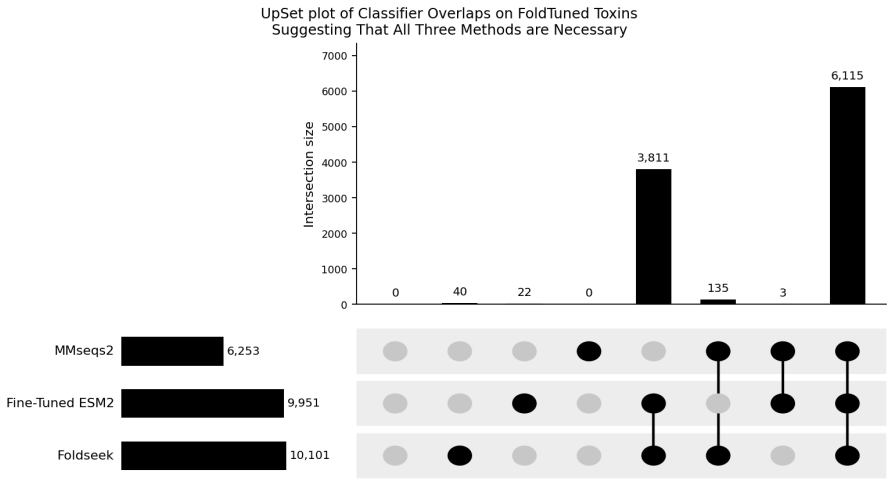


Figure 3.27: Three-way overlap of mimetic detection across the BioSentinel methods. The Foldseek-only intersection (40) and the ESM-only intersection (22) demonstrate that no single method captures every mimetic, and the cascade requires all three. The all-three intersection (6,115) is the regime where the cheapest method (alignment) would suffice in a compute-aware deployment.

Agentic cascade architecture

The empirical result above implies the deployment architecture, namely an agentic cascade in which each input sequence is routed through methods in increasing order of compute cost, with early-exit decisions at each step. Figure 3.28 illustrates the BioSentinel cascade.

The cascade concentrates expensive inference (the fine-tuned ESM2 forward pass and the ProstT5 encode + Foldseek search) on the small fraction of queries that the cheaper methods leave ambiguous. On the BioSentinel benchmark, step 1 (MMseqs2 against the curated toxin set) catches all 6,253 mimetics with detectable homology to a known toxin, namely the union of the all-three intersection (6,115), the MMseqs2-and-ESM2-but-not-Foldseek partition (135), and the MMseqs2-and-Foldseek-but-not-ESM2 partition (3), with no mimetics in the MMseqs2-only partition (Figure 3.27). Step 2

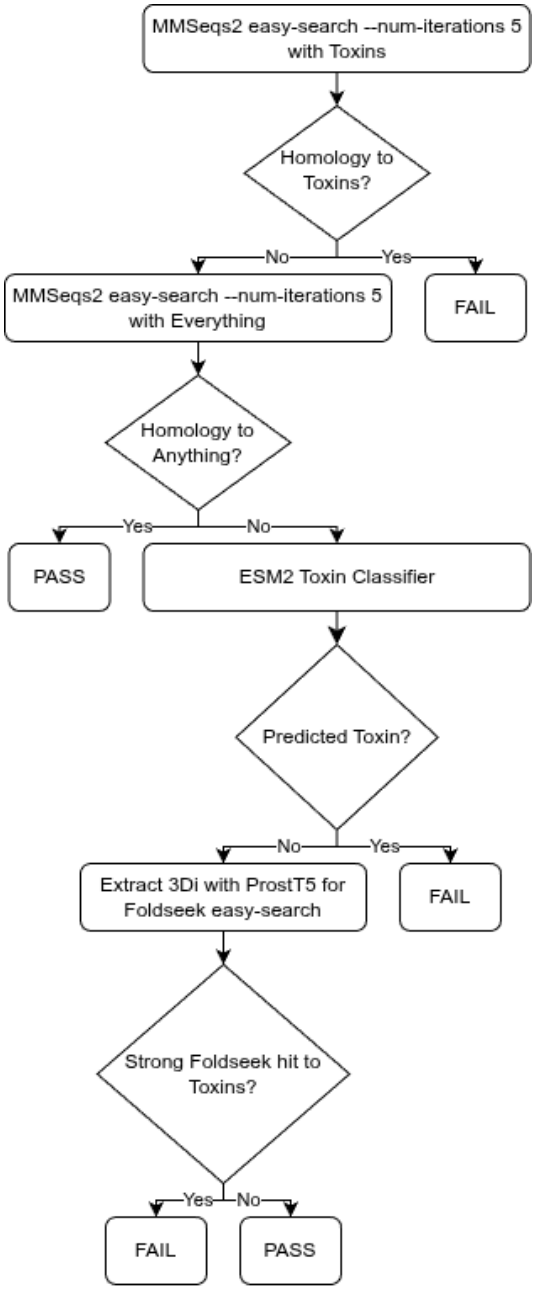


Figure 3.28: Preliminary BioSentinel workflow for screening candidate toxic proteins. Each input sequence is routed through methods in increasing order of compute cost, namely iterative MMseqs2 against the curated toxin set, then MMseqs2 against the broader sequence universe, then the fine-tuned ESM2 toxin classifier, and finally ProstT5+Foldseek structural alignment as a high-specificity backstop. Each step can early-exit with FAIL (predicted threat) or PASS (cleared), and only ambiguous inputs are escalated to the next step.

(MMseqs2 against the broader sequence universe) is a clearing rather than a catching step and contributes no positive detections on the foldtuned mimetic pool, all of which are by construction toxin-like. Step 3 (the fine-tuned ESM2 classifier, which sees only the 3,873 mimetics step 1 missed) catches a further 3,833, the 22 ESM-only sequences plus the 3,811 that ESM2 and Foldseek both flag while MMseqs2 misses. Step 4 (ProstT5+Foldseek) catches the residual 40 Foldseek-only sequences. The decomposition closes with 6,253 + 3,833 + 40 = 10,126. The expected per-query cost is therefore dominated by step 1 in the common case and bounded above by step 4 in the worst case.

Status and future development

BioSentinel as described here is a completed proof-of-concept. The curated dataset, the homology-aware splits, the fine-tuned classifier, the foldtuned-mimetic benchmark, and the cascade architecture are all in place, and the empirical question, namely whether classical homology screening misses AI-designed mimetics, and whether PLM-based and structural-alignment cascades can close the gap, has been answered quantitatively.

3.10 Lessons for TRILL users

Several conclusions emerge from this chapter. The first is that embeddings encode functional signal beyond ancestral homology, but the strength of that signal varies with how the target class is defined. For ancestrally coherent classes such as VP1, high-dimensional PLM embeddings yield outstanding classifiers; for convergently evolved functional classes such as CPPs, the embeddings remain informative but the performance ceiling is lower and the dominant variance is no longer captured by simple anomaly detection. Closely related is the observation that embedding-space dimensionality is dramatically overprovisioned for most classification tasks. PCA compression to 100–700 dimensions is essentially lossless for tasks that achieve high F1 in the first place, which matters for downstream infrastructure that can be designed around the smaller representation.

Family-based generation, by contrast, is sensitive to the fine-tuning schedule. The first epoch of fine-tuning a generative PLM yields most of the available gain, and subsequent epochs frequently degrade downstream metrics, presumably through memorization of training-set features that the classifier subsequently scores against. This chapter has also leaned heavily on cross-classifier and cross-method comparisons, which are best understood as tests of model agreement rather than checks against

ground truth, though they remain informative even with that caveat. ECPICK serves as an orthogonal check on the cellulase, antimicrobial, and threat classifiers (Sections 3.4 and 3.8). Agreement between two methods corroborates both, and disagreement can be informative in both directions.

Chapter 4

THE COMMUNITY-SCALE STRUCTURAL PROTEOME OF HCOM2

4.1 Introduction

Defined synthetic gut microbiomes such as hCom2 make microbiome–host mechanistic investigations tractable, but most of the protein-coding content of these consortia remains functionally uncharacterised at the sequence level. In this chapter I describe a community-scale predicted structural proteome of hCom2. I folded 99% of the proteomes of all 120 hCom2 strains with ESMFold (orchestrated through TRILL, the platform developed in Chapter 2), relaxed every structure under the AMBER ff14SB force field, ran structural quality-analysis, segmented every relaxed structure into domains, and assigned a CATH-S40 designation to each domain by structural search. The deduplicated table contains 802,415 hCom2 Merizo domains across the 120 strains, with a 15-strain soil synthetic-community comparator (138,482 Merizo domains) processed end to end through the identical pipeline as a bacterial-baseline outgroup. I overlay this fold-level call with EC (Enzymatic Commission), GO (Gene Ontology), KEGG (Kyoto Encyclopedia of Genes and Genomes), and COG (Clusters of Orthologous Genes) functional layers and report community-level differential signals between hCom2 and the soil panel. A complementary trait-level functional pipeline using Foldseek-based structural homology against the EcoFoldDB v2.0 curated reference set converges on the same qualitative contrast (soil aromatic / aerobic / secondary-metabolism-active, hCom2 plant-glycan / anaerobic / fermentative-central-metabolism dominated) and surfaces a specific gut-side enrichment for dissimilatory nitrate reduction to ammonium (DNRA via nrfA, 23 of 120 hCom2 genomes versus 0 of 15 soil) that the broader fold-level pipeline does not isolate. To demonstrate practical use of the resource I track the *H. pylori* immunomodulator TIPalpha (CATH 3.10.129.140) in 19 hCom2 proteins across 14 strains and two phyla, at DALI Z scores from 4.5 to 8.5 and DALI-aligned sequence identity 4 to 16 percent, while no sequence-only search returns a TIPalpha-family hit in the consortium. These are bona-fide TIPalpha CATH-fold structural homologs in commensal gut microbes at the fold level, recovered where sequence-only annotation finds nothing. At the family level by sequence, the result is best read as a population-level structural snapshot of an established multi-phylum Pfam family (PF02169 Lpp20 lipoprotein,

already documented across *Bacteroidota* and *Bacillota*) inside the hCom2 consortium rather than a discovery of a new phylum membership for the TIPalpha family proper (PF16753). Four of the 19 are PF02169-by-sequence and none are PF16753-by-sequence in the public catalogues at the time of analysis. Where each carrier sits between the TIPalpha and Lpp20 sub-families within the SHS2 Pfam clan resists a clean computational verdict. Beyond the worked example, the hCom2 structural proteome is offered as a community resource for functional and ecological inquiry into a tractable model microbiome, with strain-resolved structures that can be added or removed from the consortium in a controlled in-vivo experiment.

4.2 Why fold-level analysis of a defined consortium

The functional dark matter of the gut microbiome

The human gut microbiome harbours an enormous reservoir of protein-coding genes whose functions remain poorly characterised. A meta-analysis of 3,655 oral and gut metagenomic samples found 45,666,334 non-redundant genes at the 95 percent identity threshold, with roughly half of the catalogue arising as singletons restricted to a single sample (Tierney et al., 2019). This scale of genetic heterogeneity is the central practical problem in mechanistic microbiome biology, because most microbe–host interactions of interest cannot be cleanly attributed to specific strains, specific genes, or specific molecular activities when the underlying community is an open ecosystem.

In response to the overwhelming complexity of natural gut microbiomes, Cheng and colleagues built hCom2 by an iterative engraftment-and-augmentation procedure that added strains capable of stably colonising gnotobiotic mice after challenge with a human faecal sample and pathogenic *E. coli* (Cheng et al., 2022). Mice carrying either hCom2 or a human fecal community exhibit similar phenotypes, which makes hCom2 a tractable platform for mechanistic study of gene and strain-level contributions to microbiome-associated states (Cheng et al., 2022). The deployed consortium analysed here contains 120 strains.

What hCom2 provides, and what motivates the work described in this chapter, is the ability to attribute a phenotype to a specific gene in a specific strain in a controlled in-vivo experiment. The community can be reconstituted in gnotobiotic mice, members can be added or removed, and the resulting phenotypic shift can be traced back to molecular content. The bottleneck for this kind of mechanistic biology is often the annotation side. If most of the gene content of hCom2 is functionally

uncharacterised, then many of the gene-level hypotheses one could test are dead on arrival. This is the gap the work in this chapter is designed to narrow.

Structures can encode evolutionary history

Sequence-based annotation alone cannot resolve the function of much of this gene content. Pairwise alignment becomes unreliable in the twilight band of 20 to 35 percent identity, where false positives dominate, and the original characterisation of this band found that fewer than 5 percent of detected pairs adopted similar structures (Rost, 1999). Protein structure, by contrast, is far more robust to mutation. Across pairs of homologous domains, structural divergence accumulates at roughly one third to one tenth the rate of sequence divergence in the superposable core (Illergård, Ardell, and Elofsson, 2009). This conservation differential is what allows structure to expose relationships that sequence alone misses. Bacterial virulence factors furnish a striking biological example, because crystal structures of several effectors expose molecular mimicry of host activities in cases where the underlying sequences carry no detectable similarity to those host factors (Stebbins and Galán, 2001). Recent metagenomic work makes the same point at scale, identifying 106,198 novel metagenomic protein families with no similarity to reference genomes or Pfam, where the authors used gene-neighbourhood and structural-similarity evidence to begin annotating their unidentified sequences (Pavlopoulos et al., 2023).

The structural-proteomics era has now made such structure-driven annotation tractable for entire communities. AlphaFold2 achieved a median backbone accuracy of 0.96 angstrom on CASP14 domains, roughly threefold tighter than the next best method's 2.8 angstrom (Jumper et al., 2021). ESMFold, a language-model-based predictor that infers atomic-level structure directly from primary sequence, then made it possible to fold metagenomic catalogues by accelerating prediction by one to two orders of magnitude, enabling the ESM Metagenomic Atlas of more than 617 million predicted structures (Lin et al., 2023). Head-to-head benchmarking on monomers released after the 2022 training cutoff places AlphaFold at 88 percent accuracy and ESMFold at 76 percent, so the two methods occupy complementary niches between coverage and per-protein accuracy (Mahtha, Venkadesan, and Mohanty, 2026).

Several large-scale efforts have already exploited these predictions for fold-space cartography. Foldseek-cluster on the AlphaFold Database recovered 2.30 million non-singleton structural clusters, with 31 percent of clusters lacking annotation but covering only 4 percent of all proteins (Barrio-Hernandez et al., 2023). CATH-Assign

on roughly 370,000 confident AF2 models from 21 model organisms assigned 92 percent of domains to existing CATH superfamilies, recovered 2,367 putative novel superfamilies, and expanded CATH domain counts by 67 percent (Bordin et al., 2023). ECOD classification via DPAM2 across 48 AFDB proteomes catalogued 746,349 domains, attaching evolutionary classifications to a comparable scale of predicted structures (Schaeffer et al., 2024). Structure-based methods are not a panacea, however. A recent benchmark on the human phylome found that current structure-based phylogenomics does not outperform sequence-based methods for large-scale tree reconstruction, but recovers complementary candidate homologs that sequence alignment misses (Mutti, Ocaña-Pallarès, and Gabaldón, 2025).

What this chapter adds

Here I apply this toolkit to hCom2 itself. First, the hCom2 structural database is offered as a community resource for functional and ecological investigation of a tractable model microbiome, with strain-resolved predicted structures attached to species that can be added to or removed from a gnotobiotic experiment in a controlled way. Second, the parallel HMM-side and structure-side annotations show where structure complements sequence-based annotation across the synthetic gut consortium, alongside a 15-strain soil syncom processed through the same pipeline as a bacterial-outgroup comparator. Third, I run a complementary trait-level structural-homology pipeline against the curated EcoFoldDB v2.0 reference set across the same hCom2 and soil structures, which converges on the same qualitative community contrast as the fold-level CATH analysis and surfaces specific trait-level signals that the broader pipeline does not isolate. Fourth, I use structural search as a discovery instrument and walk one query through to family-level resolution, choosing as the query the *H. pylori* virulence factor TIPalpha for reasons described next.

Helicobacter pylori is a long-recognised risk factor for gastric adenocarcinoma, with strain-level differences in disease risk traced to the *cag* pathogenicity island and the *vacA* gene (Peek and Blaser, 2002). The *H. pylori* protein TIPalpha drives proinflammatory and tumor-promoting host responses by binding cell-surface nucleolin on gastric epithelial cells, internalising, and activating NF-κB to induce TNF-α and chemokine expression, with intermolecular Cys25–Cys25 and Cys27–Cys27 disulfide bridges at the N-terminus locking the secreted homodimer into the active form (Suganuma et al., 2008; Watanabe et al., 2010; Suganuma et al., 2021). Its closest structural neighbour Lpp20 (HP1456) shares the topology at Cα

root-mean-square deviation 3.3 angstrom by DALI superposition (Vallese et al., 2017). In CATH the TIPalpha topology is fold 3.10.129.140, named verbatim "*Helicobacter* TNF-alpha-Inducing Protein". I identify 19 hCom2 proteins across 14 strains and two phyla that map to this fold (18 *Bacteroidota* plus one *Bacillota*), confirm each by direct DALI superposition against PDB 3GIO, and find no hits at the same fold in the soil comparator. To resolve where each carrier sits within the SHS2 Pfam clan, I then re-align each pairwise against TIPalpha, Lpp20, the sibling-fold Hotdog Thioesterase, and a within-clan dodecin reference, and report the per-carrier verdict and a clan-wide robustness check across all ten Pfam CL0319 members. The structural recovery of CATH 3.10.129.140 in surface-exposed gut commensal lipoproteins is consistent with a general expectation that such proteins participate in host immunomodulation, whether protective or pathogenic, although the case does not by structure alone license any TIPalpha-specific virulence inference for the carriers.

4.3 Folding the consortium and assessing structural quality

The structure prediction step

The hCom2 consortium analysed here comprises 120 strains. As a comparator community processed end to end through the identical structure-prediction and annotation pipeline I ran the 15 strains from a soil synthetic community (Coker et al., 2022). The soil consortium is not phylogenetically matched to the human-gut hCom2 panel, and I do not run phylogenetic controls between the two communities, so the term *comparator community* is used in place of *negative control* to reflect that the soil panel measures community-level differential signal rather than a pipeline noise floor. The reference soil-syncom paper uses 18 bacterial strains, but only 15 of those had sequenced genomes available at the time of this work.

Per-protein monomeric structures were predicted with ESMFold via the TRILL `fold` command (Chapter 2, Section 2.8), which infers atomic-level structure directly from primary amino-acid sequence (Lin et al., 2023; Martinez, Murray, and Thomson, 2023). For hCom2, I used four Nvidia H100 GPUs for roughly one month of real-time compute. 72 of the hCom2 proteins were too large for the H100 vRAM to hold in memory. These were folded on CPUs at the cost of substantially longer per-protein runtime by my collaborator Joseph Boktor. ESMFold returned a predicted structure for 400,659 of the 400,685 hCom2 proteins, covering over 99.99% of the proteomes. The 15-strain soil syncom processed through the same pipeline yielded 87,635 total open reading frames of which 85,961 were folded by ESMFold.

Each predicted structure was then relaxed under the Amber ff14SB protein force field (Maier et al., 2015), run inside OpenMM 8 (Eastman et al., 2024). Structures were placed in implicit solvent, hydrogens and clashes were resolved, and the geometry was relaxed to a local energy minimum to remove steric strain inherited from the ESMFold output without perturbing the predicted fold. This relaxation step is consequential for the quality-control discussion below, since it removes most of the clash and rotamer pathologies that would otherwise dominate MolProbity scoring, so the downstream MolProbity pass rates report on a coordinate set that has already been pulled to a local minimum of a physics-based potential rather than on the raw ESMFold output.

Structural quality analysis

I assess the relaxed structures along four scoring axes, two from ESMFold and two from orthogonal validators. The ESMFold-derived metrics are pTM (a confidence estimate of global topological accuracy) and mean pLDDT (a per-residue confidence aggregated over the chain). The post-relaxation structural-evaluators are MolProbity, which returns a composite score that weights clash, Ramachandran, and rotamer terms and re-expresses the result on the experimental-resolution scale (Williams et al., 2018), and Z-DOPE, a statistical-potential score over the relaxed coordinates whose magnitude tracks deviation from native packing, with the published convention that scores below zero flag near-native models and the most accurate models cluster well below -1.5 (Eramian et al., 2008).

Across the hCom2 proteome (Table 4.1), mean pTM is 0.760 (SD 0.209) with 70.0 percent of structures passing the canonical 0.7 cutoff, mean pLDDT is 0.764 (SD 0.119) with 78.3 percent passing 0.7, mean MolProbity is 1.565 (SD 0.273) with 99.8 percent below 2.5, and mean Z-DOPE is -0.976 (SD 0.823) with 87.7 percent below zero. Figure 4.1 shows empirical cumulative distributions of all four metrics with the cutoff lines drawn, so the shape of each tail is visible alongside the headline percentages.

It is tempting to summarise prediction quality by a single number, and the field's instinct is usually model-derived confidences such as pTM or pLDDT. These metrics are correlated with experimental accuracy but not identical to it. MolProbity and Z-DOPE are external, post-relaxation validators with explicit physical interpretations. MolProbity is dominated by terms (Ramachandran outliers, rotamer outliers, all-atom clashes) that the AMBER ff14SB relaxation is specifically designed to eliminate, so

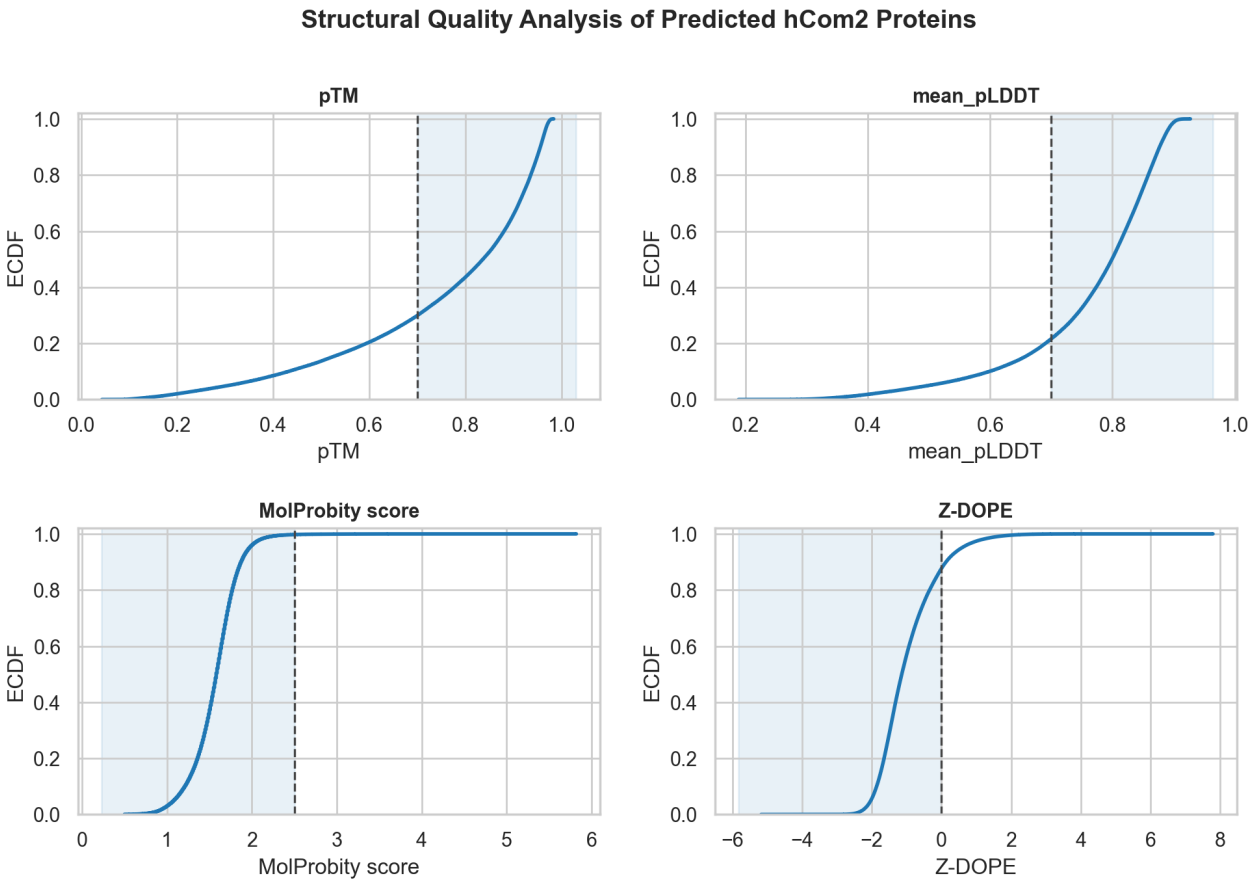


Figure 4.1: Quality-control profile of the relaxed hCom2 ESMFold proteome across four scoring axes. Empirical cumulative distributions are shown for the predictor-head metrics pTM (mean 0.760, SD 0.209) and mean pLDDT (mean 0.764, SD 0.119) and for the orthogonal validators MolProbity (mean 1.565, SD 0.273) and Z-DOPE (mean −0.976, SD 0.823) computed on the relaxed coordinates. Dashed vertical lines mark the canonical cutoffs (pTM and pLDDT at 0.7 by greater-than-or-equal-to, MolProbity below 2.5, Z-DOPE below 0). Pass rates are 70.0 percent at pTM, 78.3 percent at mean pLDDT, 99.8 percent at MolProbity, and 87.7 percent at Z-DOPE on 400,468 structures with all four metrics non-null.

Table 4.1: Quality-control headline numbers across the 400,469 hCom2 ESMFold structures after AMBER ff14SB relaxation in OpenMM 8.

Metric	Mean	SD	Heuristic cutoff	Percent passing
pTM	0.760	0.209	≥ 0.7	70.0
Mean pLDDT	0.764	0.119	≥ 0.7	78.3
MolProbity score	1.565	0.273	< 2.5	99.8
Z-DOPE	−0.976	0.823	< 0	87.7

the 99.8 percent MolProbity pass rate at score below 2.5 should be read as a property of the relaxed coordinates rather than as an independent validation of fold accuracy. Z-DOPE scores native-like packing of the relaxed coordinates rather than terms the relaxation explicitly targets, and at the stricter Z-DOPE threshold of < -1.5 (the regime of the most accurate models) only 28.7 percent of hCom2 structures pass. The two validators therefore stratify the proteome at very different levels, where roughly everything passes MolProbity, roughly the majority passes default Z-DOPE, and roughly a quarter passes strict Z-DOPE.

Generalizability against ground truth. Out of the hCom2 proteome, 128 proteins have an experimentally resolved counterpart in the PDB, of which 5 fall outside the ESMFold training set. The seen and unseen subsets return comparable mean TM-score (0.924 against 0.937) at $n = 5$ unseen proteins. I do not draw a Generalizability claim from this five-protein subset, and I instead inherit the population-scale statement from the post-2022 PDB benchmark cited above (Mahtha, Venkadesan, and Mohanty, 2026), because the five-protein hCom2-unseen subset is not powered to support a Generalizability claim on its own.

4.4 From structures to a community-scale fold-and-function portrait

Domain segmentation and structural CATH assignment

Each relaxed protein was decomposed into structural domains with Merizo, a deep-learning segmentation method built on invariant point attention (Lau, Kandathil, and Jones, 2023). The domain-level call uses FoldClass, an embedding-based structure comparison method built on a stack of E(n)-equivariant graph neural network blocks that take a domain's alpha-carbon coordinates as input and outputs a fixed-size vector representing the overall fold. The network is trained as a multiclass classifier against the class, architecture, and topology levels of CATH, after which the averaged node embeddings are used directly as fold representations and compared via cosine similarity, with the top candidates verified using TM-align. Merizo-Search combines the two, and I ran `merizo.py easy-search -k 100 -iterate` against the CATH-S40 reference set and filtered to retain the highest TM-align score per Merizo domain. I use the CATH v4.4 hierarchy (Class, Architecture, Topology, Homologous superfamily) throughout (Waman et al., 2025).

Merizo segmentation of the relaxed hCom2 proteome returns 802,415 domains across the 120 strains, and the same pipeline applied to the 15-strain soil syncom returns 138,482 domains. After deduplicating each Merizo domain to a single best

CATH-S40 hit (preferring an assigned CATH call over UNASSIGNED, then breaking ties by maximum TM-align score), this 802,415-row hCom2 table is the backbone of every domain-level analysis below. The empirical cumulative distribution of Merizo domains per kilobase of coding sequence across hCom2 and soil strains shows that domain density is comparable across the two communities once normalised by genome size, indicating that the larger hCom2 domain count reflects panel size rather than per-genome richness.

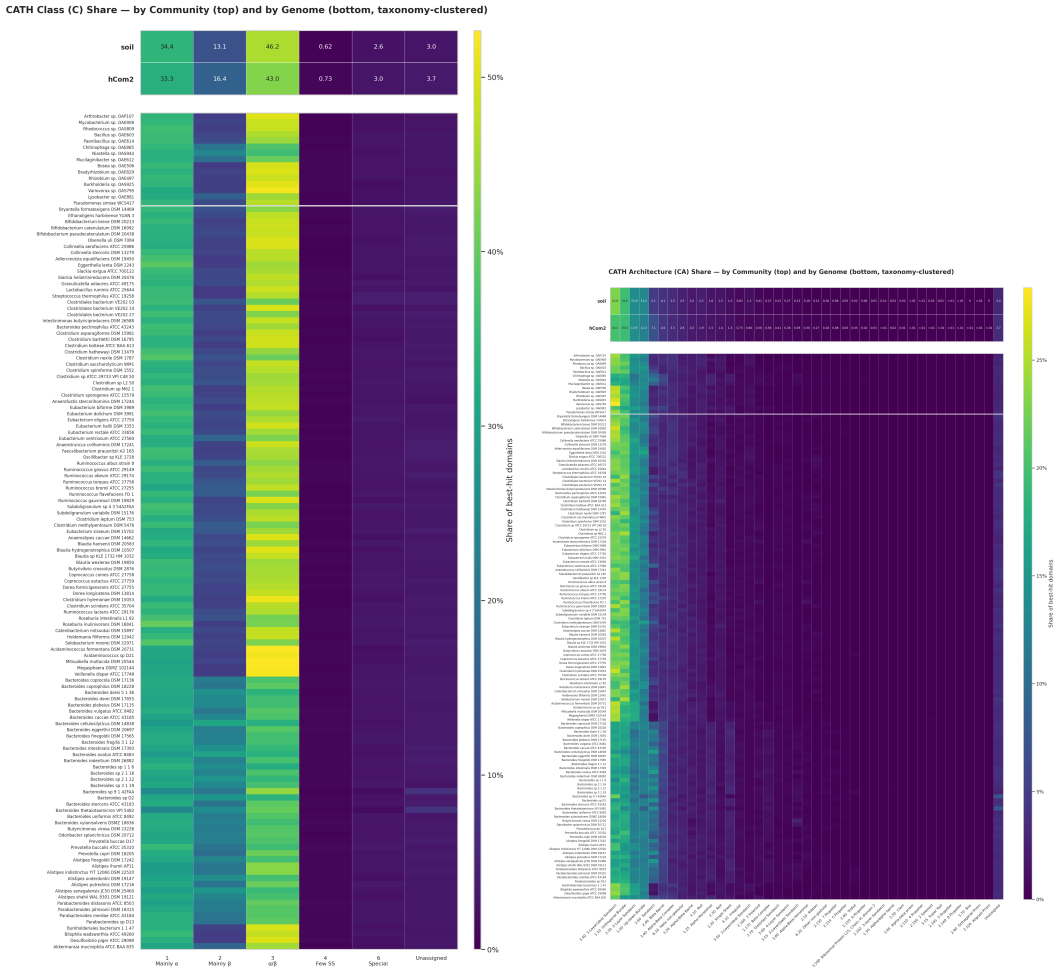
The C-level (Class) split across hCom2 strains is dominated by the alpha-beta class (C = 3) and mainly-alpha class (C = 1), with mainly-beta (C = 2) third. Per-genome heatmaps of class share and architecture share (Figure 4.2) show that the main axis of variation between strains tracks phylum, with *Bacteroidota* and *Bacillota* strains separating cleanly on the architecture axis. The phylum-level signal is unsurprising, because architecture-level fold preferences are partly determined by the metabolic and cellular lifestyle of the lineage, but it confirms that the Merizo+FoldClass call is recovering biology rather than producing noise.

Community-exclusive folds: a sampling-depth caveat

I then asked which CATH codes appear exclusively in one community. Raw counts of community-exclusive CAT and CATH codes (Figure 4.3a) show a numerically larger hCom2-exclusive bucket, but this advantage substantially shrinks when each community-exclusive code is required to also appear in at least one strain of the panel and counts are normalised per strain (Figure 4.3b). The shrinkage is expected, because hCom2 carries 120 strains versus 15 for soil, and apparent exclusivity is partly a reflection of sampling depth. I therefore treat the per-strain panel as the more conservative estimate of community-distinguishing folds. This is a worked example of a more general point that recurs throughout the chapter. When one community is sampled an order of magnitude more deeply than another, raw counts of community-exclusive features systematically overstate the strength of the differential signal, and the rate panels in the next subsection are constructed to push back against this kind of inflation.

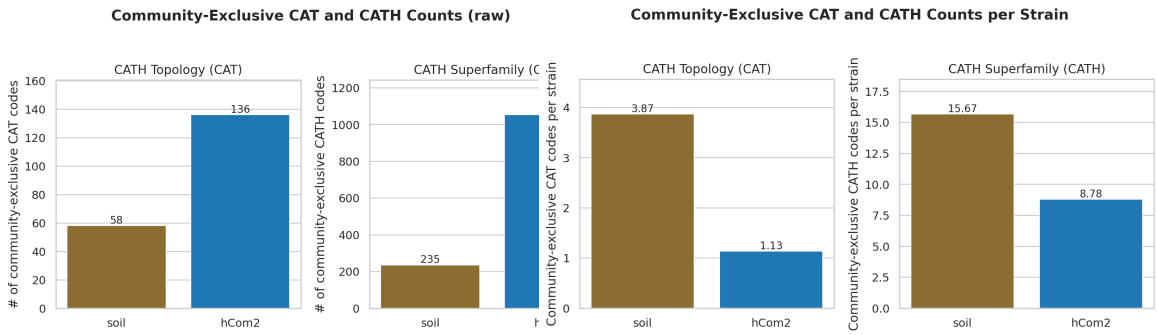
Differential fold-prevalence between hCom2 and the soil panel

A per-strain density test by Mann–Whitney U on per-1000 domain rates (minimum-of-three carrier-genomes filter) returned 4,051 testable CATH designations, of which 855 cleared $q < 0.05$ by Benjamini–Hochberg correction (Benjamini and Hochberg, 1995). The top-20 hCom2-leaning superfamilies are shown in Figure 4.4.



(a) CATH Class (C-level) share (b) CATH Architecture (CA-level) share

Figure 4.2: Per-genome CATH share at two levels of the CATH hierarchy. (a) C-level (Class) share across the four CATH classes plus an Unassigned bucket. (b) CA-level (Architecture) share across the ∼40 architectures populated by the joint hCom2–soil panel. In each subfigure, the top two rows are the community-level aggregates (soil, hCom2) and the bottom block is the per-genome breakdown clustered taxonomically, with rows labelled by strain. Cell shading encodes per-genome share of Merizo domains in that CATH category. The architecture panel (b) most clearly resolves the phylum-level fold-preference axis, with *Bacteroidota* and *Bacillota* strains separating into distinct architecture-share regimes.



(a) Raw counts (b) Per-strain normalised

Figure 4.3: Community-exclusive CATH codes between hCom2 and the soil syncom. (a) Raw counts at the CAT (Topology) and full CATH (Homologous Superfamily) levels. A code is counted exclusive when it appears in one community and is absent from the other. The hCom2 bucket is numerically larger at both levels, but the comparison is unweighted with respect to panel size and therefore conflates true community specificity with the difference in strain count between the two consortia (120 against 15). (b) The same exclusivity test after normalisation by panel size, expressed per strain (totals divided by 120 for hCom2 and 15 for soil).

The hCom2-leaning panel reads as the obligate-anaerobic gut consortium one would predict from the underlying *Bacteroidota*-dominated composition. Two biologically interpretable themes occupy five of its twenty rows. Four rows carry the mobile-genetic-element load characteristic of the *Bacteroidales* conjugative mobilome that distributes antimicrobial resistance and accessory loci between strains, namely the Tyrosine recombinase N-terminal domain (CATH 1.10.150.130) at rank 7 (per-1000 mean 3.21 in hCom2 and 0.99 in soil, $q = 6.3 \times 10^{-6}$), the Integrase catalytic core (1.10.443.10) at rank 8 (2.80 versus 0.70, $q = 1.2 \times 10^{-6}$), the Resolvase N-terminal catalytic domain (3.40.50.1390) at rank 13 (1.54 versus 0.38, $q = 3.6 \times 10^{-4}$), and the HUH-endonuclease relaxase fold (3.30.930.30, named in the CATH-S40 representative library after the MobM MOBV-family relaxase) at rank 14 (1.26 versus 0.10, $q = 2.6 \times 10^{-5}$) (Grindley, Whiteson, and Rice, 2006; Francia et al., 2004; Smillie et al., 2011). One row contributes a *Bacillota*-specific surface signature within the mixed-phylum hCom2 consortium, namely the choline-binding repeat fold (2.10.270.10) at rank 16 (per-1000 mean 1.14 in hCom2 and 0.00 in soil, $q = 2.4 \times 10^{-2}$), the modular cell-wall anchor of *Bacillota* surface proteins in *Streptococcus* and related lineages with choline-decorated teichoic acids (Fernández-Tornero et al., 2001; Maestro and Sanz, 2016).

The Immunoglobulin fold (2.60.40.10) at rank 2 of the same panel (per-1000 mean 21.53 in hCom2 and 12.85 in soil, $q = 1.8 \times 10^{-2}$) and other broad housekeeping

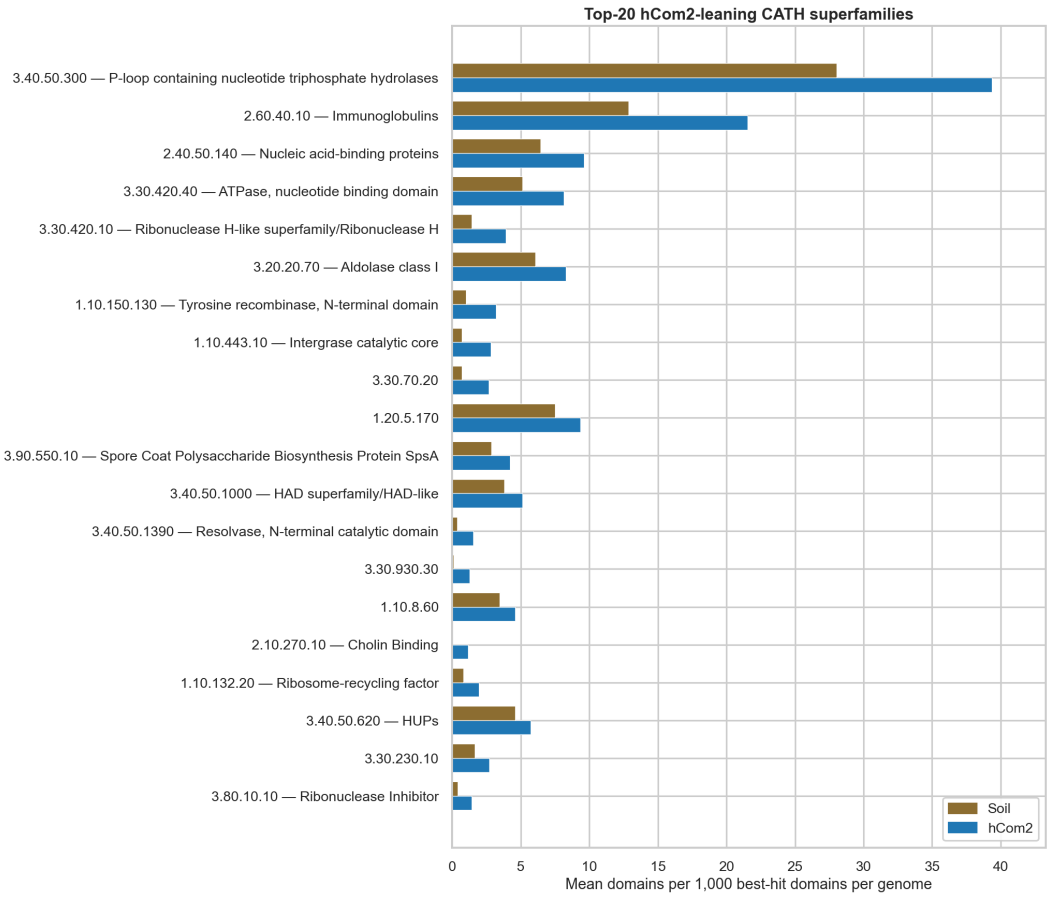


Figure 4.4: Top-20 hCom2-leaning CATH superfamilies by Mann–Whitney U on per-1000 domain rates per genome, after a minimum-of-three carrier filter (4,051 testable CATH codes, 855 significant at $q < 0.05$ by Benjamini–Hochberg false discovery rate correction). Bars indicate per-genome mean per-1000 rates. Each row carries its CATH code and the FoldClass annotation string.

folds enter the top-20 panel largely through compositional displacement, because hCom2's smaller mean genome size raises the per-1000 fractional share of generic carrier domains (Bodelón, Palomino, and Fernández, 2013). This compositional-displacement effect, which I return to at the COG layer below, is the most important interpretive caveat for any per-1000-style rate panel built on communities with substantially different mean genome sizes.

Below the top-20 panel cutoff but at $q < 0.05$ in the same per-1000 layer, four further folds anchor classical gut-lifestyle chemistry that the ranking by absolute per-1000 rate difference does not surface. Pyruvate-ferredoxin oxidoreductase (CATH 3.40.920.10, per-1000 mean 0.278 in hCom2 and 0.041 in soil, $q = 5.4 \times 10^{-4}$, carried by 91 of 120 hCom2 strains and 3 of 15 soil strains) is the thiamine-pyrophosphate

plus iron-sulfur unit that obligate anaerobes use in place of the oxygen-dependent pyruvate dehydrogenase complex, also serving as the entry to acetate fermentation and reductive carboxylation routes used by acetogens and so linking directly to formyltetrahydrofolate synthetase below (Charon et al., 1999). The Ntn-hydrolase fold of bile-salt hydrolase (3.60.60.10, per-1000 mean 0.245 in hCom2 and 0.027 in soil, $q = 6.3 \times 10^{-4}$) deconjugates glyco- and tauro-bile acids, contributing to a microbial bile-acid pool that also includes amine-conjugated species and feeds host FXR signalling (Jones et al., 2008; Quinn et al., 2020; Rimal et al., 2024). The FeoB ferrous-iron transport GTPase (2.30.30.90, per-1000 mean 0.511 in hCom2 and 0.136 in soil, $q = 6.3 \times 10^{-4}$) is the inner-membrane Fe^{2+} permease tuned to the oxygen-limited large-intestinal lumen where ferrous iron dominates the bioavailable pool. *feoB* loss attenuates gut colonisation in *Salmonella* Typhimurium, *Campylobacter jejuni*, and *Clostridioides difficile* (Lau, Krewulak, and Vogel, 2016; Naikare et al., 2006; Costa et al., 2016; Deshpande et al., 2022). Formyltetrahydrofolate synthetase (3.10.410.10, per-1000 mean 0.183 in hCom2 and 0.037 in soil, $q = 9.9 \times 10^{-5}$, carried by 98 of 120 hCom2 strains and 4 of 15 soil strains) is a key enzyme of the Wood–Ljungdahl methyl branch by which gut acetogens (*Blautia*, *Eubacterium*, and *Ruminococcus* in the *Lachnospiraceae*) recycle H_2 and CO_2 into acetate, the dominant short-chain fatty acid of the colon, doubles as the standard phylogenetic marker for surveying homoacetogen diversity in gut communities, and is also reused by gut anaerobes for folate-cycle one-carbon transfers (Ragsdale and Pierce, 2008; Ottesen and Leadbetter, 2011).

These four folds together form the chemistry I would have hoped a fold-level pipeline would surface in a hCom2-vs-soil contrast, namely oxygen-independent central metabolism, bile-acid deconjugation, anaerobic iron acquisition, and the Wood–Ljungdahl acetogenic axis. That they appear at FDR-controlled $q < 0.05$ when not visible in the top-20 absolute-rate ranking is the most concrete sanity check available for the pipeline, where the chemistry passes the differential-prevalence test even when it does not dominate the per-1000 budget.

The soil panel reads as an aerobic, secondary-metabolite-active community. I will not reproduce the soil-leaning top-20 here, but the same statistical test applied in the opposite direction recovers the aerobic, free-living, secondary-metabolite-active lifestyle one would predict from a soil panel dominated by *Pseudomonadota* and *Actinomycetota*. NAD(P)-binding Rossmann domains (CATH 3.40.50.720) and FAD/NAD(P)-binding domains lead the panel at per-1000 mean 29.47 versus 18.10

and 8.85 versus 4.59 respectively, both at $q < 10^{-4}$, carrying the aerobic respiratory cofactor pool and the flavoprotein disulfide reductase family of redox enzymes (Lesk, 1995; Argyrou and Blanchard, 2004). Cytochrome c-like fold (1.10.760.10) appears at per-1000 mean 1.88 in soil versus 0.16 in hCom2 ($q = 2.4 \times 10^{-7}$), and the soil-leaning direction is the one would predict from the lower load of canonical aa_3- and bo_3-type heme-copper terminal oxidases in obligate gut anaerobes relative to free-living soil aerobes (Allen et al., 2003; Winstedt and Von Wachenfeldt, 2000), though I cannot from CATH calls alone distinguish c-type cytochromes in canonical aerobic respiratory chains from c-type cytochromes in anaerobic electron-transfer pathways, and some *Bacteroidota* including *B. fragilis* are known to carry NQR-family NADH oxidoreductases and cytochrome bd-type oxidases that support partial respiration at nanomolar O_2 (Ito et al., 2020). A complete fatty-acid β-oxidation chain spans four further soil-leaning rows (Butyryl-CoA Dehydrogenase domains, Acyl-CoA dehydrogenase/oxidase, 2-enoyl-CoA Hydratase), reflecting the long-chain fatty-acid catabolic pathways that fuel aerobic chemoorganotrophic growth in soil and the documented mycolate and lipid catabolism of the *Mycobacterium* and *Rhodococcus* lineages in the soil panel (Fujita, Matsuoka, and Hirooka, 2007; Holder et al., 2011). Secondary-metabolite biosynthesis enters at rank 14 with the Thiolase/Chalcone synthase fold (3.40.47.10) at per-1000 mean 3.44 versus 1.28 ($q = 6.9 \times 10^{-7}$), paired at rank 6 with the TetR-family Tetracycline Repressor domain 2 fold (1.10.357.10) at 6.09 versus 2.27 ($q = 1.7 \times 10^{-4}$). These are the transcriptional-repressor and polyketide-initiation backbones of antibiotic-resistance and secondary-metabolism gene clusters across soil heterotrophs (Austin and Noel, 2003; Ramos et al., 2005; Santos-Aberturas and Vior, 2022; Novakova et al., 2022). Environmental sensing occupies three rows (response regulator receiver fold, PAS domain, GAF domain), all consistent with the dense two-component signal-transduction repertoire of free-living chemoheterotrophs in chemically heterogeneous habitats (Stock, Robinson, and Goudreau, 2000; Gao and Stock, 2009; Taylor and Zhulin, 1999; Ho, 2000; Mo et al., 2022).

Below this top-20 cutoff, the soil panel additionally surfaces a UV-resistance and aromatic-catabolism signature. The cryptochrome-photolyase FAD-binding barrel (CATH 1.25.40.80, per-1000 mean 0.000 in hCom2 and 0.109 in soil, $q = 6.2 \times 10^{-23}$) is the UV-photoreactivation signature and the most significant CATH in the entire dataset, paired with the DNA cyclobutane-dipyrimidine photolyase CPD-lid lobe (1.10.579.10) (Sancar, 2003; Portero et al., 2019). The luciferase-like-monooxygenase F_{420}-dependent TIM barrel (3.20.20.30) and cytochrome P450

(1.10.630.10) round out the aromatic-catabolism and secondary-metabolite-tailoring axis in soil actinobacteria (Jirapanjawat et al., 2016; Greule et al., 2018; Kelly and Kelly, 2013), with the NRPS condensation domain (3.30.559.30) closing the assembly-line axis (Bloudoff and Schmeing, 2017; Dror et al., 2020). None of these UV-resistome or NRPS folds exist in any meaningful share in hCom2, consistent with a dark colonic lumen and the absence of soil-style secondary-metabolite gene clusters in the gut consortium.

Functional-layer concordance

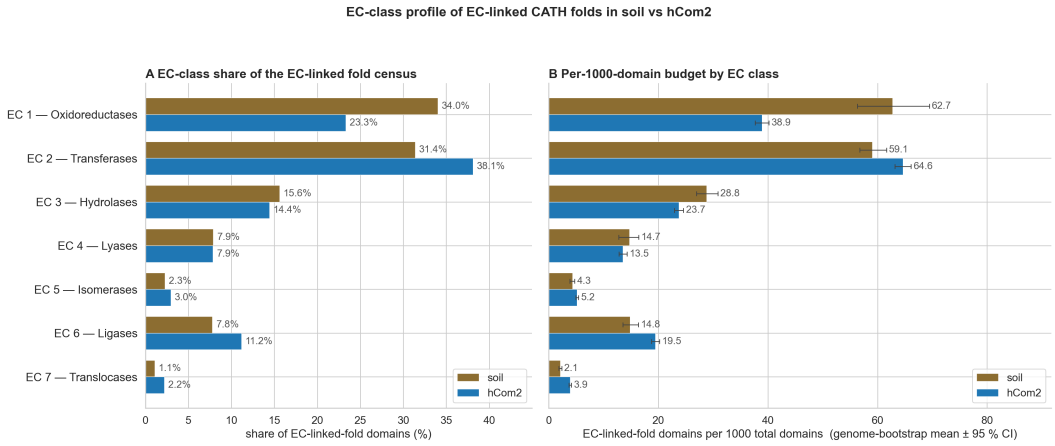


Figure 4.5: EC-class share at the per-1000 domain rate that retains 212 CATH codes at the SwissProt reviewed-only EC1 level. Bars compare the hCom2 and soil per-1000 rates across the seven top-level EC classes (oxidoreductases, transferases, hydrolases, lyases, isomerases, ligases, translocases).

To go from a fold-level call to functional context, every CATH superfamily that survived domain segmentation was attached to four functional layers (EC, GO, 1, COG) by a chained API pipeline.

The CATH-to-EC mapping. For every CATH code I queried the InterPro REST API for the CATH-Gene3D signature and any integrated InterPro entry. For each integrated entry I then queried InterPro for GO terms (Biological Process, Molecular Function, Cellular Component) and EC numbers, with a regex fallback over the entry name and description for partial four-component EC strings. Of the 4,947 CATH codes carried by hCom2 and soil, this pass populated EC numbers for 694 CATHs and GO terms for 452. Many CATH-Gene3D signatures lack a corresponding integrated InterPro entry. To recover these annotations I queried UniProt directly for entries cross-referencing the Gene3D signature of each EC- or GO-empty CATH

and aggregated EC numbers and GO IDs across the returned entries. This back-fill raised CATH-with-EC from 694 to 2,758 and CATH-with-GO from 452 to 4,532.

Catalytic-function claims rest on a tightened CATH-to-EC1 mapping that filters the merged annotation by three rules. I drop CATH families with exclusively TrEMBL support, keep a CATH only when its dominant EC1 is supported by at least τ fraction of the weighted reviewed UniProt members carrying that CATH (with the headline cutoff at $\tau = 0.75$), and assign each retained CATH to a single dominant EC1 rather than splitting fractionally across multiple EC1s. EC1 is the top-level enzymatic commission (oxidoreductase, transferase, hydrolase, lyase, isomerase, ligase, translocase) and the per-1000 EC1 rate panel of Figure 4.5 is computed against the resulting 212-CATH whitelist.

The CATH-level signal is reinforced rather than redirected at the functional-annotation layer. Across the 212-CATH-derived enzymatic commissions, oxidoreductases (EC 1) take 34.0 percent of the soil EC-linked-fold budget at a per-1000 rate of 62.74 (95 percent CI 56.78 to 69.52) versus 23.3 percent and 38.93 (95 percent CI 37.62 to 40.17) in hCom2, in line with the soil aromatic-catabolism cluster above. Transferases (EC 2) take a 38.1 percent share in hCom2 with a per-1000 ratio of 1.09 over soil, anchoring on the well-documented *Bacteroidota* investment in carbohydrate-active enzymes organised as polysaccharide utilisation loci for foreign-glycan deconstruction, alongside capsular- and exopolysaccharide-glycosyltransferase loci that build the surface polysaccharides of these same lineages (Grondin et al., 2017; Coyne et al., 2005). Translocases (EC 7) lean hCom2 with a per-1000 ratio of 1.87, consistent with the dense ABC-transporter and F-type / Na^+-translocating ATP-synthase repertoire of small fermentative anaerobes.

The COG layer and its compositional caveat. The COG-level differential view at COG-category granularity and at COG-implied KEGG-pathway granularity carries the same axis with cross-database support. At the category level, hCom2 is enriched in central-machinery and mobile-element categories. Replication, recombination, and repair (category L) is at per-1000 mean 39.32 in hCom2 and 29.25 in soil, Translation, ribosomal structure, and biogenesis (J) is at 64.95 and 59.69, Nucleotide transport and metabolism (F) is at 26.42 and 19.98, Cell cycle control (D) at 29.87 and 22.53, and Mobilome (X, covering prophages and transposons) at 5.67 and 2.63 (the largest positive fold change at $\log_2$ +1.11), reinforcing the mobile-genetic-element load reading. Soil is enriched in catabolic and biosynthetic categories.

Secondary metabolites biosynthesis, transport, and catabolism (Q) at per-1000 mean 26.22 in hCom2 and 38.46 in soil (the largest negative fold change at $\log_2$ -0.55), Lipid transport and metabolism (I) at 35.87 and 38.08, Inorganic ion transport and metabolism (P) at 47.22 and 49.39, Amino acid transport and metabolism (E) at 91.59 and 98.27, and Energy production and conversion (C) at 46.40 and 50.69, all in line with the aerobic-respiration and secondary-metabolism reading.

At the pathway level, the top hCom2-leaning pathways are translation, ribosome-assembly, and nucleotide-metabolism modules. Aminoacyl-tRNA synthetases lead at Δ per-1000 +12.67, followed by Ribosome 30S subunit (Δ +12.37), 23S rRNA modification (Δ +10.64), Translation factors (Δ +9.53), Purine biosynthesis (Δ +8.99), Ribosome 50S subunit (Δ +8.35), and Pyrimidine biosynthesis (Δ +7.21). Lipid A biosynthesis (Δ +6.41) tracks the gram-negative cell-wall investment of the *Bacteroidota* majority, and CRISPR-Cas system (Δ +5.66) tracks the defense load against the *Bacteroidales* mobilome. The top soil-leaning pathways are dominated by aerobic respiratory and biosynthetic chains. Fatty acid biosynthesis is the single largest contrast at Δ -53.38, followed by Menaquinone (Δ -42.72), Heme (Δ -33.70), Molybdopterin (Δ -33.61), Pyrimidine degradation (Δ -30.03), TCA cycle (Δ -24.01), and Ubiquinone biosynthesis (Δ -14.30).

The translation, ribosome-assembly, and nucleotide-metabolism enrichment in hCom2 is best read as compositional displacement driven by a smaller mean genome size in the gut consortium (3.81 Mb across 120 hCom2 strains versus 6.63 Mb across the 15 soil-syncom strains). This is consistent with the universal sub-linear scaling of translation and other housekeeping COG categories. The translation category has an effective scaling exponent near zero (fixed absolute count regardless of genome size), so its per-1000 fractional share scales as $1/\text{genome size}$, while super-linearly scaling categories (transcription factors, kinases, and the secondary-biosynthesis machinery that anchors much of the soil-leaning signal) compositionally displace housekeeping share in the larger soil-syncom genomes (Grilli et al., 2012). I report this as a structural property of the pipeline rather than as a confound, since the per-1000 panels are descriptive layers of community composition, and the fold-level Mann–Whitney panels carry the FDR-controlled signal.

The GO and KEGG layers inherit annotation-propagation biases from reference proteomes dominated by characterised eukaryotic and model-organism entries. Known propagation artifacts (vertebrate-only GO terms applied to bacterial folds, plant or insect KEGG maps drawn onto bacterial enzymes, mammalian-lipid ChEBI

entries attached to bacterial CYP and LOX folds, teichoic-acid ChEBI entries co-counted across multiple CATH codes) are flagged in the figure captions where they appear rather than algorithmically removed from the underlying tables, because at this propagation distance the boundary between labelling artifact and weak biological signal cannot be set by a single rule. The fold-level (CATH) layer is therefore the primary signal at the figure level, with EC, GO, KEGG, and COG layers reported as supporting context.

Structural CATH calls complement the HMM-side annotation

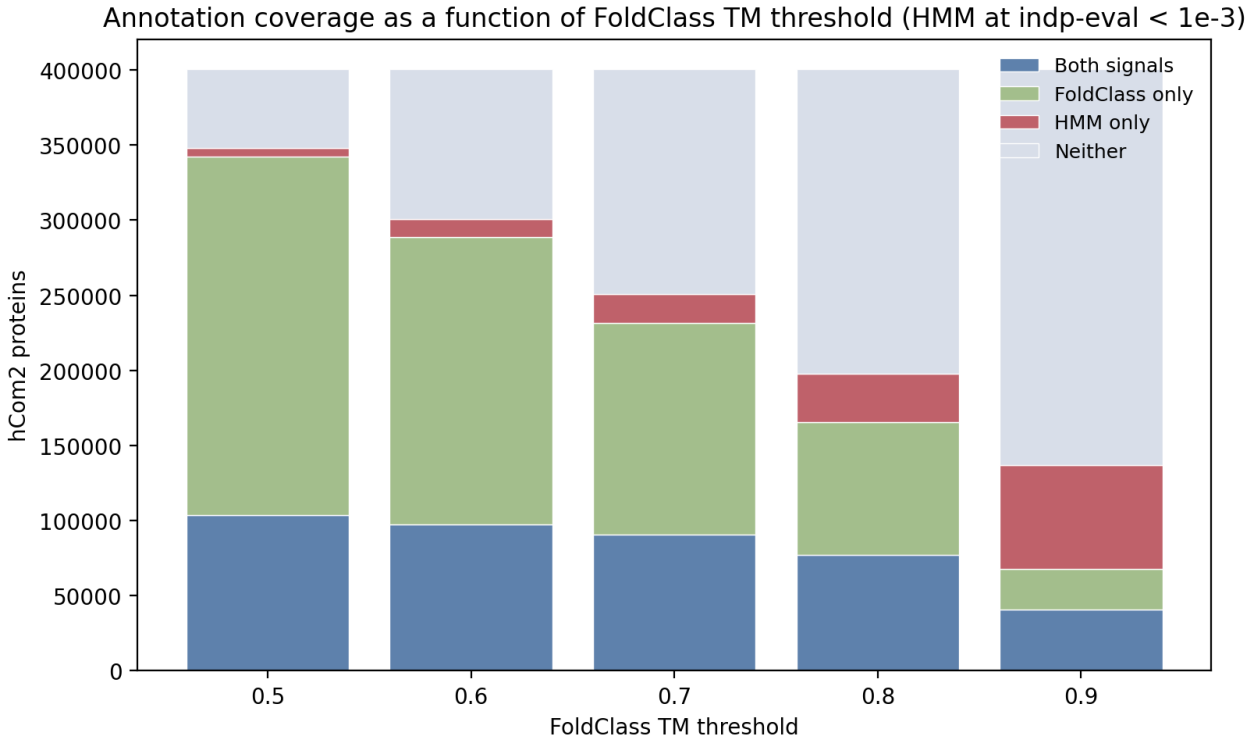


Figure 4.6: Stacked-bar visualisation of the four-bucket HMM-vs-structure coverage cross at FoldClass TM thresholds 0.5, 0.6, 0.7, 0.8, and 0.9, with the HMM side held fixed at cath-tools-genomescan default independent-evalue < 10^{-3}. Across the 400,469-protein hCom2 universe, FoldClass-only is the largest non-shared bucket at the canonical fold-level cutoff (TM at least 0.5) and shrinks monotonically as the threshold tightens.

Across these layers, the structural side of the pipeline does meaningful work that a sequence-only HMM annotation pass would not. In parallel with the structural pipeline, the same hCom2 protein sequences were annotated by cath-tools-genomescan, which runs the CATH Functional Family (FunFams) HMMs (Lewis, Sillitoe, and Lees, 2019), followed by cath-resolve-hits to identify the best non-

overlapping subset of domains per query protein (Lewis, Sillitoe, and Lees, 2019). This serves as the orthogonal sequence-only baseline against which the structural CATH pipeline (Merizo-Search) is compared.

Of the 400,469 hCom2 ESMFold structures in the master ID map, 238,617 (59.6 percent) carry a FoldClass CATH hit at TM at least 0.5 without a significant cath-tools-genomescan HMM call at the default independent e-value cutoff, while only 5,716 (1.4 percent) carry an HMM hit without a FoldClass call. The structure-only block shrinks monotonically as the FoldClass TM cutoff is tightened (191,313 of 400,469 proteins at TM ≥ 0.6, 141,181 at ≥ 0.7, 88,241 at ≥ 0.8, and 27,145 at ≥ 0.9), with the full sweep shown in Figure 4.6. The FoldClass-only Merizo-domain count at each threshold is dominated by alpha-beta folds (C = 3) and mainly-alpha folds (C = 1), consistent with structural detection picking up folds whose underlying sequences sit inside the twilight zone where alignment scoring is no longer reliable (Rost, 1999).

I frame this asymmetry as structure *complementing* rather than *outperforming* sequence-based annotation. The FoldClass-only set will include calls that an upstream HMM correctly declined to make at its default threshold, the two annotation routes use different reference databases and cutoffs, and a structural classification pipeline can confidently call a fold for a domain whose underlying sequence has drifted far from any HMM-defined family. The right way to read the structure-only block is as an additional layer of fold-level inference layered on top of, rather than replacing, the sequence-side HMM annotation that already exists. Section 4.4 applies the same complementary logic at a different annotation granularity by re-annotating the proteome through a structural-homology pipeline curated for ecological traits, and the TIPalpha case study in Section 4.5 then demonstrates the structure-search-as-discovery-instrument pattern in detail on a single fold.

Trait-level functional comparison via EcoFoldDB structural homology

The CATH-and-EC/KEGG/COG functional-layer pipeline above operates over the entire fold catalogue and reads the hCom2-vs-soil contrast through broad biochemistry classes. To read the same contrast through a complementary lens curated specifically for ecological traits, I ran a parallel structural-homology annotation against EcoFoldDB v2.0 (Ghaly et al., 2025), a Foldseek-searchable structural reference database covering eight ecologically relevant categories, namely carbon cycling, nitrogen cycling, sulphur cycling, phosphorus cycling, iron cycling, trace-gas oxidation, plant–microbe interactions, and osmotic-stress tolerance. The same 120

hCom2 and 15 soil-syncom ESMFold structures used throughout this chapter were fed directly into EcoFoldDB-annotate, with Foldseek converting each predicted structure to its 3Di alphabet internally. The ProstT5 sequence-to-3Di translation hop that EcoFoldDB-annotate uses by default was bypassed in favour of our pre-computed full-atom structures, so the structural representation is the same one used for every other analysis in this chapter. A hit is defined as a query protein with a curated EcoFoldDB structural homolog at e-value below 10^{-10} and at least 80 percent bidirectional coverage. The pipeline returned 6,834 structural-homology hits across the 135 genomes.

However, EcoFoldDB v2.0 was curated for soil and environmental traits. Many gut-relevant biochemistries (mucin O-glycan utilisation, bile-acid biotransformation, host-interaction toxins, complete polyamine and short-chain-fatty-acid biosynthesis pathways, vitamin synthesis) are not represented in the reference set. The absolute hits-per-1000-proteins gap of roughly two-fold in soil's favour at the headline level is therefore partly a database-scoping artifact rather than a community-level functional asymmetry. The qualitative trait-distribution comparison (which traits each community encodes, and at what prevalence) is much more defensible than the absolute rate.

At the category level(Figure 4.7), soil leads in every category except carbon cycling, where the hCom2 per-1000 rate is roughly half soil's despite the large *Bacteroidota* CAZyme load that drives the carbon-cycling signal in hCom2. A few patterns are apparent. First, niche environmental categories with no clear gut-physiological role show roughly six- to ten-fold soil enrichment per 1000 proteins, namely trace-gas oxidation (0.06 hCom2 versus 0.35 soil hits per 1000 proteins), iron cycling (0.04 versus 0.42), and osmotic-stress tolerance (0.08 versus 0.52). Gut chemistry has tightly host-regulated O_2 tension, iron availability, and osmolarity, so the asymmetry is biologically reasonable rather than a methodological surprise. Second, carbon cycling is roughly two-fold higher in soil per 1000 proteins, but the contrast is driven by aromatic and polyphenol pathways (lignin breakdown, polyphenol catabolism) rather than sugar metabolism, consistent with the soil-leaning EC1 oxidoreductase per-1000 rate already visible in Section 4.4. Third, the plant–microbe interactions category is roughly tied on rate (1.71 versus 1.75 hits per 1000 proteins) but differs sharply in content. hCom2's signal is dominated by polyamine biosynthesis, a basal anabolic pathway that is not plant-specific, while soil's spans ACC deaminase, IAA, siderophores, and GABA. At the per-genome level, trait load is roughly three-fold

higher in soil at the median (median 117 hits per genome in soil versus 32 in hCom2, with Mann–Whitney U $p < 10^{-7}$ on each of total hits, unique sub-categories, and hits per 1000 proteins).

A single EcoFoldDB hit is structural-homology evidence for one gene. The trait-level calls in Ghaly et al. require a minimum gene set per trait. For example, N_2 fixation requires nifHDK, anfHDK, or vnfHDK. ACC deaminase requires acdS. GABA production requires (gadA or gadB) and gadC. DNRA requires nrfA. Applying those rules verbatim to the per-genome hit table changes the headline reading on several traits and surfaces one finding worth flagging in its own right(Figure 4.8).

The hit-level signal of 68 nif-family hits in hCom2 collapses, after the rule (nifHDK or anfHDK or vnfHDK) is applied, to a single trait-positive hCom2 genome (*Slackia exigua* ATCC-700122) and zero soil-positive genomes. The vast majority of hCom2 nif-family hits are isolated nifH or vnfK without the full operon, consistent with prior reports that ruminant gut clostridia carry partial-but-not-complete nif suites (Kim et al., 2014).

The single trait-level enrichment that runs gut-positive against soil-negative is dissimilatory nitrate reduction to ammonium (DNRA), called via nrfA. 23 of 120 hCom2 genomes are nrfA-positive versus 0 of 15 soil. The carriers are 14 *Bacteroides* species (including *B. xylanisolvens*, *B. cellulosilyticus*, *B. thetaiotaomicron*, *B. ovatus*, *B. fragilis*, *B. uniformis*, *B. caccae*, *B. plebeius*, *B. intestinalis*, *B. rodentium*, plus four unnamed *Bacteroides* spp.), 2 *Parabacteroides*, 2 *Prevotella*, *Burkholderiales bacterium* 1-1-47, *Slackia heliotrinireducens*, *Desulfovibrio piger* and *Bilophila wadsworthia*. All 28 underlying nrfA structural hits match the *Wolinella succinogenes* reference (UniProt Q65RI4) at e-values 10^{-13} to 10^{-38} with both query and target coverage at or above 0.80. nrfA encodes the periplasmic cytochrome c nitrite reductase that performs the second step of DNRA, reducing NO_2^- to NH_4^+. The gut produces nitrite from host nitrate via salivary nitrate reductase and from amino acid fermentation, and nrfA-mediated DNRA is a known anaerobic gut nitrogen-retention pathway. Independent biochemistry shows *B. xylanisolvens* and *B. cellulosilyticus* (both in the carrier list above) actively reduce nitrite *in vitro* (Hager et al., 2026). This is the only canonical N-cycling trait the gut community encodes more often than the soil syncom in this analysis, and it has an immediate biological rationale that the EcoFoldDB paper does not itself flag.

ACC deaminase / ethylene suppression (acdS) shows the opposite asymmetry and recovers a published soil-microbiome finding. Four of 15 soil genomes are acdS-

positive, three of which are *Bacteroidota* (*Mucilaginibacter* OAE612, *Niastella* OAS944, *Chitinophaga* OAE865) plus the canonical plant growth-promoting rhizobacteria *Pseudomonas simiae* WCS417 (NCBI assembly GCF_000698265.1) (Berendsen et al., 2015). Only 1 of 120 hCom2 genomes is acdS-positive (*Catenibacterium mitsuokai* DSM-15897).

The hCom2 polyphenol-cycling signal is concentrated in the same Coriobacteriia genera that the gut-microbiome literature already identifies as the dominant gut polyphenol biotransformers. *Eggerthella lenta* carries 37 polyphenol-flavonoid hits plus 11 hydroxycinnamic-acid hits, and *Slackia exigua* carries the matching enrichments, both consistent with prior work on *E. lenta*'s cardiac-glycoside-reductase activity and its broader polyphenol-active phenotype (Koppel et al., 2018; Haiser et al., 2013).

Synthesizing findings across two functional pipelines. The CATH-and-EC/KEGG/COG analysis of Section 4.4 and the EcoFoldDB-annotate analysis here read the hCom2-vs-soil contrast through different reference databases (CATH-S40 plus reviewed UniProt EC mapping versus the EcoFoldDB-curated structural references) and at different functional granularities (broad EC1 / KEGG-pathway / COG-category for the first, trait-level minimum-gene-set rules over a curated environmental panel for the second). They converge qualitatively. Soil leans aromatic / aerobic / secondary-metabolism active, and hCom2 leans plant-glycan / anaerobic / fermentative-central-metabolism dominated. The EcoFoldDB layer adds two specific observations that the Merizo-search layer does not surface directly. The first is the DNRA enrichment across 23 hCom2 commensal genomes (the bulk being *Bacteroides* and *Parabacteroides*, plus *Prevotella*, *Slackia heliotrinireducens*, a *Burkholderiales* isolate, *Desulfovibrio piger* and *Bilophila wadsworthia*), which is not predicted by the EcoFoldDB paper itself and which carries a clean gut-nitrogen-biochemistry rationale. The second is the recovery of the published soil-Bacteroidota ACC-deaminase enrichment in our 15-strain syncom subset. Convergence between the two pipelines is the decisive signal, and the trait-level surfacings are the granular complement to the broader fold-and-function portrait.

4.5 The TIPalpha case study: structural search as a discovery instrument

To illustrate practical use of the hCom2 structural proteome I use the *H. pylori* virulence factor TIPalpha as the query. The case study makes one strong claim alongside several deliberately weaker ones. The strong claim is that the TIPalpha

CATH fold 3.10.129.140 is recovered in 19 hCom2 proteins by structural search, with DALI Z 4.5–8.5 against the truncated TIPalpha reference at DALI-aligned sequence identity 4 to 16 percent and zero conventional-threshold MMseqs2 alignments to either TIPalpha or Lpp20 across the same 19 carriers. These are bona-fide TIPalpha CATH-fold structural homologs in gut commensals on the metric the chapter is built around, and the structural search recovers them where sequence-only annotation finds nothing. The weaker claims concern where each carrier sits inside the SHS2 Pfam clan, whether the carriers reproduce TIPalpha's molecular machinery, and whether they engage TIPalpha's published receptors. Computational triangulation through within-clan DALI Z, sequence Pfam, predicted processing class, spatial residue equivalencing, and AlphaFold-3 cofolding screen against the published TIPalpha receptors does not converge on a single functional verdict for the carriers.

Throughout the case I distinguish four levels of claim, and ask the reader to keep them distinct. *Fold-level* means CATH H-superfamily 3.10.129.140 membership by structural alignment, evaluated at the canonical TM at least 0.5 fold-level cutoff and DALI Z = 2 noise floor. *Family-level* means Pfam family membership by sequence, either PF02169 (Lpp20 lipoprotein) or PF16753 (*Helicobacter* TNF-alpha-Inducing protein). *Clan-level* means Pfam clan CL0319 (SHS2), the structural neighbourhood that contains both of those families plus dodecin (PF07311) and seven further entries. The case's strong claim is at the fold level. The four-axis triangulation in Section 4.5 operates at the family and clan levels. The AF3 cofolding screen in Section 4.5 operates at the functional-cognate level. The four levels are not interchangeable, and the chapter is calibrated to claim only what the evidence at each level supports.

Why TIPalpha

TIPalpha (HP0596, UniProt O25318, 192 residues) is a secreted *H. pylori* virulence factor that binds cell-surface nucleolin on gastric epithelial cells, internalises, and activates NF-κB to induce TNF-α and chemokine expression, with intermolecular Cys25–Cys25 and Cys27–Cys27 disulfide bridges at the N-terminus locking the secreted homodimer into the active form (Suganuma et al., 2008; Watanabe et al., 2010; Suganuma et al., 2021). The same disulfide-locked dimer binds a 20-bp GC-rich double-stranded DNA substrate at $K = 1.02 \times 10^6$ M^{-1} by isothermal titration calorimetry, with the DNA-binding interface formed by a positively-charged patch at the dimer interface (His60, Arg77, Arg81) (Gao et al., 2012). *H. pylori* is a long-recognised risk factor for gastric adenocarcinoma at the population level (Peek and Blaser, 2002), and an SS1-strain murine knockout of TIPalpha shows reduced

gastric inflammation and hyperplasia compared to wild type at one and four months post-infection (Morningstar-Wright et al., 2022).

Two practical features make TIPalpha a useful worked example for a structural-search resource. First, the family scope in the public sequence catalogues is conspicuously narrow. Pfam PF16753 (the TIPalpha family) contains only *Helicobacter* members at the time of analysis, so any non-*Helicobacter* fold-level hit recovered by structural search is interesting on the published-catalogue framing alone. Second, the family has a well-characterised structural neighbour at the family level (Lpp20, HP1456, Pfam PF02169) and a well-characterised related fold (Hotdog Thioesterase, CATH 3.10.129.10), so the search can be cross-checked against multiple references and a candidate sub-family assignment can be evaluated for robustness across reference choices. These two properties together make TIPalpha a case study with a clean question (does the TIPalpha CATH fold appear in commensals?) and a clean robustness test (where within the SHS2 Pfam clan does each carrier sit?).

Family scope and the SHS2 clan

In CATH the TIPalpha topology is fold 3.10.129.140, an H-level superfamily under parent topology 3.10.129 that is disambiguated from the sibling H-family 3.10.129.10 (Hotdog Thioesterase, FabZ-dominant in bacteria) at both the H-level and the Pfam-family level (PF16753 and InterPro IPR031923). The verbatim H-level name "*Helicobacter* TNF-alpha-Inducing Protein" reflects the membership of the curated CATH-S40 reference set at the time of curation, where all S40 representatives at this H-level were *Helicobacter* TIPalpha and Lpp20 entries, and is not a biological claim of phylum-level exclusivity.

The historical scope of the TIPalpha family was verified before any hCom2 hit was claimed. Pfam clan CL0319 (SHS2 clan) groups TIPalpha (PF16753), the Lpp20 lipoprotein family (PF02169), the Dodecin family (PF07311) and seven other Pfam entries under one structural fold neighbourhood. PF16753 contains only *Helicobacter* members in the public catalogues at the time of analysis, while PF02169 already spans both *Bacteroidota* and *Bacillota* by sequence. The closest pre-existing structural neighbour at the family level is the *H. pylori* paralog Lpp20 (HP1456), which shares the topology at Cα RMSD 3.3 angstrom by DALI superposition and is the named member of PF02169 (Vallese et al., 2017). A residue-level structural decomposition further notes that TIPalpha breaks into three regions, namely a flexible N-terminal extension, a central beta-sheet matching the dodecin fold, and a C-terminal helical

bundle resembling the SAM (sterile alpha motif) family (Tosi et al., 2009). Individual subdomains of the fold are therefore not unique to the TIPalpha family proper, and the published uniqueness claim sits on the integrated three-domain entry IPR031923 rather than on any single subdomain.

Two TIPalpha crystal structures plus one AlphaFold model anchor the structural search. PDB 3GUQ is the CATH-S40 alignment target, deposited as the truncated del-Tipα construct with chain A resolving 142 ordered residues and the biologically observed heart-shaped non-covalent homodimer reconstructed by crystallographic two-fold symmetry (Tsuge et al., 2009). PDB 3GIO is the full-length crystal, resolving construct residues 28 to 192 across two chains in the asymmetric unit (Jang et al., 2009). Both deposited crystal models lack the N-terminal Cys25 and Cys27 disulfide-bridge residues that lock the secreted dimer *in vivo* (the residues are not modelled because the N-terminal segment is disordered in the crystallographic constructs), so the pipeline searches for fold-level homologs of the central scaffold rather than for the disulfide-locked dimer interface. The AlphaFold Database monomer model of UniProt O25318 (AF-O25318-F1-model_v6) is the full 192-residue prediction and *does* include Cys25 and Cys27 in primary sequence and predicted local structure (Fleming et al., 2025). I use it later as the spatial-residue-equivalencing reference (Section 4.5, axis 4) at the points in the analysis that depend on the catalytic-residue positions being present in the reference structure.

Structural search recovers 19 hCom2 carriers of the TIPalpha CATH fold 3.10.129.140

Structural search recovers a clear signal. Foldseek in TM-align mode against the hCom2 structural database identifies 19 hits at normalised TM scores from 0.351 to 0.700 against PDB 3GUQ (Van Kempen et al., 2024). Two of these (*Mitsuokella multacida* 0933 and *Alistipes onderdonkii* 2463) were recovered only after re-running Foldseek with `-alignment-type 1 -exhaustive-search 1` to bypass the default 3Di prefilter. This is itself a useful operational lesson. The default Foldseek prefilter is tuned for speed at proteome scale and can drop fold-level matches that lie outside its k-mer index. Downstream users running case-study queries against the hCom2 database that are sensitive to remote-homolog recall should pair the default Foldseek search with an `-exhaustive-search 1` pass.

Pairwise DALI alignment confirms structural homology at Z scores from 4.5 to 8.5 against the truncated 3GUQ reference (median 6.9) at DALI-aligned sequence

identity 4 to 16 percent, with all 19 carriers comfortably above the Z = 2 noise floor for similar folds (Holm, 2020) and 10 of 19 carriers above Z = 8 against the strongest Lpp20 reference at the within-clan classification step (Section 4.5) (Holm, 2022). The cross-method comparison of Foldseek TM-align, Foldseek LoL-Align, Foldseek probability, and DALI Z-scores for the 19 carriers is shown in Figure 4.9. The 19 carriers span 14 hCom2 strains across two phyla, namely 18 *Bacteroidota* across 13 strains of five genera (*Alistipes*, *Bacteroides*, *Parabacteroides*, *Phocaeicola*, *Prevotella*), plus one *Bacillota* (*Mitsuokella multacida*).

Three representative pairwise DALI superpositions illustrate the topology recovery across genera. *Phocaeicola vulgatus* 0947 (formerly *Bacteroides vulgatus* per the 2020 reclassification, hereafter *P. vulgatus*) is shown in Figure 4.10a at Z 7.9. *B. ovatus* 3440 (Figure 4.10b, Z 8.5) is the highest-scoring carrier by DALI Z and one of four carriers flagged by InterProScan as a Pfam PF02169 (Lpp20-family) member (the other three are documented in Section 4.5). *A. onderdonkii* 2463 (Figure 4.10c, Z 5.8) is recovered only after the Foldseek prefilter was relaxed. Across all three, the TIPalpha CATH fold is recognisable despite the carrier sequence sitting at single-digit percent identity to TIPalpha.

Two paired sequence-conservation views (Figure 4.11) display the per-pair sequence relationships. Panel (a) is an MMseqs2 easy-search at default sensitivity (Steinegger and Söding, 2017) and shows that at conventional sequence-search settings neither the TIPalpha row nor the Lpp20 row returns any alignment to any carrier, with 0 of 38 reference-vs-carrier pairs aligning at the sequence level. A maximum-sensitivity check on the TIPalpha-vs-carrier subset (`-s 7.5 -exhaustive-search 1 -mask 0 -comp-bias-corr 0 -e inf`) recovers two borderline alignments at *e*-values 9.3 and 0.43, neither at conventional significance, confirming that the failure mode persists across MMseqs2 parameter regimes. Panel (b) is the DaliLite v5 pairwise percent identity over the structurally aligned residues only, following the convention used by the DALI tutorial and the remote-homolog template (Holm et al., 2023). The DALI-aligned percent identity is uniformly low (4 to 16 percent against TIPalpha) but the structural alignment is well-defined, which is exactly the regime in which structural search is meant to add value over sequence search.

Genomic context

Before triangulating where each carrier sits within the SHS2 clan, it is informative to look at what lives next to each carrier in its genome. I extracted the ±10-neighbour

window around each of the 19 carriers from its Bakta-annotated proteome, and ran the same Pfam scan over each neighbourhood that the proteome itself was run through.

The most informative pattern that emerges is a recurring tandem-cassette architecture. Five of fourteen carrier-bearing strains carry adjacent TIPalpha-CATH-fold proteins on the same strand, separated by intergenic gaps on the order of thirty base pairs. *B. xylanisolvens* DSMZ-18836 carries the cleanest case, where 5623 (193 residues) and 6064 (625 residues) sit at Bakta gene positions 866 and 867 with a 30-bp intergenic gap on strand −1. *B. ovatus* ATCC-8483 carries the same architecture, with 3440 (625 residues) plus a 193-residue TIPalpha-CATH-fold neighbour at relative position −1 on the same strand (this 193-residue neighbour is not in the original 19-carrier set because it sat below the structural-search threshold against the truncated 3GUQ reference, but the Pfam PF02169 hit on the neighbour and the genomic adjacency make it the small partner of the cassette). *B.* sp. 2-1-22 carries 3096 (625 residues) plus a 193-residue TIPalpha-CATH-fold neighbour at relative +1, the same architecture once more. *P. copri* DSM-18205 carries 1704 (219 residues) plus a 593-residue TIPalpha-CATH-fold neighbour at relative +1. *B. stercoris* ATCC-43183 carries the reversed-size-order version, where 2624 (433 residues) and 2529 (209 residues) sit 31 bp apart on the same strand with the multi-domain partner upstream. Both partners of the *B. stercoris* cassette are in the original 19-carrier set, while in the other four cassettes only one partner is.

A sixth strain, *A. onderdonkii* DSM-19147, carries 2419 and 2463 three genes apart on the same strand (gene indices 1855 and 1858, ∼4.5 kb total span across two intervening CDSs), but their pairwise TM-align score (TM-score ≈ 0.28 between the two carriers' relaxed structures) is too low to call them recent tandem duplicates. This could suggest that they are old paralogs whose structural alignment has decayed beyond the fold-level cutoff. *B. cellulosilyticus* DSM-14838 carries three carriers (2407, 3007, 3223) at three separate loci hundreds of kilobases apart, with 3007 and 3223 reading as intra-strain paralogs by TM-align (TM-score ≈ 0.59) and 2407 independent.

For context, a *H. pylori* reference genome MT5136 itself carries three TIPalpha/Lpp20-CATH-fold loci rather than the two implied by the standard "stand-alone TIPalpha plus HP1454–HP1457 operon" framing. There is a stand-alone TIPalpha locus at 589 kb, a tandem Lpp20 plus 303-residue-paralog cluster at 1530 kb (the canonical HP1454–HP1457 region), and a separate tandem LPP20 plus HP0838-family pair

at 852 kb. The tandem architecture is therefore not a *Bacteroidota* innovation. *H. pylori* carries two independent tandem cassettes of its own, and the hCom2 *Bacteroidota* pattern recapitulates a structural neighbourhood that already exists in the *Helicobacter* genome.

The second layer the genomic context adds is an HGT signal. AlienHunter (Vernikos and Parkhill, 2006) flags genomic regions whose interpolated variable-order motif statistics depart from the rest of the host genome, picking up candidate horizontally-acquired DNA without requiring a sequence-based homology call. I ran AlienHunter on each carrier-bearing genome and asked whether each carrier (or its tandem cassette) sat inside an AH-flagged region or within a ±5 kb buffer around one. The *B. xylanisolvens* 5623 + 6064 cassette is the strongest match in the set. AlienHunter flags a region at 1,111 to 1,118 kb on the carrier-bearing contig, and the 5623 + 6064 cassette sits at 1,107.4 to 1,109.9 kb, immediately outside the flagged region and inside the ±5 kb buffer. Most other carriers do not produce a strong AH signal, and the per-strain breakdown is in the underlying tables.

A potential hypothesis gleaned from the genomic-context layer is that the TIPalpha CATH fold has been disseminated through gut *Bacteroidota* as a structural neighbourhood rather than as a single gene, with tandem cassettes recurring across at least five strains and with at least one cassette sitting in a candidate HGT-adjacent locus. The pattern is consistent with the *Bacteroidales* conjugative mobilome that the community-portrait analysis already flagged at the CATH-superfamily level (Section 4.4, namely Tyrosine recombinase, Integrase catalytic core, Resolvase N-terminal, HUH-endonuclease relaxase enrichment in hCom2 versus soil). It is not, however, evidence that the carriers themselves are pathobiology-relevant. The same mobilome that distributes antimicrobial-resistance cassettes between *Bacteroidales* strains is the most plausible mechanism for distributing TIPalpha-CATH-fold surface proteins between them, and the carriers' biological roles are still set by the lipidation pathway, the receptor repertoire, and the host context, not by their genetic mobility per se.

To probe the operon context beyond the immediate cassette partners, I extended the analysis to a wider ±15 kb window around each carrier and annotated every coding sequence in every window via the InterProScan REST API (Jones et al., 2014) (PfamA, NCBIfam, HAMAP, Phobius, SignalP, MobiDBLite member databases). The annotated windows were then compared to the equivalent ±15 kb windows of the canonical *H. pylori* TIPalpha and Lpp20 reference loci (*H. pylori* 26695 HP0596

and HP1456, plus the three MT5136 TIPalpha/Lpp20-CATH-fold loci described above), asking the operon-level question of which carrier-flanking families are also present in a TIPalpha-context but not in a Lpp20-context reference window, and vice versa, at a strict cross-context cutoff $E < 10^{-5}$. Across all 19 carriers, only one observation passes the cutoff. Both *A. onderdonkii* hits (2419 and 2463) sit adjacent to a Class A bifunctional penicillin-binding protein (PfamA PF00912 transglycosylase plus PF00905 transpeptidase plus PF28500 middle domain, $E = 3.6 \times 10^{-49}$ for the transglycosylase domain) that is structurally analogous to *pbp1a* (HP0597), the gene immediately downstream of TIPalpha at the canonical 589-kb locus in both *H. pylori* 26695 and MT5136. This operon-context match is convergent with the SignalP-6 axis. Both *A. onderdonkii* carriers are predicted Sec/SPI secreted-soluble (the TIPalpha-like processing pattern) at $P_{SP} = 0.999$, so the genomic neighbourhood and the central-gene processing class point in the same direction for this strain. Figure 4.12 compares the *A. onderdonkii* window to the *H. pylori* 26695 TIPalpha and Lpp20 reference loci at the same ±15 kb scale.

Two important hedges qualify the *A. onderdonkii* observation. The effective sample size is $N = 1$ rather than $N = 2$. The two carriers are adjacent paralogs in the same strain (*A. onderdonkii* DSM-19147, with the three-gene/~4.5 kb adjacency already noted above) and they share the same flanking penicillin-binding-protein gene because their ±15 kb windows overlap. The flanking gene carries the full Class A architecture (transglycosylase plus transpeptidase plus middle domain) but lacks the PBP1A-specific NCBIfam TIGR02074, which the *H. pylori* references hit at $E = 0.0$, and the architectural class is conserved but orthology to *pbp1a* is not claimed. The corresponding negative findings reinforce the sparse-signal reading. At the same strict cutoff, no carrier in the panel carries a Lpp20-LpoB operon partner, in particular none of the four SignalP-LIPO carriers shows an operon-level Lpp20-context family signature. Furthermore, 17 of 19 carriers show no shared family at all with any *H. pylori* TIPalpha or Lpp20 reference window beyond ubiquitous housekeeping families (ABC-transporter components, ribosomal protein L34) that co-occur by chance across genomes. The operon-context layer therefore produces one positive observation, 17 effective negatives, and zero observations of Lpp20-context conservation, and is best read as a sparse but interpretable additional line of evidence rather than as a converging signal across the full panel.

A recurring functional split within the cassette pattern is worth flagging for follow-up. In *B. xylanisolvens* the small partner is the strongest SignalP-6 lipoprotein call in

the set (5623, $P_{\mathrm{LIPO}} = 1.000$ on the SignalP axis described in Section 4.5) and the multi-domain partner is borderline (6064, $P_{\mathrm{LIPO}} = 0.333$). In *B. stercoris* the small partner is a marginal lipoprotein call (2529) and the multi-domain partner is predicted Sec/SPI-secreted soluble (2624). The per-carrier processing-class calls behind this lipoprotein-versus-soluble split are unpacked in Section 4.5 axis 3.

Multi-axis classification of the carriers

A structural hit against TIPalpha alone does not tell me where each carrier sits inside the SHS2 Pfam clan. The carriers share the TIPalpha CATH fold (3.10.129.140), and the question is whether they are most cleanly read as members of the TIPalpha family proper (Pfam PF16753), the closely related Lpp20 family (Pfam PF02169), or as fold-level boundary cases belonging to neither published Pfam family by sequence. I utlize five computational assays for this question. However, do not converge on a single answer, and I think the honest interim reading is that no one of them by itself is enough to license a definitive sub-family assignment or functionality for any individual carrier. Each axis carries its own diagnostic value, and each has its own characteristic failure mode.

I re-aligned each of the 19 carriers pairwise against seven references, namely three for TIPalpha (the truncated 3GUQ monomer used as the structural query above, the full-length 3GIO crystal, and the AFDB v6 monomer of UniProt O25318), two for Lpp20 (the 5OK8 crystal and the AFDB monomer of UniProt P0A0V0), the Hotdog Thioesterase 1U1Z (FabZ from *H. pylori*, the canonical bacterial member of the sibling H-family CATH 3.10.129.10) as a related-fold outgroup, and the *Mycobacterium tuberculosis* dodecin crystal 2YIZ (Pfam PF07311, same SHS2 clan as PF16753 and PF02169) as a within-clan partial-overlap reference motivated by the residue-level decomposition that places dodecin as the central-subdomain match for TIPalpha (Jang et al., 2009; Vallese et al., 2017; Tosi et al., 2009; Fleming et al., 2025). Multi-chain crystal structures (5OK8, 1U1Z, 2YIZ) were reduced to chain A before alignment. Three tools were run on every (carrier, reference) pair, namely MMseqs2 easy-search at maximum sensitivity for sequence-level identity, Foldseek easy-search in TM-align mode (`-alignment-type 1 -exhaustive-search 1`) for structural similarity normalised by query length, and DaliLite v5 `dali.pl -outfmt summary -clean` for Z-score, RMSD, alignment length, and DALI-aligned percent identity.

Picking the strongest reference within each fold family by DALI Z, 14 of 19 carriers achieve a higher Z against the better of the two Lpp20 references than against the

better of the three TIPalpha references, and 5 of 19 achieve the reverse. Zero carriers return any DALI hit above the Z = 2 noise floor against the Hotdog reference, and zero carriers return a higher Z against dodecin than against the better of TIPalpha or Lpp20. The mean DALI Z delta (Lpp20 minus TIPalpha) across the 19 carriers is +1.07 (median +1.7, range −2.2 to +3.4), and the asymmetry is robust to TIPalpha-reference truncation, because using the full-length 3GIO crystal in place of the truncated 3GUQ monomer raises every carrier's TIPalpha Z score by at most +0.4 (insufficient to flip any carrier), and AFDB monomers of both TIPalpha and Lpp20 reproduce the same per-carrier ranking. DALI-aligned percent identity over the structurally equivalent core follows the same direction (*P. vulgatus* 0947 at 9 percent against TIPalpha versus 17 percent against Lpp20, *B. ovatus* 3440 at 10 percent versus 16 percent). Per-carrier DALI Z scores against the strongest within-family reference and the resulting within-clan ranking are reported in Table 4.2.

MMseqs2 at maximum sensitivity returns alignments for only two of the 19 carriers across the 114 (carrier, reference) pairs at conventional significance. *P. johnsonii* 2360 aligns to both Lpp20 references (24.4 percent identity to the 5OK8 crystal at e-value 9.6×10^{-4} and 22.5 percent identity to the AFDB monomer at 5.9×10^{-6}), while *B. cellulosilyticus* 2407 aligns only weakly to TIPalpha (27.6 percent identity to 3GUQ at e-value 0.49 and 26.3 percent to the AFDB monomer at e-value 0.12, neither reaching the conventional 10^{-3} floor), with a single borderline *A. onderdonkii* 2419 alignment to the TIPalpha AFDB monomer at 20.9 percent identity and e-value 1.4. The remaining 16 carriers return no MMseqs2 alignment under any of these settings.

Dodecin (PF07311) returns DALI hits in all 19 carriers at Z scores from 3.2 to 6.0 but with mean alignment length 58 residues against the 69-residue dodecin chain, less than half the mean alignment length of 125 to 127 residues attained against TIPalpha and Lpp20. The dodecin signal is therefore a partial-subdomain match limited by the size of the dodecin chain rather than a full-topology alternative.

The DALI Z asymmetry is real and reproducible, but its interpretive weight as a sub-family verdict is limited. The within-family DALI Z difference between an Lpp20 reference and a TIPalpha reference for the same carrier is small-fold noise of order ±2 when both references share the same fold, because both Lpp20 and TIPalpha sit at the same CATH H-level. A +1 Z toward Lpp20 over TIPalpha is therefore not by itself a strong family assignment for any individual carrier. The absolute Z magnitude (6 to 11 against Lpp20, 5 to 8.5 against TIPalpha) does establish that the

carriers share the CATH H-superfamily 3.10.129.140 with both references, but the within-family direction sits inside the noise band for half the carriers.

Table 4.2: Per-carrier DALI Z scores for the 19 hCom2 carriers of CATH 3.10.129.140 against the strongest within-family reference, with the verdict by maximum DALI Z within each fold family. **Z TIPa max** is the maximum DALI Z across the three TIPalpha references (3GUQ, 3GIO, AF-O25318), **Z Lpp20 max** the maximum across the two Lpp20 references (5OK8, AF-P0A0V0), and Δ_{L-T} the within-family Z delta. **Z Hotdog** is the DALI Z against 1U1Z FabZ (the related-fold sibling H-family outgroup), with a dash indicating no DALI hit above the Z = 2 noise floor. **Z Dodecin** is the DALI Z against the *M. tuberculosis* 2YIZ reference, the within-clan partial-subdomain reference (Tosi et al., 2009). **Verdict** is the within-clan family with the highest maximum Z.

Carrier	Z TIPa max	Z Lpp20 max	Δ_{L-T}	Z Hotdog	Z Dodecin	Verdict
P. vulgatus 0947	7.9	11.3	+3.4	–	5.3	Lpp20
B. ovatus 3440	8.7	11.1	+2.4	–	4.5	Lpp20
B. sp 3096	8.3	11.0	+2.7	–	4.6	Lpp20
B. xylanisolvens 6064	8.1	10.3	+2.2	–	3.7	Lpp20
B. rodentium 2626	7.7	9.4	+1.7	–	4.4	Lpp20
B. cellulosilyticus 3007	6.9	9.2	+2.3	–	6.0	Lpp20
B. xylanisolvens 5623	5.6	8.9	+3.3	–	3.2	Lpp20
B. fragilis 3363	6.6	8.7	+2.1	–	6.0	Lpp20
B. sp 3173	6.4	8.6	+2.2	–	6.0	Lpp20
M. multacida 0933	6.3	8.6	+2.3	–	4.9	Lpp20
B. plebeius 2530	6.6	7.9	+1.3	–	4.4	Lpp20
B. cellulosilyticus 2407	4.8	6.4	+1.6	–	5.0	Lpp20
A. onderdonkii 2419	5.9	6.0	+0.1	–	4.3	Lpp20
B. stercoris 2624	5.3	5.8	+0.5	–	3.7	Lpp20
P. johnsonii 2360	7.7	6.0	-1.7	–	4.6	TIPalpha
B. stercoris 2529	7.8	5.9	-1.9	–	4.0	TIPalpha
P. copri 1704	7.1	5.7	-1.4	–	6.0	TIPalpha
B. cellulosilyticus 3223	7.5	5.3	-2.2	–	4.0	TIPalpha
A. onderdonkii 2463	5.8	5.2	-0.6	–	5.2	TIPalpha

I additionally re-ran DALI for each carrier against AFDB monomer references for the remaining seven Pfam CL0319 clan members (GyrI-like PF06445, Archease PF01951, SHS2-Rpb7-N PF03876, SHS2-FtsA PF02491, SOUL PF04832, Cass2 PF14526, DUF6926 PF21977). 19 of 19 carriers still select either Lpp20 or TIPalpha as the closest within-clan family. The largest DALI Z attained against any of the seven newly tested references is 4.9 (Archease against *M. multacida* 0933, SOUL against *B. fragilis* 3363), well below the 6 to 11 Z range against Lpp20 and TIPalpha. Two clan caveats apply. The DUF6926 representative AFDB monomer is small (83 residues) and returned no DALI hit against any carrier. SHS2 sits as a subdomain

inside FtsA and Rpb7-N rather than as a full-chain match, so full-chain DALI underweights the SHS2 contribution and 17 of 19 FtsA pairs and 13 of 19 Rpb7-N pairs returned no DALI hit at all, leaving the SHS2-only subdomain re-extraction for follow-up.

InterProScan was run on each of the 19 carriers and on the *H. pylori* TIPalpha (HP0596) and Lpp20 (HP1456) references (Jones et al., 2014). All 21 inputs return an N-terminal signal peptide call from Phobius and at least one SignalP HMM (residue ranges 1 to 18 through 1 to 32). Pfam PF02169 (LPP20 lipoprotein, InterPro IPR024952 "Lipoprotein LPP20-like domain") is detected on four hCom2 carriers, namely *B. ovatus* 3440 (residues 26 to 118, *e*-value 0.01), *B.* sp. 2-1-22 3096 (residues 26 to 118, *e*-value 0.012), *B. xylanisolvens* 6064 (residues 26 to 118, *e*-value 0.012), and *M. multacida* 0933 (residues 57 to 125, *e*-value 1.4×10^{-3}). Pfam PF16753 (*Helicobacter* TNF-alpha-Inducing protein) is detected on the *H. pylori* TIPalpha reference (residues 32 to 181, *e*-value 4.0×10^{-83}) and on no hCom2 carrier. Pfam PF19672 (DUF6175) is detected on *A. onderdonkii* 2463 at residues 163 to 419 (*e*-value 2.7×10^{-40}) and on no other input. Gene3D superfamily 3.10.28.20 covers residues 24 to 125 of the three *Bacteroides* PF02169 carriers (*B. ovatus* 3440, *B.* sp. 2-1-22 3096, *B. xylanisolvens* 6064) at *e*-values from 1.1×10^{-20} to 1.3×10^{-20}. Gene3D superfamily 3.10.129.140 (the CATH H-level superfamily named "*Helicobacter* TNF-alpha-Inducing protein") is detected on both *H. pylori* references, on TIPalpha at residues 34 to 192 (*e*-value 1.3×10^{-72}) and on Lpp20 at residues 39 to 174. The ProSite PS51257 "Prokaryotic membrane lipoprotein lipid attachment site profile" is recovered on four hCom2 carriers (*P. johnsonii* 2360, *B. xylanisolvens* 5623, *B. stercoris* 2529, and *B. rodentium* 2626) and on the *H. pylori* TIPalpha reference. The *H. pylori* Lpp20 reference additionally carries the InterPro Lpp20 family entry IPR002217 over residues 1 to 175 plus the PIRSF (PIRSF011368) and PRINTS (PR01019) family signatures over the full chain.

The Pfam axis is the only axis on which a published-catalogue assignment is unambiguous. Four carriers are PF02169 members by sequence, while the remaining fifteen are not in any of the SHS2-clan Pfam families by sequence. Zero of nineteen are PF16753 by sequence. The four PF02169-by-sequence carriers also rank high by axis 1 DALI Z against Lpp20 (*B. ovatus* 3440 at Z 11.1, *B.* sp. 2-1-22 3096 at 11.0, *B. xylanisolvens* 6064 at 10.3, *M. multacida* 0933 at 8.6). The remaining 15 carriers are at the structural fold boundary in both directions in the public catalogues. *A. onderdonkii* 2463 additionally carries Pfam PF19672 (DUF6175) on a C-terminal

domain (residues 163 to 419), which suggests a fusion architecture rather than a pure Lpp20-family lipoprotein.

Some short biological context helps frame the Pfam axis. TIPalpha's molecular signature in *H. pylori* is a secreted soluble homodimer locked by intermolecular Cys25–Cys25 and Cys27–Cys27 disulfide bridges at the N-terminus, whose active dimer binds cell-surface nucleolin on gastric epithelial cells, internalises, and drives NF-κB activation to induce TNF-α and chemokine expression, while the same disulfide-locked dimer also binds GC-rich double-stranded DNA at $K = 1.02 \times 10^6$ M^{-1} by isothermal titration calorimetry with a positively-charged patch at the dimer interface (His60, Arg77, Arg81) (Tsuge et al., 2009; Jang et al., 2009; Suganuma et al., 2008; Watanabe et al., 2010; Suganuma et al., 2021; Gao et al., 2012). Lpp20's molecular signature, by contrast, is a triacylated outer-membrane lipoprotein recognised by patient sera, with C16:0 and C18:0 acyl chains attached by the canonical Lgt + LspA + Lnt pipeline and with TLR2 partner choice (TLR2/TLR1 versus TLR2/TLR6) tracking the triacylated-versus-diacylated state of the lipid anchor (Keenan et al., 2000; McClain, Voss, and Cover, 2020; McClain et al., 2024; Jung et al., 2023). The two signatures imply different InterProScan footprints. The Lpp20 footprint is the lipobox motif PS51257 plus the PF02169 (or related) Pfam call, and the TIPalpha footprint is the N-terminal signal peptide plus the PF16753 family call. In the public catalogues at the time of analysis only *H. pylori* TIPalpha itself carries PF16753, so the TIPalpha-side annotation is structurally narrow by construction and a *Bacteroidota* carrier cannot be flagged as TIPalpha-family-by-sequence under this scheme. *Bacteroides* also triacylate their lipoproteins through the recently characterised Lnb N-acyltransferase rather than canonical Lnt (Armbruster et al., 2024), so the TLR2-partner-choice phenotype that tracks the triacylated-versus-diacylated state in *H. pylori* Lpp20 is not automatically reproducible in commensal carriers without direct lipid analysis.

SignalP-6.0 (Teufel et al., 2022) returns calibrated per-protein probabilities for five processing classes (Sec/SPI, Sec/SPII = LIPO, TAT, TATLIPO, PILIN), and the relevant axis for TIPalpha-versus-Lpp20 is Sec/SPI (secreted soluble, the TIPalpha pattern) versus Sec/SPII (lipoprotein, the Lpp20 pattern). PROSITE PS51257 (Sigrist et al., 2010) is run alongside as the canonical PROSITE lipobox profile that InterProScan uses, and returns a binary lipobox call. The 19 carriers (with the canonical *H. pylori* TIPalpha and Lpp20 references as positive controls) sort into four buckets. Three carriers are predicted lipoproteins by SignalP-6 ($P_{\mathrm{LIPO}} \approx 1.000$)

with PS51257 positive (*B. xylanisolvens* 5623, *B. rodentium* 2626, *P. johnsonii* 2360). One is a marginal lipoprotein call at $P_{\mathrm{LIPO}} = 0.500$ where PS51257 still hits but the SignalP probability is genuinely undetermined (*B. stercoris* 2529). Thirteen are predicted Sec/SPI secreted-soluble with $P_{\mathrm{SP}} \geq 0.83$ and eleven of thirteen at $P_{\mathrm{SP}} \geq 0.99$. Two are borderline at $P_{\mathrm{LIPO}} \approx 0.333$ (*B. xylanisolvens* 6064 and *B.* sp. 2-1-22 3096, both 625-residue chimeras with the TIPalpha-CATH-fold module at residues 26 to 118 and a long unrelated C-terminal extension that confounds the per-protein lipidation call).

The caveat is that SignalP-6 mis-classifies a ground-truth case in the panel. One of thw two proteins in the 21-input panel whose processing class is biologically known is canonical *H. pylori* TIPalpha, which is biologically a secreted soluble protein (Sec/SPI). SignalP-6 calls it as LIPO at $P_{\mathrm{LIPO}} = 1.000$. PS51257 also hits the N-terminal Cys25/Cys27 motif on TIPalpha. This is a documented sequence-level false positive driven by the Cys25/Cys27 disulfide-bridge pair masquerading as a lipobox motif, but it means that the only in-panel ground-truth comparison shows SignalP-6 systematically over-calling LIPO on TIPalpha-like proteins. The four hCom2 carriers SignalP-6 calls as LIPO could in principle be similar false positives. PS51257 cross-validation (all four also positive) does not eliminate the concern because PS51257 also hits canonical TIPalpha. Symmetrically, the thirteen Sec/SPI calls could be over-confident, since the only ground-truth case the predictor sees in this panel is one where it errs in the opposite direction. The SignalP-6 axis is therefore informative as a heuristic but cannot by itself license a lipoprotein-versus-secreted-soluble verdict for any individual carrier without experimental follow-up.

TIPalpha's biological mechanism mainly depends on five key residues, namely Cys25 and Cys27 for the disulfide-locked dimer interface, and His60, Arg77, Arg81 for the DNA-binding patch at the dimer interface (Tsuge et al., 2009; Gao et al., 2012). At the 4 to 16 percent sequence identity at which the carriers sit, sequence-alignment-based residue mapping is unreliable. The defensible alternative is to take the DALI structural alignment between each carrier and a TIPalpha reference that contains the catalytic residues, read off the residue at the position structurally equivalent to each TIPalpha position, and classify the carrier residue as identical, chemically compatible, different, or gap.

The reference structure for this spatial residue test is the AlphaFold Database monomer model AF-O25318-F1-model_v6. The AFDB monomer is the full 192-residue prediction and does include Cys25 and Cys27 in primary sequence and

predicted local structure, so it is the correct reference for this test. I ran DALI pairwise alignments between AF-O25318-F1-model_v6 and each of the 19 carriers, and extracted the carrier residue at each of the five TIPalpha critical positions via the DALI structural-equivalence section.

The aggregate result is that none of the carriers reproduce TIPalpha's catalytic-residue layout. Zero of nineteen carriers retain both Cys25 and Cys27. One carrier (*B. cellulosilyticus* 2407) retains Cys27 alone at the structurally equivalent position. Eight carriers have no DALI structural equivalence at the Cys25 position (the position is in TIPalpha's flexible N-terminal extension and the structural alignment often does not pair it). The remaining ten have non-Cys residues at the equivalent positions. Zero of nineteen carriers retain all three of His60, Arg77, Arg81. Two carriers (*B. stercoris* 2624 and *P. johnsonii* 2360) retain Arg81 alone at the structurally equivalent position. The disulfide-locked dimer interface and the DNA-binding patch that together define TIPalpha biology in *H. pylori* are therefore not preserved in any of the carriers at residue-level resolution, even when the reference is the AFDB structure that contains those residues.

The previous four axes do not converge on a clean Lpp20-versus-TIPalpha sub-family verdict for the 19 carriers, and I do not think computational triangulation alone can produce one for a target this distant in sequence space. They produce four partially overlapping reads. Axis 1 (DALI Z) puts 14 of 19 closer to Lpp20 than to TIPalpha and 5 of 19 closer to TIPalpha than to Lpp20, but with within-family Z deltas inside the same-fold noise band for many carriers, so the per-carrier sub-family assignment from this axis alone is not robust. Axis 2 (Pfam) confidently identifies 4 of 19 as PF02169 by sequence and 0 of 19 as PF16753 by sequence. The remaining 15 are not in either Pfam family at the time of analysis. Axis 3 (SignalP-6) splits the 19 into 3 confident lipoprotein plus 1 marginal lipoprotein plus 13 secreted-soluble plus 2 borderline calls, with the caveat that the only protein in the panel with biological ground truth (TIPalpha itself) is mis-classified by SignalP-6 in the opposite direction, which limits how much weight the axis can carry on its own. Axis 4 (spatial residue equivalencing) rules out reproduction of TIPalpha's catalytic-residue layout for all 19 carriers regardless of which sub-family they belong to.

Axes 1–3 operate at the family and clan levels. Axis 4 alone crosses into the functional-cognate level (at the residue layout, not the binding event itself), and the remaining complex-level question is taken up by the AlphaFold-3 cofolding screen in Section 4.5 as the fifth axis of the same triangulation.

An AlphaFold-3 cofolding screen against TIPalpha receptors

The four-axis triangulation above does not reach the complex-level question of whether any carrier engages TIPalpha's published receptors. To probe that directly I ran an AlphaFold-3 cofolding screen (Abramson et al., 2024) that asks whether any carrier engages the receptors TIPalpha engages in *H. pylori*, treating the screen as the fifth axis of the same case-study triangulation.

The receptor side of the screen comes from the TIPalpha and Lpp20 biology in the literature. TIPalpha's published partners are cell-surface nucleolin (NCL), which it binds via the four RNA-binding domains plus GAR/RGG tail of the nucleolin 284–710 fragment, with the engaged TIPalpha–NCL complex then internalising into gastric epithelial cells and driving NF-κB activation (Watanabe et al., 2010; Suganuma et al., 2008), and a 20-bp GC-rich double-stranded DNA substrate that the disulfide-locked TIPalpha homodimer binds dimer-interface patch (His60, Arg77, Arg81) (Gao et al., 2012). A recent context-dependent "death triad" model places TIPalpha at a cell-fate switch between proinflammatory NF-κB signaling (dimer plus saturated nucleolin) and Ras-pathway proliferation (monomer plus desensitised nucleolin) (Mahant et al., 2021). The published Lpp20 partners are the TLR2/TLR1 and TLR2/TLR6 heterodimers, with lipidation-state-dependent partner choice (triacylated Lpp20 engages TLR2/TLR1, diacylated Lpp20 engages TLR2/TLR6) (Jung et al., 2023; McClain et al., 2024). Lpp20 also engages platelet surfaces in chronic immune thrombocytopenia (Takeuchi et al., 2021), and the HP1454 T-cell agonist appears as an operonic partner in the canonical 1530-kb cluster (Capitani et al., 2019). For the present screen I focused on the TIPalpha receptors (NCL plus dsDNA), because Lpp20's receptor on platelets is unknown and TLR activation is mainly mediated through the lipidation of the protein itself.

The screen inputs are the 19 hCom2 carriers plus *H. pylori* TIPalpha and Lpp20 as references (21 inputs total), each run as both a monomer and a homodimer, and each predicted in the presence of either the mature form (post-signal-peptide-cleavage) or the full-length form of the bait. The receptor panel is NCL plus dsDNA-20 plus six structured negative-control receptors deliberately chosen to be biology-bearing rather than random decoys, namely U2AF2-RRMs and hnRNPA1-UP1 as fold-class controls (NCL's binding domain is built of four RRMs, so a candidate that scores on U2AF2 or hnRNPA1 is matching RRM-fold rather than NCL-specific surface), plus CALR, ANXA2, LGALS3-CRD, and MBP as broader sticky-surface controls. The full panel is 21 inputs $\times$ 18 partner combinations = 378 AF3 jobs.

The default AF3 ipTM score gives an uninformative reading on this dataset because both bait-side and receptor-side disorder bias the metric. Nucleolin's GAR/RGG tail (residues 648–710) is intrinsically disordered, and several hCom2 carriers carry their own flanking disorder. I therefore rescored every cofolding job with two metrics that correct for this. ipSAE (Dunbrack, 2025) re-normalises the interface confidence by the cross-chain pairwise alignment-error matrix to suppress the contribution of accessory disordered regions. iLIS (Kim et al., 2024) is a 2026 extension of the 2024 Local Interaction Score framework that computes a contact-filtered interface score, with iLIS ≥ 0.223 as the published binarising threshold on yeast/fly/human Y2H reference sets.

The first presented finding should shape how the rest of the results should be interpreted.

First, *H. pylori* TIPalpha itself shows no AF3 receptor signal in this screen. All six TIPalpha-versus-receptor pairs (monomer / dimer × mature / full × NCL / dsDNA) return ipSAE = 0.0 with zero interface residues at PAE below 10, and the sensitivity rerun at PAE below 15 still returns zero on every chain pair. AF3 produces a confident TIPalpha homodimer when NCL is also present in the box ($\text{ipSAE}_{AB} = 0.490$, $\text{iLIS}_{AB} = 0.551$ for the `dim_full_ncl` job) but does not place any TIPalpha residue at the published nucleolin or DNA interface in any of the six TIPalpha receptor jobs. The straightforward interpretation is that AF3 has not learned the TIPalpha–receptor interface from its training set. The consequence for the screen is a real limitation rather than a strengthening of any carrier-side finding. The screen has no internal AF3-validated TIPalpha-mimicry reference, so any positive call on the carrier side rests entirely on the negative-control panel for its specificity claim and not on an internal positive-control benchmark. A carrier-side hit means "AF3 places this carrier at the NCL surface with a confidence that exceeds the structured-control-receptor baseline," not "AF3 places this carrier where AF3 places TIPalpha." The screen design partly compensates for the missing positive control with structured biology-bearing negative controls rather than random decoys, but it cannot make the absence of an AF3-validated TIPalpha-mimicry baseline go away.

Secondly, the *P. copri* 1704 monomer (full-length bait) plus NCL prediction is the only carrier-receptor combination in the main batch that passes the iLIS threshold cleanly, with iLIS = 0.529 and ipSAE = 0.542. The same bait at mature-form returns iLIS = 0.209, just below the 0.223 cutoff but well separated from the two RRM fold-

class negative controls (U2AF2 and hnRNPA1, both at iLIS = 0) and from the other four negative controls (CALR at iLIS = 0.025, ANXA2 / LGALS3-CRD / MBP at iLIS = 0). The mature-form prediction therefore passes the fold-class specificity test against the two RRM controls, and the full-form prediction is consistent with the same direction but does not yet have a matched fold-class control on file because U2AF2 and hnRNPA1 were only run against the mature-form bait. The two additional AF3 jobs needed to lock the full-form call (full-form *P. copri* 1704 against U2AF2-RRMs and against hnRNPA1-UP1) are in the controls-still-running set. Without those controls the full-form result is the strongest single carrier-side AF3 signal in the dataset, with the caveat from observation one that the absence of an AF3-validated TIPalpha-mimicry positive control means the result rests on the negative-control panel alone. The *P. vulgatus* 0947 dimer (mature form) plus dsDNA-20, also passes the iLIS threshold at iLIS = 0.491, but the same job's homodimer A↔B interface scores at ipSAE$_{AB}$ = 0.359, comparable to the protein–DNA signal in the same prediction, so AF3 may be primarily predicting the homodimer in this job and the DNA contact may be secondary. This DNA-side hit is a borderline finding rather than a clean positive call.

Furthermore, four carriers score on RRM- or annexin-class negative controls but not on NCL itself, namely *B. fragilis* 3363 (iLIS 0.540 on U2AF2-RRMs and 0.391 on ANXA2), *B.* sp. 2-1-16 3173 (iLIS 0.445 on U2AF2 and 0.379 on ANXA2), *B. cellulosilyticus* 3007 (iLIS 0.438 on U2AF2), and *A. onderdonkii* 2463 (iLIS 0.392 on CALR), with iLIS = 0 against NCL itself in every case. These cannot be distinguished by AF3 alone between two interpretations, namely a generic sticky surface that AF3 places against many partners, or a real RRM- or annexin-class binding affinity. What the panel does establish unambiguously is that an NCL-specific hit is not licensed for any of these four carriers, which is the panel's stated purpose.

The headline at the time of writing is that under panel-controlled scoring with three independent rescoring metrics, the AF3 cofolding screen finds one carrier (*P. copri* 1704) with a cleanly specific NCL-side signal, one carrier (*B. vulgatus* 0947) with a borderline DNA-side signal, four carriers with non-NCL-specific structured-receptor binding, and the remaining thirteen with no AF3 signal on either receptor. TIPalpha itself is silent on AF3-predicted receptor binding, which means none of these carrier-side calls have an internal AF3 positive-control reference. They rest entirely on the negative-control panel. The screen is still acquiring controls (full-form-bait fold-class controls for the *P. copri* 1704 hit, DNA-side negative controls for the *B. vulgatus* 0947

borderline, and a wider PAE-cutoff sensitivity sweep on the homodimer signals), and the call set above should be expected to firm up, or to revise, as those controls land.

Further biological interpretation of the *P. copri* 1704 hit and of the *B. vulgatus* 0947 borderline requires experimental follow-up that this thesis does not perform. Until that follow-up exists, the AF3 layer is informative about which carriers are worth experimental priority rather than about which carriers reproduce TIPalpha biology.

Figure 4.13 renders the full panel as a 42-row by 9-column iLIS heatmap, with rows being the 21 inputs at each of two stoichiometries (monomer and homodimer) and columns the nine receptor or receptor-form combinations on file. iLIS (Kim et al., 2024) is the geometric mean of the published LIS metric and a contact-restricted variant, computed at PAE $\leq$ 12 Å and a Cβ-Cβ contact cutoff of 8 Å, and is bounded between 0 and 1. The published binarising threshold of 0.223 is deliberately not overlaid because that cutoff was derived from yeast, fly, and human Y2H reference sets that do not include disordered receptors or protein-DNA contacts, and because per-protein decoy-controlled comparison (Figure 4.14) is a more defensible call for a hypothesis-generation screen than a single hard cutoff. The dsDNA-20 column should be read as "AF3 finds an interface" rather than "AF3 finds a base-specific contact," since DNA has no Cβ atom for the cLIS filter as published. The heatmap is sparse, with most cells exactly zero, and the non-zero signal concentrates on the top rows of the plot in the same four cases the prose above describes, namely *P. copri* 1704 (two NCL cells lit at 0.21 and 0.53, all decoy cells dark), *B. vulgatus* 0947 in the dimer row (a single dsDNA-20 cell at 0.49), and four monomer rows (*B. fragilis* 3363, *B.* sp. 2-1-16 3173, *B. cellulosilyticus* 3007, *A. onderdonkii* 2463) with non-zero cells on RRM-class or CALR decoys but zero on NCL itself. The TIPalpha rows at the bottom of the plot are flat in every multi-chain cell, the receptor-dependent dimer pathology already noted and reproduced again in Figure 4.15.

To highlight any NCL-binders, Figure 4.14 replaces the heatmap's per-cell view with the panel-controlled NCL-specificity question and uses ipSAE rather than iLIS as the scoring metric, because ipSAE (Dunbrack, 2025) normalises by the structured-interface size rather than the raw chain length and is therefore receptor-size invariant in a way iLIS is not (this matters because receptor sizes in the panel range from roughly 140 residues for LGALS3-CRD to roughly 427 for full-form NCL). For each monomer-form bait that passes a minimum-signal filter (max of NCL_mat, NCL_full, and best decoy ipSAE at least 0.05), the figure plots the signed difference ΔipSAE = ipSAE(NCL_mat) − ipSAE(best decoy), with positive Δ meaning NCL-

specific and negative Δ meaning decoy-preferring. The decoy panel is restricted to mat-form bait against six controls (U2AF2-RRMs, hnRNPA1-UP1, CALR, ANXA2, LGALS3-CRD, MBP) to keep the fold-class comparison symmetric. The full-form NCL ipSAE is shown as a per-bar annotation for context but not used in the Δ. Five baits survive the filter, and they split sharply. *P. copri* 1704 is the only positive (Δ = +0.21, with NCL_mat ipSAE = 0.21 and a best decoy ipSAE of 0.01 against CALR, plus full-form NCL ipSAE = 0.54 in annotation), and is the only candidate in the screen that scores on NCL without also scoring somewhere on the decoy panel. The other four baits (*B. fragilis* 3363, *B.* sp. 2-1-16 3173, *B. cellulosilyticus* 3007, *A. onderdonkii* 2463) are all negative-Δ, with best-decoy ipSAE between 0.39 and 0.55 on either U2AF2-RRMs or CALR and NCL ipSAE exactly zero. Sixteen further baits have no detectable signal anywhere on the receptor panel and are excluded from the figure for clarity. The qualitative reading is unchanged from the heatmap, namely one NCL-specific candidate and four decoy-preferring sticky binders, but the per-bar annotation makes the magnitude of the difference visible and resistant to receptor-size confounding.

In order to provide further evidence whether these structural homologs are TIPalpha or Lpp20-leaning, Figure 4.15 reports the homodimer A-B interface across three cofolding contexts, namely the bait dimer alone (no third chain), the dimer in the presence of NCL (taking whichever bait form gives the higher A-B score), and the dimer in the presence of dsDNA-20. Per protein per panel two bars are shown, iLIS in blue and ipSAE in grey, both restricted to the A-B chain pair. Global confidence metrics (ipTM) are intentionally omitted because they score the entire complex rather than the interface and can therefore be high even when the A-B interface is barely formed. Twenty proteins are included (19 hCom2 carriers plus TIPalpha as a positive control). Three findings sharpen the AF3-axis verdict. First, TIPalpha is by some distance the strongest dimer in the dataset in the alone column (iLIS = 0.71, ipSAE = 0.74), so the published disulfide-locked TIPalpha homodimer is recovered cleanly by AF3 when the dimer is run without a receptor in the box. Second, *B. cellulosilyticus* 3007 is the only hCom2 carrier with a real if modest dimer-alone signal (iLIS = 0.245), and *B. stercoris* 2529, *B. plebeius* 2530, and *B. vulgatus* 0947 show essentially no dimer-alone signal but substantial dimer signal in the dim + dsDNA-20 panel ($iLIS_{AB}$ between 0.4 and 0.6), consistent with DNA-scaffolded rather than intrinsic homodimerization. Third, the same TIPalpha A-B sequence that gives iLIS = 0.71 alone drops to 0.55 with NCL and to 0.24 with dsDNA, the receptor-dependent dimer suppression noted above, so a low score in the right

two panels is not in itself evidence against intrinsic dimerization. The alone panel is therefore the cleaner reference for any intrinsic-dimer claim, and the right two panels are most useful for separating the DNA-scaffolded *Bacteroides* pattern from a TIPalpha-style intrinsic dimer.

4.6 Lessons going forward

The TIPalpha case is a worked example, and like any worked example it carries general lessons. I record them here because the same template (structural search, multi-reference re-alignment to triangulate within-clan family membership, principled co-folding) is the one I expect to be most reusable when other queries are run against the hCom2 structural proteome.

Fold-level recovery does not imply functional carryover

The commensal carriers I identify structurally are not flagged as virulence-associated by orthology or by sequence-level annotation, further suggesting that fold-level recovery of a TIPalpha-like topology in gut commensals does not gurantee functional overlap. TIPalpha drives proinflammatory and tumor-promoting activity in *H. pylori* through a disulfide-locked dimer plus nucleolin plus NF-κB axis, and the 19 carriers are searched against the truncated monomeric construct PDB 3GUQ that does not include the Cys25–Cys25 plus Cys27–Cys27 disulfide-bridge residues required for the *in vivo* mechanism (Suganuma et al., 2008; Watanabe et al., 2010; Suganuma et al., 2021; Gao et al., 2012). The verdict subsection additionally re-aligned each carrier against the full-length 3GIO crystal (also lacking Cys25 and Cys27 in the deposited model) and against the AFDB monomers of TIPalpha (UniProt O25318) and Lpp20 (UniProt P0A0V0). The AFDB monomer of O25318 is the full 192-residue prediction and does include Cys25 and Cys27 in primary sequence and predicted local structure, so the family-level verdict is informed by the residues that mediate the *in vivo* dimer-interface chemistry even though no monomeric reference can capture the inter-chain disulfide-bonded geometry per se.

The Lpp20-side framing carries an analogous caveat. Lpp20 in *H. pylori* has been reported to promote epithelial–mesenchymal transition in cultured gastric cells, on the basis of which Vallese and colleagues propose to include Lpp20 in the list of *H. pylori* virulence factors (Vallese et al., 2017), and Lpp20 is independently the immunodominant triacylated outer-membrane lipoprotein recognised by patient sera during infection and engaging TLR2 in a lipidation-dependent manner (Keenan et al., 2000; McClain et al., 2024; Jung et al., 2023). In commensal *Bacteroides*

the same fold sits in a body site (the gut lumen) and a host-cell context (intestinal epithelium under steady-state mucosal sampling) that does not parallel the *H. pylori* gastric-cancer pathobiology in which Lpp20 was assayed, so the EMT phenotype reported there is not by itself transferable to *Bacteroidota* carriers without direct experimental evidence in commensal settings.

The *Bacteroidota* and *Bacillota* carriers documented here should therefore not be read as latent virulence factors without direct biochemical or *in vivo* follow-up at the dimer-interface level on the TIPalpha side and at the EMT-induction level on the Lpp20 side. The published baseline against which to read these carriers is that gut commensal surface molecules routinely engage host innate immunity, often through TLR2 and often with beneficial rather than pathogenic outcomes. *B. fragilis* polysaccharide A directs maturation of host immune development and induces $Foxp3^{+}$ regulatory T cells through TLR2 signalling that protects against experimental colitis, and the *Akkermansia muciniphila* outer-membrane protein Amuc_1100 functions as a TLR2 ligand that improves intestinal-barrier function and metabolic markers in mouse models of obesity and diabetes (Mazmanian et al., 2005; Round and Mazmanian, 2010; Plovier et al., 2017). The hCom2 carriers in the PF02169-adjacent SHS2-clan neighbourhood are best read as candidate immunomodulatory surface molecules of gut commensals, where the direction of any host response (tolerogenic, homeostatic, or proinflammatory) is the experimental question and where the fold-level signal alone does not license a TIPalpha-style virulence interpretation.

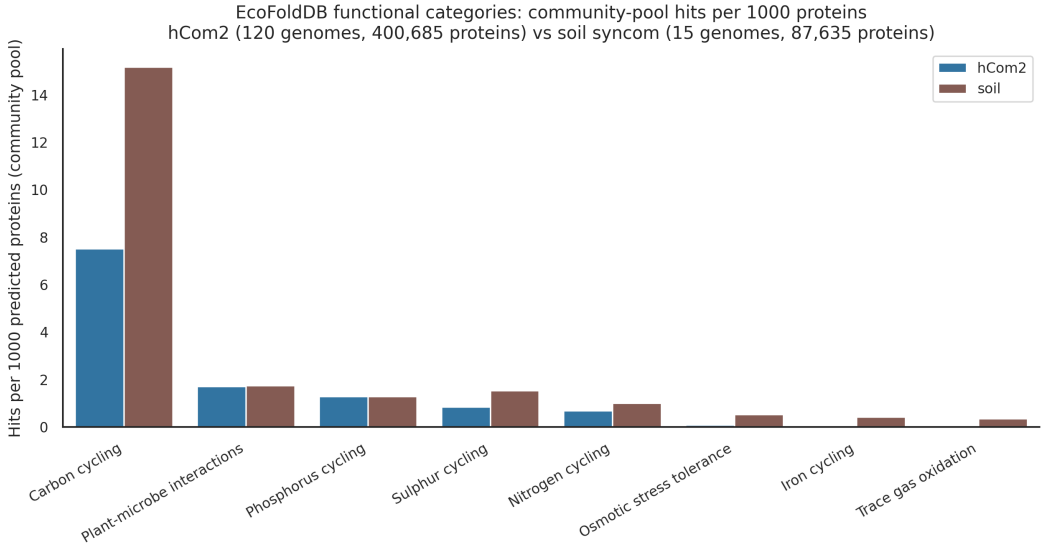
4.7 Looking ahead

First, this chapter is concretely a community-scale application of TRILL in conjuction with other methods. Every initial structure prediction in the proteome was issued through a TRILL `fold` command and run on the same hardware abstractions that Chapter 2 describes. The pattern is not unique to hCom2. Any defined microbial consortium for which translated open reading frames are available is a candidate for the same pipeline (ESMFold via TRILL, Merizo-Search against CATH-S40, functional-layer mapping, structural search as a discovery instrument), and the engineering cost of running the pipeline on a new consortium is dominated by GPU hours rather than by software-stack reassembly.

Lastly, the TIPalpha case is one worked example, not a recipe. The general pattern (structural search, multi-reference re-alignment to triangulate within-clan family membership) is reusable, but the biological reading is always specific to the query. A

virulence factor with a narrow Pfam scope and a structurally wide sister family will produce a different verdict structure than, say, a host-targeting glycoside hydrolase whose sequence-side family entry already spans multiple phyla, and the calibration of how much fold-level signal a community-scale search produces depends on where the query sits in the existing family-catalogue topology. The chapter does not attempt to enumerate query types. It offers one detailed worked case to serve as inspiration for others to leverage this structural resource.

Figure 4.7: EcoFoldDB community-pool hits per 1000 predicted proteins, by functional category, for hCom2 (120 genomes, 400,685 predicted proteins) versus the Coker soil syncom (15 genomes, 87,635 predicted proteins). Soil leads in every category except complex-carbohydrate-driven Carbon cycling. The largest soil enrichments per 1000 proteins are in niche environmental categories (trace-gas oxidation, iron cycling, osmotic-stress tolerance) that have no clear gut-physiological role.



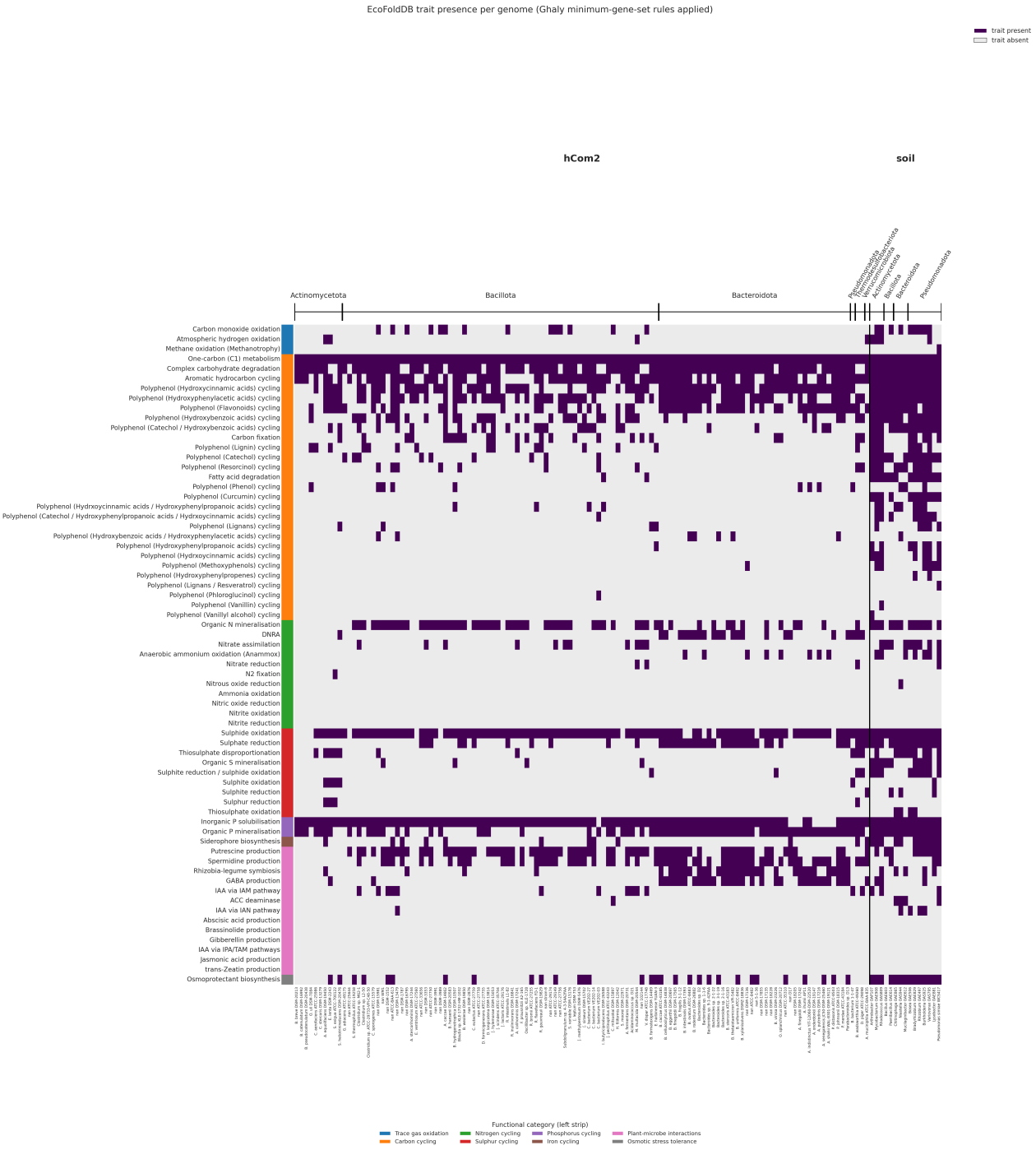


Figure 4.8: Per-genome trait presence across the 135 genomes (120 hCom2 plus 15 soil) and 67 EcoFoldDB sub-category traits, after applying the minimum-gene-set rules from Figures 3 and 4 of Ghaly et al. 2025 (purple cell = trait present, light cell = absent). Genomes are grouped by phylum within each community block. The cleanest gut-side enrichment in this view is dissimilatory nitrate reduction to ammonium (DNRA, nrfA), positive in 23 of 120 hCom2 and 0 of 15 soil. The soil-side ACC-deaminase enrichment is concentrated in 3 of 4 acdS-positive soil strains being *Bacteroidota*, recapitulating the central finding of Ghaly et al.

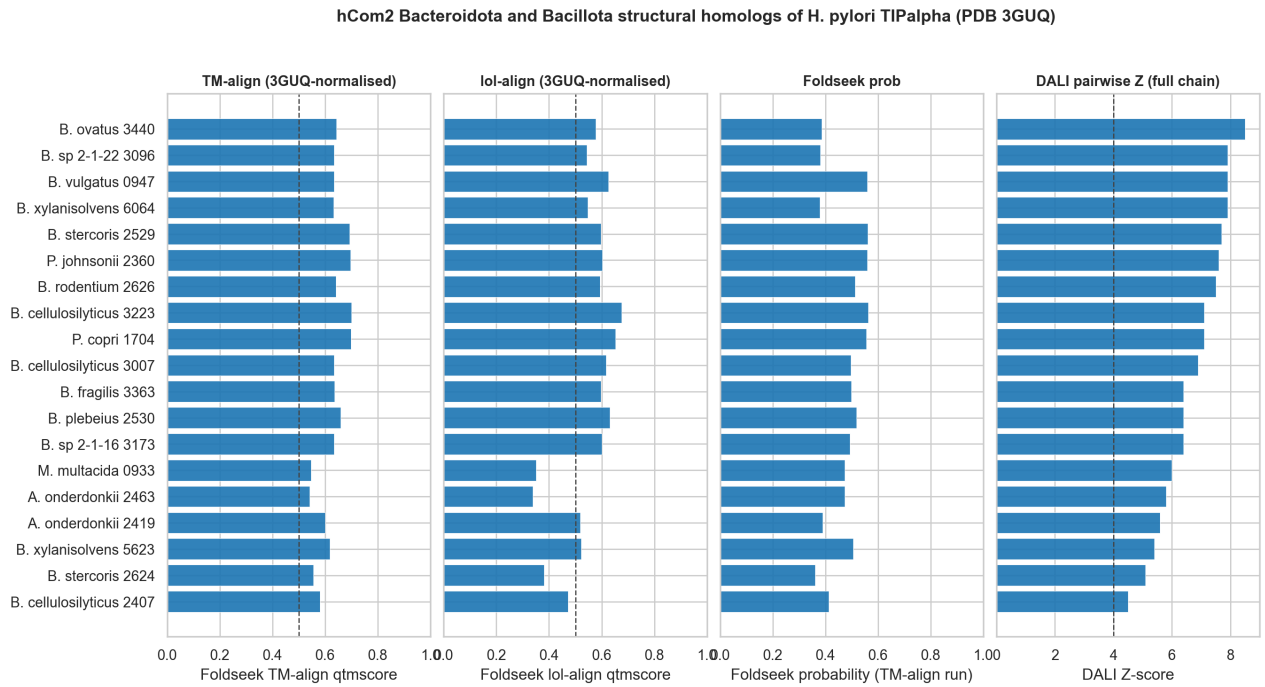
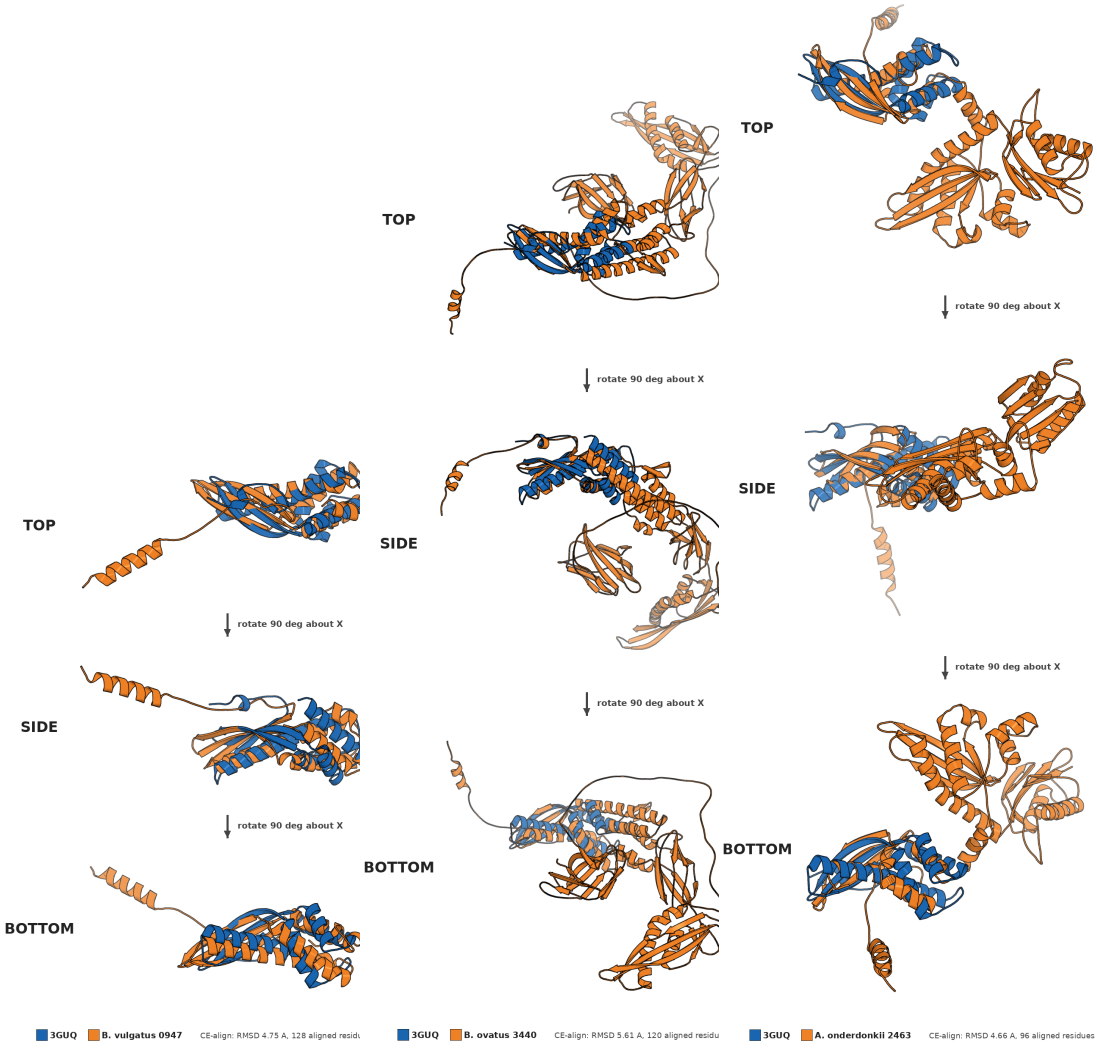


Figure 4.9: Four-metric structural-search summary for the 19 hCom2 carriers of CATH 3.10.129.140 against the *H. pylori* TIPalpha reference structure PDB 3GUQ (18 *Bacteroidota* plus one *Bacillota Mitsuokella multacida* 0933). From left to right, Foldseek TM-align qtmscore, Foldseek LoL-Align qtmscore, Foldseek probability score from the TM-align run, and DALI pairwise Z-score. Bars are sorted by DALI Z descending. Dashed lines mark the canonical TM = 0.5 fold-level cutoff and the DALI Z = 2 noise floor for similar folds. The *Mitsuokella* entry and the *A. onderdonkii* 2463 entry in both the TM-align and LoL-Align panels were recovered by re-running Foldseek with `-alignment-type 1 -exhaustive-search 1` after the default 3Di prefilter excluded them.



(a) *P. vulgatus* 0947 (Z 7.9) (b) *B. ovatus* 3440 (Z 8.5) (c) *A. onderdonkii* 2463 (Z 5.8)

Figure 4.10: Three representative pairwise DALI superpositions of hCom2 *Bacteroidota* carriers (colour) onto the *H. pylori* TIPalpha reference PDB 3GUQ (grey), each shown as a three-view rotation panel. (a) *P. vulgatus* 0947 at DALI Z 7.9 at 9 percent sequence identity, TM-align qtmscore 0.634. (b) *B. ovatus* 3440 is the highest-scoring carrier in the dataset by DALI Z and one of four carriers flagged by InterProScan as a Pfam PF02169 (Lpp20 lipoprotein) family member, the same family as the *H. pylori* TIPalpha paralog Lpp20. DALI Z 8.5 at 10 percent sequence identity, TM-align qtmscore 0.643. (c) *A. onderdonkii* 2463, one of the two carriers recovered only after Foldseek was re-run with `-alignment-type 1 -exhaustive-search 1`. DALI Z 5.8 at 8 percent sequence identity, TM-align qtmscore 0.540.

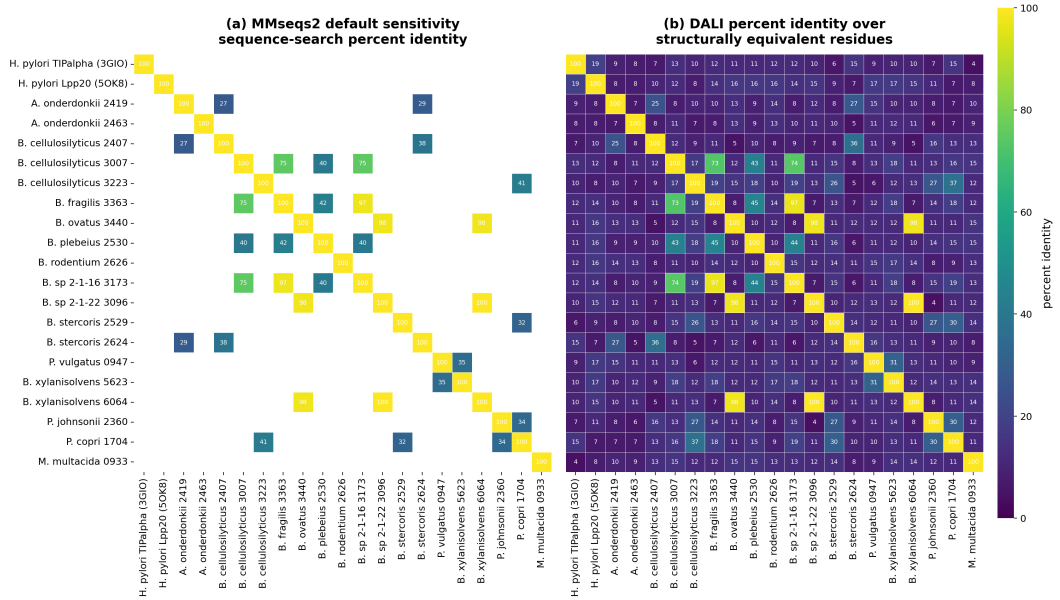


Figure 4.11: Sequence-conservation views of the 21-input set (full-length *H. pylori* TIPalpha at 192 aa with structural reference 3GIO, full-length *H. pylori* Lpp20 at 175 aa with structural reference 5OK8, and the 19 hCom2 carriers). Rows and columns are ordered taxonomically with the two *H. pylori* references first, the 18 *Bacteroidota* carriers grouped by genus next, and the *Bacillota Mitsuokella multacida* 0933 last. (a) MMseqs2 easy-search all-vs-all at default settings, BLOSUM62 scoring, `pident` reported as matches divided by alignment length. Grey cells are pairs for which MMseqs2 returns no alignment. At default sensitivity neither reference row returns any alignment to any carrier, that is 0 of 38 reference-vs-carrier pairs align at the sequence level. (b) DaliLite v5 pairwise alignment percent identity computed only over the structurally aligned residues. The colour scale is shared between panels.

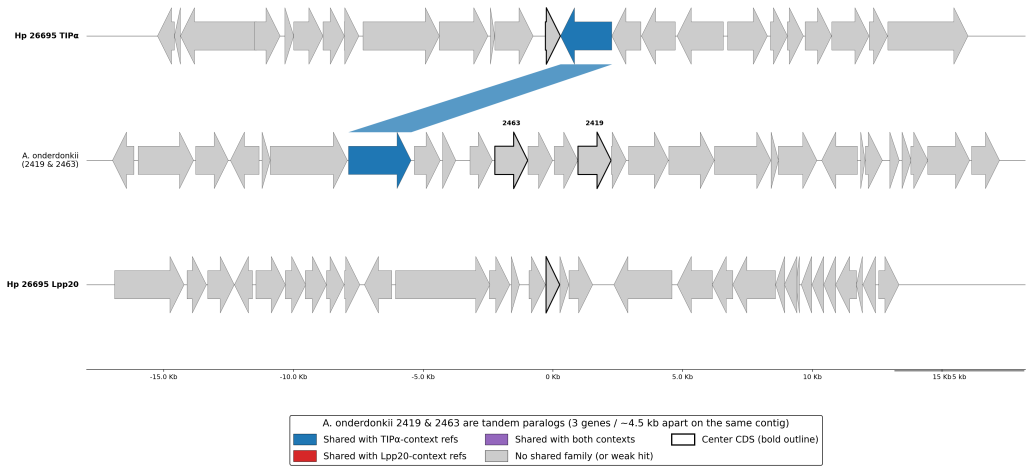


Figure 4.12: Three-track ±15 kb synteny comparison of *Alistipes onderdonkii* DSM-19147 carriers 2419 and 2463 (middle track, both shown as bold-outlined centre CDSs three genes apart) against the canonical *H. pylori* 26695 TIPalpha locus (top) and Lpp20 locus (bottom). Each arrow is a Prodigal-called CDS. Arrow direction encodes coding strand. Coding sequences are coloured by Pfam/NCBIfam/HAMAP family content relative to the two *H. pylori* reference contexts, with **blue** for a family shared with a TIPalpha-context reference window, **red** for a family shared with a Lpp20-context reference window, **purple** for a family shared with both, and **grey** for no shared family (or a hit weaker than the strict cross-context cutoff $E < 10^{-5}$). The single blue CDS upstream of the *A. onderdonkii* carriers is the Class A bifunctional penicillin-binding protein (Pfam PF00912 + PF00905 + PF28500) that is structurally analogous to *pbp1a* (HP0597) immediately downstream of TIPalpha in *H. pylori*, and the ribbon links the two matching CDSs. The Lpp20 reference track (bottom) is entirely grey, reflecting the chapter-wide negative finding that no carrier exhibits operon-level Lpp20-context family sharing.

Figure 4.13: iLIS heatmap of the AF3 cofolding screen. Rows are the 21 inputs (19 hCom2 carriers plus *H. pylori* TIPalpha and Lpp20 references) at each of two stoichiometries (monomer and homodimer, 42 rows total). Columns are the nine receptor or receptor-form combinations on file, namely NCL_mat, NCL_full, dsDNA20, the two RRM-fold controls U2AF2-RRMs and hnRNPA1-UP1, and four broader sticky-surface controls CALR, ANXA2, LGALS3-CRD, and MBP. Each cell is the best iLIS for the relevant chain pair in that AF3 job, sorted top-to-bottom by per-row maximum iLIS so any input with signal floats to the top. iLIS is the geometric mean of LIS and a contact-restricted variant cLIS, both computed at PAE $\leq$ 12 Å (cLIS additionally requires Cβ-Cβ contact within 8 Å) (Kim et al., 2024). The dsDNA-20 column is reported under the same colour scale but should be read as "AF3 finds an interface" rather than "AF3 finds a base-specific contact," since DNA carries no Cβ atom for the cLIS filter. The 0.223 published Y2H-reference cutoff is intentionally not annotated for the reasons given in the main text.

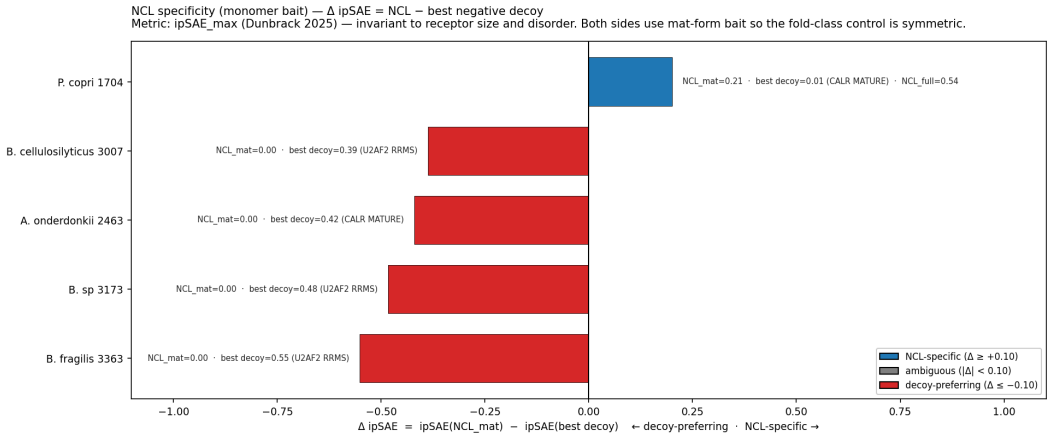


Figure 4.14: Per-protein NCL-specificity delta, ΔipSAE = ipSAE(NCL_mat) − ipSAE(best decoy), for the monomer-form baits that pass a minimum-signal filter (max of NCL_mat, NCL_full, and best decoy ipSAE at least 0.05). Positive Δ (blue, right) is NCL-specific. Negative Δ (red, left) is decoy-preferring. Each bar is annotated to the right with the raw NCL_mat ipSAE, the best-decoy ipSAE with the decoy receptor named, and the full-form NCL ipSAE where it exists (the latter for context only, since no full-form decoy panel is on file). The decoy panel is restricted to mat-form bait against six controls (U2AF2-RRMs, hnRNPA1-UP1, CALR, ANXA2, LGALS3-CRD, MBP) to keep the fold-class comparison symmetric. ipSAE (Dunbrack, 2025) is used rather than iLIS because ipSAE's d_0 normalisation scales with the structured-interface size and is therefore invariant to receptor-chain length and disordered-tail extent, a property iLIS does not strictly share. Sixteen baits with no detectable signal anywhere on the receptor panel are excluded for clarity. The companion dim-form delta is empty at PAE ≤ 10 (every dim_mat NCL ipSAE is exactly zero and every dim_mat decoy ipSAE is below 0.05) and is not shown.

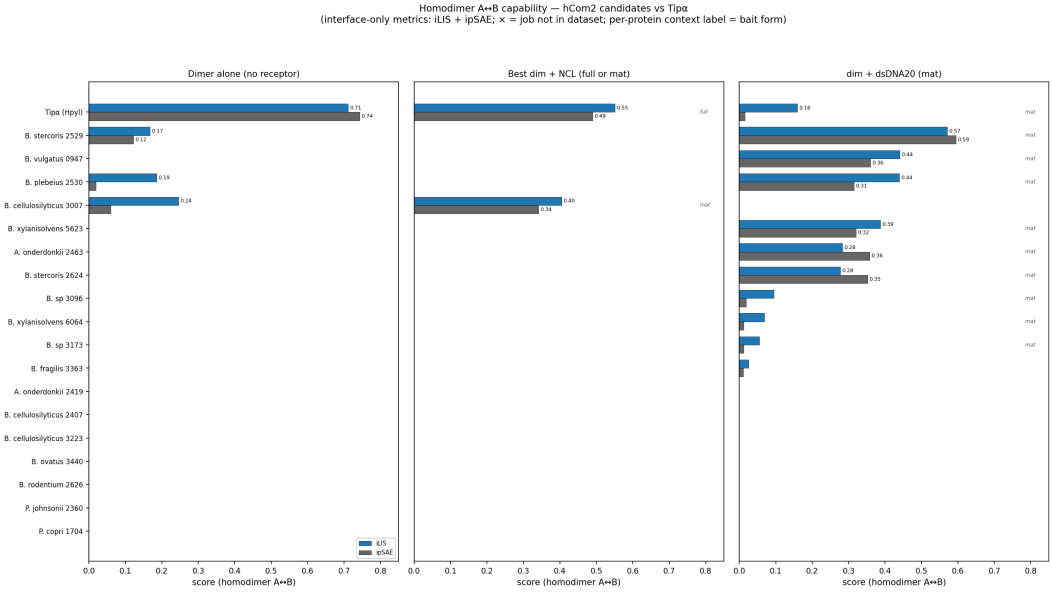


Figure 4.15: Per-protein homodimer A-B interface confidence across three AF3 cofolding contexts. **Left**. Dimer alone (chains A and B only, no third chain). **Middle**. Dimer in the presence of NCL, taking the higher A-B score across the full-form and mat-form bait runs. **Right**. Dimer in the presence of dsDNA-20 (mat-form bait). Twenty proteins are shown (19 hCom2 carriers plus *H. pylori* TIPalpha as a positive control), with Lpp20 omitted because no dimer-alone AF3 job has been run. Per protein in each panel, two bars are plotted, iLIS in blue (primary) and ipSAE in grey (secondary), both restricted to the A-B chain pair. The bait form (mat or full) that produced the bar in the right two panels is annotated to the right of each row, since the same bait can give substantially different dimer signals at different forms (TIPalpha dim_full + NCL gives iLIS_{AB} = 0.55 versus dim_mat + NCL = 0.11). Global confidence metrics (ipTM, actifpTM) are deliberately not shown because they score the entire complex and can therefore be high even when the A-B interface is barely formed. Dimer-alone scores are from the rank-0 model only and are not cross-ranked across the AF3 ensemble.

Chapter 5

CONCLUDING REMARKS

The body chapters of this thesis are not all about TRILL, and they were not built primarily as demonstrations of it. What unifies the three chapters is the declarative posture rather than a uniform reliance on a single piece of software.

The first key condition for TRILL is that the platform's seams are sharp enough to be recombined by people who did not build it. Standardised file formats and small per-command argument surfaces are easy to undervalue at the time they are written, but their payoff is that an external user can compose a workflow the original developer never anticipated. The cleanest in-thesis evidence is Foldtuning, where Subramanian and colleagues built an iterative biomimetic-design loop out of TRILL's existing commands using only the documented command-line interface (Subramanian et al., 2023) with minimal glue code needed. The second is the spider-threat pipeline of Chapter 3, which threads ten TRILL commands end to end from NR-scale embedding through molecular dynamics. The condition is not that the platform supports many models, but that the joints between models are thin and stable enough to invite recombination.

Furthermore, abstraction should not ask the user to take deep-learning verdicts as final. A pattern recurs across the application chapters, in which a fast learned screen is paired with a physics-based check-out at the point in the pipeline where the cost of overconfidence is highest. The spider-threat binders go through molecular dynamics and subsequent ProLIF interaction fingerprints calculations, and decomposition of RMSD. Without this pairing, a declarative interface that hides operational detail also hides the moments at which a deep-learning prediction has run beyond what its training distribution allows. With it, end-users can have more confidence in their final predictions.

These technical conditions let TRILL run, but they do not by themselves let it reach the populations the abstraction was built for. That is the work of permissive licensing, no required API keys, fully local execution, public repositories, and preprint-first publication. The deeper point is that TRILL would not exist without the upstream open-source ecosystem on which it depends, including PyTorch Lightning, HuggingFace Transformers, OpenMM, Foldseek, ESMFold, RFDiffusion, and

Merizo among many others, each of which is itself a community gift, and a closed downstream artifact built on an open upstream substrate violates the terms that made it possible in the first place.

Each era of biology has had to climb a rung. Pairwise sequence alignment gave way to learned residue-level representations, those representations gave way to predicted three-dimensional structures, and individual structures are now beginning to give way to community-scale structural proteomes of defined consortia. Declarative interfaces are one of this era's climb. They are not a replacement for any prior rung, rather they are the arrangement under which a non-specialist can stand on top of all of them at once and ask a question whose answer would otherwise have required a different specialist for each layer. The climb is not finished, but the rung the next scientist might need to to stand on is the one this work has tried to put in place as a prototype.

BIBLIOGRAPHY

Abramson, Josh et al. (June 13, 2024). "Accurate structure prediction of biomolecular interactions with AlphaFold 3". In: *Nature* 630.8016, pp. 493–500. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-024-07487-w.

Akiba, Takuya et al. (July 25, 2019). "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19: The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Anchorage AK USA: ACM, pp. 2623–2631. ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330701.

Allen, James W. A. et al. (Jan. 29, 2003). "C-type cytochromes: diverse structures and biogenesis systems pose evolutionary problems". In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 358.1429. Ed. by J. F. Allen and J. A. Raven, pp. 255–266. ISSN: 0962-8436, 1471-2970. DOI: 10.1098/rstb.2002.1192.

Alley, Ethan C. et al. (Dec. 2019). "Unified rational protein engineering with sequence-based deep representation learning". In: *Nature Methods* 16.12, pp. 1315–1322. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0598-1.

Altschul, Stephen F. et al. (Oct. 1990). "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3, pp. 403–410. ISSN: 00222836. DOI: 10.1016/S0022-2836(05)80360-2.

Argyrou, Argyrides and John S. Blanchard (2004). "Flavoprotein Disulfide Reductases: Advances in Chemistry and Function". In: *Progress in Nucleic Acid Research and Molecular Biology*. Vol. 78. Elsevier, pp. 89–142. ISBN: 978-0-12-540078-7. DOI: 10.1016/S0079-6603(04)78003-4.

Armbruster, Krista M. et al. (Nov. 12, 2024). "Identification and characterization of the lipoprotein *N* -acyltransferase in *Bacteroides*". In: *Proceedings of the National Academy of Sciences* 121.46, e2410909121. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2410909121.

Austin, Michael B. and Joseph P. Noel (Jan. 21, 2003). "The chalcone synthase superfamily of type III polyketide synthases". In: *Natural Product Reports* 20.1, pp. 79–110. ISSN: 02650568, 14604752. DOI: 10.1039/b100917f.

Barrio-Hernandez, Inigo et al. (Oct. 19, 2023). "Clustering predicted structures at the scale of the known protein universe". In: *Nature* 622.7983, pp. 637–645. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06510-w.

Benjamini, Yoav and Yosef Hochberg (Jan. 1, 1995). "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". In: *Journal of the*

Royal Statistical Society Series B: Statistical Methodology 57.1, pp. 289–300. ISSN: 1369-7412, 1467-9868. DOI: 10.1111/j.2517-6161.1995.tb02031.x.

Berendsen, Roeland L. et al. (Dec. 2015). "Unearthing the genomes of plant-beneficial Pseudomonas model strains WCS358, WCS374 and WCS417". In: *BMC Genomics* 16.1, p. 539. ISSN: 1471-2164. DOI: 10.1186/s12864-015-1632-z.

Bernhofer, Michael and Burkhard Rost (Aug. 8, 2022). "TMbed: transmembrane proteins predicted through language model embeddings". In: *BMC Bioinformatics* 23.1, p. 326. ISSN: 1471-2105. DOI: 10.1186/s12859-022-04873-x.

Besse, Alison et al. (Nov. 2015). "Antimicrobial peptides and proteins in the face of extremes: Lessons from archaeocins". In: *Biochimie* 118, pp. 344–355. ISSN: 03009084. DOI: 10.1016/j.biochi.2015.06.004.

Bloudoff, Kristjan and T. Martin Schmeing (Nov. 2017). "Structural and functional aspects of the nonribosomal peptide synthetase condensation domain superfamily: discovery, dissection and diversity". In: *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* 1865.11, pp. 1587–1604. ISSN: 15709639. DOI: 10.1016/j.bbapap.2017.05.010.

Bodelón, Gustavo, Carmen Palomino, and Luis Ángel Fernández (Mar. 2013). "Immunoglobulin domains in *Escherichia coli* and other enterobacteria: from pathogenesis to applications in antibody technologies". In: *FEMS Microbiology Reviews* 37.2, pp. 204–250. ISSN: 1574-6976. DOI: 10.1111/j.1574-6976.2012.00347.x.

Boorla, Veda Sheersh and Costas D. Maranas (Feb. 28, 2025). "CatPred: a comprehensive framework for deep learning in vitro enzyme kinetic parameters". In: *Nature Communications* 16.1, p. 2072. ISSN: 2041-1723. DOI: 10.1038/s41467-025-57215-9.

Bordin, Nicola et al. (Feb. 8, 2023). "AlphaFold2 reveals commonalities and novelties in protein structure space for 21 model organisms". In: *Communications Biology* 6.1, p. 160. ISSN: 2399-3642. DOI: 10.1038/s42003-023-04488-9.

Bordon, Karla C.F. et al. (Dec. 2012). "Isolation, enzymatic characterization and antiedematogenic activity of the first reported rattlesnake hyaluronidase from Crotalus durissus terrificus venom". In: *Biochimie* 94.12, pp. 2740–2748. ISSN: 03009084. DOI: 10.1016/j.biochi.2012.08.014.

Bouysset, Cédric and Sébastien Fiorucci (Dec. 2021). "ProLIF: a library to encode molecular interactions as fingerprints". In: *Journal of Cheminformatics* 13.1, p. 72. ISSN: 1758-2946. DOI: 10.1186/s13321-021-00548-6.

Burley, Stephen K et al. (Jan. 6, 2025). "Updated resources for exploring experimentally-determined PDB structures and Computed Structure Models at the RCSB Protein Data Bank". In: *Nucleic Acids Research* 53 (D1), pp. D564–D574. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkae1091.

Capitani, Nagaja et al. (May 2019). “The lipoprotein HP1454 of *Helicobacter pylori* regulates T -cell response by shaping T -cell receptor signalling”. In: *Cellular Microbiology* 21.5, e13006. ISSN: 1462-5814, 1462-5822. DOI: 10.1111/cmi.13006.

Case, David A. et al. (Oct. 23, 2023). “AmberTools”. In: *Journal of Chemical Information and Modeling* 63.20, pp. 6183–6191. ISSN: 1549-9596, 1549-960X. DOI: 10.1021/acs.jcim.3c01153.

Chai Discovery et al. (Oct. 11, 2024). *Chai-1: Decoding the molecular interactions of life*. DOI: 10.1101/2024.10.10.615955.

Chaim, Olga M. et al. (Feb. 2011). “Phospholipase-D activity and inflammatory response induced by brown spider dermonecrotic toxin: Endothelial cell membrane phospholipids as targets for toxicity”. In: *Biochimica et Biophysica Acta (BBA) - Molecular and Cell Biology of Lipids* 1811.2, pp. 84–96. ISSN: 13881981. DOI: 10.1016/j.bbalip.2010.11.005.

Charon, Marie–Helene et al. (Feb. 1, 1999). “[No title found]”. In: *Nature Structural Biology* 6.2, pp. 182–190. ISSN: 10728368. DOI: 10.1038/5870.

Chen, Jiayang et al. (Aug. 7, 2022). *Interpretable RNA Foundation Model from Unannotated Data for Highly Accurate RNA Structure and Function Predictions*. DOI: 10.1101/2022.08.06.503062.

Chen, Tianqi and Carlos Guestrin (Aug. 13, 2016). “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco California USA: ACM, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785.

Cheng, Alice G. et al. (Sept. 2022). “Design, construction, and in vivo augmentation of a complex gut microbiome”. In: *Cell* 185.19, 3617–3636.e19. ISSN: 00928674. DOI: 10.1016/j.cell.2022.08.003.

Chu, Lee-Shin et al. (Feb. 2024). “Flexible protein–protein docking with a multitrack iterative transformer”. In: *Protein Science* 33.2, e4862. ISSN: 0961-8368, 1469-896X. DOI: 10.1002/pro.4862.

Clough, Emily et al. (Jan. 5, 2024). “NCBI GEO: archive for gene expression and epigenomics data sets: 23-year update”. In: *Nucleic Acids Research* 52 (D1), pp. D138–D144. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkad965.

Cock, Peter J. A. et al. (June 1, 2009). “Biopython: freely available Python tools for computational molecular biology and bioinformatics”. In: *Bioinformatics* 25.11, pp. 1422–1423. ISSN: 1367-4811, 1367-4803. DOI: 10.1093/bioinformatics/btp163.

Coker, Joanna et al. (Dec. 20, 2022). "A Reproducible and Tunable Synthetic Soil Microbial Community Provides New Insights into Microbial Ecology". In: *mSystems* 7.6. Ed. by Ryan McClure, e00951–22. ISSN: 2379-5077. DOI: 10.1128/msystems.00951-22.

Corso, Gabriele et al. (2022). *DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking*. Version Number: 2. DOI: 10.48550/ARXIV.2210.01776.

Costa, Luciana F. et al. (Dec. 2016). "Iron acquisition pathways and colonization of the inflamed intestine by Salmonella enterica serovar Typhimurium". In: *International Journal of Medical Microbiology* 306.8, pp. 604–610. ISSN: 14384221. DOI: 10.1016/j.ijmm.2016.10.004.

Coyne, Michael J. et al. (Mar. 18, 2005). "Human Symbionts Use a Host-Like Pathway for Surface Fucosylation". In: *Science* 307.5716, pp. 1778–1781. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1106469.

Dallago, Christian et al. (Nov. 11, 2021a). *FLIP: Benchmark tasks in fitness landscape inference for proteins*. DOI: 10.1101/2021.11.09.467890.

Dallago, Christian et al. (May 2021b). "Learned Embeddings from Deep Learning to Visualize and Predict Protein Sets". In: *Current Protocols* 1.5, e113. ISSN: 2691-1299, 2691-1299. DOI: 10.1002/cpz1.113.

Dauparas, J. et al. (Oct. 7, 2022). "Robust deep learning–based protein sequence design using ProteinMPNN". In: *Science* 378.6615, pp. 49–56. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.add2187.

Dauparas, Justas et al. (Apr. 2025). "Atomic context-conditioned protein sequence design using LigandMPNN". In: *Nature Methods* 22.4, pp. 717–723. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-025-02626-1.

Deshpande, Aditi et al. (Mar. 4, 2022). *The Ferrous Iron Transporter FeoB1 is Essential for Clostridioides difficile Toxin Production and Pathogenesis in Mice*. DOI: 10.1101/2022.03.03.482942.

Dhananjaya, B. L. and C. J. M. D'souza (Jan. 2010). "An overview on nucleases (DNase, RNase, and phosphodiesterase) in snake venoms". In: *Biochemistry (Moscow)* 75.1, pp. 1–6. ISSN: 0006-2979, 1608-3040. DOI: 10.1134/S0006297910010013.

Dror, Barak et al. (Dec. 22, 2020). "Elucidating the Diversity and Potential Function of Nonribosomal Peptide and Polyketide Biosynthetic Gene Clusters in the Root Microbiome". In: *mSystems* 5.6. Ed. by Marnix Medema, e00866–20. ISSN: 2379-5077. DOI: 10.1128/mSystems.00866-20.

Dunbrack, Roland L. (Feb. 14, 2025). *Rēs ipSAE loquuntur : What's wrong with AlphaFold's ipTM score and how to fix it*. DOI: 10.1101/2025.02.10.637595.

Eastman, Peter et al. (July 26, 2017). "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics". In: *PLOS Computational Biology* 13.7. Ed. by Robert Gentleman, e1005659. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005659.

Eastman, Peter et al. (Jan. 11, 2024). "OpenMM 8: Molecular Dynamics Simulation with Machine Learning Potentials". In: *The Journal of Physical Chemistry B* 128.1, pp. 109–116. ISSN: 1520-6106, 1520-5207. DOI: 10.1021/acs.jpcb.3c06662.

Eberhardt, Jerome et al. (Aug. 23, 2021). "AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings". In: *Journal of Chemical Information and Modeling* 61.8, pp. 3891–3898. ISSN: 1549-9596, 1549-960X. DOI: 10.1021/acs.jcim.1c00203.

Eddy, Sean R. (Oct. 20, 2011). "Accelerated Profile HMM Searches". In: *PLoS Computational Biology* 7.10. Ed. by William R. Pearson, e1002195. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1002195.

Edwards, Carl et al. (2022). "Translation between Molecules and Natural Language". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 375–413. DOI: 10.18653/v1/2022.emnlp-main.26.

Elnaggar, Ahmed et al. (Oct. 1, 2022). "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10, pp. 7112–7127. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2021.3095381.

Elnaggar, Ahmed et al. (2023). *Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling*. Version Number: 1. DOI: 10.48550/ARXIV.2301.06568.

Eramian, David et al. (Nov. 2008). "How well can the accuracy of comparative protein structure models be predicted?" In: *Protein Science* 17.11, pp. 1881–1893. ISSN: 0961-8368, 1469-896X. DOI: 10.1110/ps.036061.108.

Falcon, William et al. (May 15, 2020). *PyTorchLightning/pytorch-lightning: 0.7.6 release*. Version 0.7.6. DOI: 10.5281/ZENODO.3828935.

Fernández-Tornero, Carlos et al. (Dec. 1, 2001). "[No title found]". In: *Nature Structural Biology* 8.12, pp. 1020–1024. ISSN: 10728368. DOI: 10.1038/nsb724.

Ferruz, Noelia, Steffen Schmidt, and Birte Höcker (July 27, 2022). "ProtGPT2 is a deep unsupervised language model for protein design". In: *Nature Communications* 13.1, p. 4348. ISSN: 2041-1723. DOI: 10.1038/s41467-022-32007-7.

Fleming, Jennifer et al. (Aug. 2025). "AlphaFold Protein Structure Database and 3D-Beacons: New Data and Capabilities". In: *Journal of Molecular Biology* 437.15, p. 168967. ISSN: 00222836. DOI: 10.1016/j.jmb.2025.168967.

Francia, M. Victoria et al. (Feb. 2004). "A classification scheme for mobilization regions of bacterial plasmids". In: *FEMS Microbiology Reviews* 28.1, pp. 79–100. ISSN: 1574-6976. DOI: 10.1016/j.femsre.2003.09.001.

Fujita, Yasutaro, Hiroshi Matsuoka, and Kazutake Hirooka (Nov. 2007). "Regulation of fatty acid metabolism in bacteria". In: *Molecular Microbiology* 66.4, pp. 829–839. ISSN: 0950-382X, 1365-2958. DOI: 10.1111/j.1365-2958.2007.05947.x.

Gado, Japheth E. et al. (June 22, 2023). *Machine learning prediction of enzyme optimum pH*. DOI: 10.1101/2023.06.22.544776.

Ganz, Tomas (Sept. 2003). "Defensins: antimicrobial peptides of innate immunity". In: *Nature Reviews Immunology* 3.9, pp. 710–720. ISSN: 1474-1733, 1474-1741. DOI: 10.1038/nri1180.

Gao, Linyi et al. (Aug. 2017). "Engineered Cpf1 variants with altered PAM specificities". In: *Nature Biotechnology* 35.8, pp. 789–792. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt.3900.

Gao, Mingming et al. (July 31, 2012). "Crystal Structure of TNF-α-Inducing Protein from Helicobacter Pylori in Active Form Reveals the Intrinsic Molecular Flexibility for Unique DNA-Binding". In: *PLoS ONE* 7.7. Ed. by Anthony George, e41871. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0041871.

Gao, Rong and Ann M. Stock (Oct. 1, 2009). "Biological Insights from Structures of Two-Component Proteins". In: *Annual Review of Microbiology* 63.1, pp. 133–154. ISSN: 0066-4227, 1545-3251. DOI: 10.1146/annurev.micro.091208.073214.

Ghaly, Timothy M. et al. (Sept. 2025). "EcoFoldDB : Protein Structure-Guided Functional Profiling of Ecologically Relevant Microbial Traits at the Metagenome Scale". In: *Environmental Microbiology* 27.9, e70178. ISSN: 1462-2912, 1462-2920. DOI: 10.1111/1462-2920.70178.

Glass, Magdalena et al. (Apr. 2015). "Endo-β-1,4-glucanases impact plant cell wall development by influencing cellulose crystallization". In: *Journal of Integrative Plant Biology* 57.4, pp. 396–410. ISSN: 1672-9072, 1744-7909. DOI: 10.1111/jipb.12353.

Gowers, Richard et al. (2016). "MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations". In: Python in Science Conference. Austin, Texas, pp. 98–105. DOI: 10.25080/Majora-629e541a-00e.

Greule, Anja et al. (2018). "Unrivalled diversity: the many roles and reactions of bacterial cytochromes P450 in secondary metabolism". In: *Natural Product Reports* 35.8, pp. 757–791. ISSN: 0265-0568, 1460-4752. DOI: 10.1039/C7NP00063D.

Grilli, J. et al. (Jan. 2012). "Joint scaling laws in functional and evolutionary categories in prokaryotic genomes". In: *Nucleic Acids Research* 40.2, pp. 530–540. ISSN: 1362-4962, 0305-1048. DOI: 10.1093/nar/gkr711.

Grindley, Nigel D.F., Katrine L. Whiteson, and Phoebe A. Rice (June 1, 2006). "Mechanisms of Site-Specific Recombination". In: *Annual Review of Biochemistry* 75.1, pp. 567–605. ISSN: 0066-4154, 1545-4509. DOI: 10.1146/annurev.biochem.73.011303.073908.

Grondin, Julie M. et al. (Aug. 2017). "Polysaccharide Utilization Loci: Fueling Microbial Communities". In: *Journal of Bacteriology* 199.15. Ed. by George O'Toole. ISSN: 0021-9193, 1098-5530. DOI: 10.1128/JB.00860-16.

Grossfield, Alan and Daniel M. Zuckerman (2009). "Chapter 2 Quantifying Uncertainty and Sampling Quality in Biomolecular Simulations". In: *Annual Reports in Computational Chemistry*. Vol. 5. Elsevier, pp. 23–48. ISBN: 978-0-444-53359-3. DOI: 10.1016/S1574-1400(09)00502-7.

Gugger, Sylvain et al. (2022). *Accelerate: Training and inference at scale made simple, efficient and adaptable.*

Hager, Natalie et al. (Mar. 2026). "Distribution and activity of nitrate and nitrite reductases in the microbiota of the human intestinal tract". In: *The FEBS Journal* 293.6, pp. 1624–1642. ISSN: 1742-464X, 1742-4658. DOI: 10.1111/febs.70299.

Haiser, Henry J. et al. (July 19, 2013). "Predicting and Manipulating Cardiac Drug Inactivation by the Human Gut Bacterium *Eggerthella lenta*". In: *Science* 341.6143, pp. 295–298. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1235872.

Han, So-Ra et al. (Nov. 22, 2023). "Evidential deep learning for trustworthy prediction of enzyme commission number". In: *Briefings in Bioinformatics* 25.1, bbad401. ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/bbad401.

Hauenstein, Julia et al. (Jan. 6, 2026). "BRENDA in 2026: a Global Core Biodata Resource for functional enzyme and metabolic data within the DSMZ Digital Diversity". In: *Nucleic Acids Research* 54 (D1), pp. D527–D534. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkaf1113.

Heinzinger, Michael et al. (Sept. 28, 2024). "Bilingual language model for protein sequence and structure". In: *NAR Genomics and Bioinformatics* 6.4, lqae150. ISSN: 2631-9268. DOI: 10.1093/nargab/lqae150.

Ho, Y.-S. J. (Oct. 16, 2000). "Structure of the GAF domain, a ubiquitous signaling motif and a new class of cyclic GMP receptor". In: *The EMBO Journal* 19.20, pp. 5288–5299. ISSN: 14602075. DOI: 10.1093/emboj/19.20.5288.

Holder, Jason W. et al. (Sept. 8, 2011). "Comparative and Functional Genomics of Rhodococcus opacus PD630 for Biofuels Development". In: *PLoS Genetics* 7.9. Ed. by Paul M. Richardson, e1002219. ISSN: 1553-7404. DOI: 10.1371/journal.pgen.1002219.

Holm, Liisa (Jan. 2020). "DALI and the persistence of protein shape". In: *Protein Science* 29.1, pp. 128–140. ISSN: 0961-8368, 1469-896X. DOI: 10.1002/pro.3749.

Holm, Liisa (July 5, 2022). “Dali server: structural unification of protein families”. In: *Nucleic Acids Research* 50 (W1), W210–W215. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkac387.

Holm, Liisa et al. (Jan. 2023). “DALI shines a light on remote homologs: One hundred discoveries”. In: *Protein Science* 32.1, e4519. ISSN: 0961-8368, 1469-896X. DOI: 10.1002/pro.4519.

Hopf, Thomas A. et al. (Mar. 19, 2026). *evedesign: accessible biosequence design with a unified framework*. DOI: 10.64898/2026.03.17.712115.

Hsu, Chloe et al. (Apr. 10, 2022). *Learning inverse folding from millions of predicted structures*. DOI: 10.1101/2022.04.10.487779.

IBM Research and Indra Priyadarsini (2026). “materials.selfies-ted”. In: (). Version Number: 55e8339. DOI: 10.57967/HF/6690.

Illergård, Kristoffer, David H. Ardell, and Arne Elofsson (Nov. 15, 2009). “Structure is three to ten times more conserved than sequence—A study of structural response in protein cores”. In: *Proteins: Structure, Function, and Bioinformatics* 77.3, pp. 499–508. ISSN: 0887-3585, 1097-0134. DOI: 10.1002/prot.22458.

Ito, Takeshi et al. (Feb. 25, 2020). “Genetic and Biochemical Analysis of Anaerobic Respiration in Bacteroides fragilis and Its Importance *In Vivo*”. In: *mBio* 11.1. Ed. by Derek R. Lovley, e03238–19. ISSN: 2161-2129, 2150-7511. DOI: 10.1128/mBio.03238-19.

Jang, Jun Young et al. (Sept. 2009). “Crystal Structure of the TNF-α-Inducing Protein (Tipα) from Helicobacter pylori: Insights into Its DNA-Binding Activity”. In: *Journal of Molecular Biology* 392.1, pp. 191–197. ISSN: 00222836. DOI: 10.1016/j.jmb.2009.07.010.

Jarzab, Anna et al. (May 2020). “Meltome atlas—thermal proteome stability across the tree of life”. In: *Nature Methods* 17.5, pp. 495–503. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-020-0801-4.

Jiménez-García, Brian et al. (Jan. 1, 2018). “LightDock: a new multi-scale approach to protein–protein docking”. In: *Bioinformatics* 34.1. Ed. by Alfonso Valencia, pp. 49–55. ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btx555.

Jirapanjawat, Thanavit et al. (Dec. 2016). “The Redox Cofactor F_{420} Protects Mycobacteria from Diverse Antimicrobial Compounds and Mediates a Reductive Detoxification System”. In: *Applied and Environmental Microbiology* 82.23. Ed. by H. Atomi, pp. 6810–6818. ISSN: 0099-2240, 1098-5336. DOI: 10.1128/AEM.02500-16.

Jones, Brian V. et al. (Sept. 9, 2008). “Functional and comparative metagenomic analysis of bile salt hydrolase activity in the human gut microbiome”. In: *Proceedings of the National Academy of Sciences* 105.36, pp. 13580–13585. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0804437105.

Jones, Philip et al. (May 1, 2014). “InterProScan 5: genome-scale protein function classification”. In: *Bioinformatics* 30.9, pp. 1236–1240. ISSN: 1367-4811, 1367-4803. DOI: 10.1093/bioinformatics/btu031.

Jumper, John et al. (Aug. 26, 2021). “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873, pp. 583–589. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-021-03819-2.

Jung, Matthew S. et al. (Dec. 12, 2023). “Essential role of *Helicobacter pylori* apolipoprotein N-acyltransferase (Lnt) in stomach colonization”. In: *Infection and Immunity* 91.12. Ed. by Denise Monack, e00369–23. ISSN: 0019-9567, 1098-5522. DOI: 10.1128/iai.00369-23.

Kaminski, Kamil et al. (Oct. 3, 2023). “pLM-BLAST: distant homology detection based on direct comparison of sequence representations from protein language models”. In: *Bioinformatics* 39.10. Ed. by Lenore Cowen, btad579. ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btad579.

Karasikov, Mikhail et al. (Nov. 27, 2025). “Efficient and accurate search in petabase-scale sequence repositories”. In: *Nature* 647.8091, pp. 1036–1044. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-025-09603-w.

Katz, Kenneth et al. (Jan. 7, 2022). “The Sequence Read Archive: a decade more of explosive growth”. In: *Nucleic Acids Research* 50 (D1), pp. D387–D390. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkab1053.

Keenan, Jacqueline et al. (June 2000). “Immune Response to an 18-Kilodalton Outer Membrane Antigen Identifies Lipoprotein 20 as a *Helicobacter pylori* Vaccine Candidate”. In: *Infection and Immunity* 68.6. Ed. by J. D. Clements, pp. 3337–3343. ISSN: 0019-9567, 1098-5522. DOI: 10.1128/IAI.68.6.3337-3343.2000.

Kelly, Steven L. and Diane E. Kelly (Feb. 19, 2013). “Microbial cytochromes P450: biodiversity and biotechnology. Where do cytochromes P450 come from, what do they do and what can they do for us?” In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 368.1612, p. 20120476. ISSN: 0962-8436, 1471-2970. DOI: 10.1098/rstb.2012.0476.

Kemparaju, K. and K. S. Girish (Jan. 2006). “Snake venom hyaluronidase: a therapeutic target”. In: *Cell Biochemistry and Function* 24.1, pp. 7–12. ISSN: 0263-6484, 1099-0844. DOI: 10.1002/cbf.1261.

Kessler, Pascal et al. (Aug. 2017). “The three-finger toxin fold: a multifunctional structural scaffold able to modulate cholinergic functions”. In: *Journal of Neurochemistry* 142 (S2), pp. 7–18. ISSN: 0022-3042, 1471-4159. DOI: 10.1111/jnc.13975.

Kim, Jong Nam et al. (May 15, 2014). “Nitrogen Utilization and Metabolism in Ruminococcus albus 8”. In: *Applied and Environmental Microbiology* 80.10. Ed. by C. R. Lovell, pp. 3095–3102. ISSN: 0099-2240, 1098-5336. DOI: 10.1128/AEM.00029-14.

Kim, Ah-Ram et al. (Feb. 21, 2024). *Enhanced Protein-Protein Interaction Discovery via AlphaFold-Multimer*. DOI: 10.1101/2024.02.19.580970.

Kirchberger, Paul C., Zachary A. Martinez, and Howard Ochman (June 28, 2022). "Organizing the Global Diversity of Microviruses". In: *mBio* 13.3. Ed. by Graham F. Hatfull, e00588–22. ISSN: 2150-7511. DOI: 10.1128/mbio.00588-22.

Koh, Huan Yee et al. (June 17, 2024). "Physicochemical graph neural network for learning protein–ligand interaction fingerprints from sequence data". In: *Nature Machine Intelligence* 6.6, pp. 673–687. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00847-1.

Koppel, Nitzan et al. (May 15, 2018). "Discovery and characterization of a prevalent human gut bacterial enzyme sufficient for the inactivation of a family of plant toxins". In: *eLife* 7, e33953. ISSN: 2050-084X. DOI: 10.7554/eLife.33953.

Landrum, Greg et al. (Apr. 30, 2026). *rdkit/rdkit: 2026_03_2 (Q1 2026) Release*. Version Release_2026_03_2. DOI: 10.5281/ZENODO.591637.

Lang, Eric J. M. et al. (July 12, 2022). "Generalized Born Implicit Solvent Models Do Not Reproduce Secondary Structures of *De Novo* Designed Glu/Lys Peptides". In: *Journal of Chemical Theory and Computation* 18.7, pp. 4070–4076. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.1c01172.

Lau, Andy M., Shaun M. Kandathil, and David T. Jones (Dec. 19, 2023). "Merizo: a rapid and accurate protein domain segmentation method using invariant point attention". In: *Nature Communications* 14.1, p. 8445. ISSN: 2041-1723. DOI: 10.1038/s41467-023-43934-4.

Lau, Andy M. et al. (Nov. 2024). "Exploring structural diversity across the protein universe with The Encyclopedia of Domains". In: *Science* 386.6721, eadq4946. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.adq4946.

Lau, Cheryl K. Y., Karla D. Krewulak, and Hans J. Vogel (Mar. 2016). "Bacterial ferrous iron transport: the Feo system". In: *FEMS Microbiology Reviews* 40.2. Ed. by Wilbert Bitter, pp. 273–298. ISSN: 1574-6976. DOI: 10.1093/femsre/fuv049.

Lawrence, Michael C. and Peter M. Colman (Dec. 1993). "Shape Complementarity at Protein/Protein Interfaces". In: *Journal of Molecular Biology* 234.4, pp. 946–950. ISSN: 00222836. DOI: 10.1006/jmbi.1993.1648.

Le Guilloux, Vincent, Peter Schmidtke, and Pierre Tuffery (Dec. 2009). "Fpocket: An open source platform for ligand pocket detection". In: *BMC Bioinformatics* 10.1, p. 168. ISSN: 1471-2105. DOI: 10.1186/1471-2105-10-168.

Leclercq, Mickael and Arnaud Droit (Feb. 6, 2026). "Protein Language Models: Applications and Perspectives". In: *Journal of Proteome Research* 25.2, pp. 507–524. ISSN: 1535-3893, 1535-3907. DOI: 10.1021/acs.jproteome.5c00506.

Lensink, Marc F., Raúl Méndez, and Shoshana J. Wodak (Dec. 2007). "Docking and scoring protein complexes: CAPRI 3rd Edition". In: *Proteins: Structure, Function, and Bioinformatics* 69.4, pp. 704–718. ISSN: 0887-3585, 1097-0134. DOI: 10.1002/prot.21804.

Lesk, Arthur M (Dec. 1995). "NAD-binding domains of dehydrogenases". In: *Current Opinion in Structural Biology* 5.6, pp. 775–783. ISSN: 0959440X. DOI: 10.1016/0959-440X(95)80010-7.

Lewis, T E, I Sillitoe, and J G Lees (May 15, 2019). "cath-resolve-hits: a new tool that resolves domain matches suspiciously quickly". In: *Bioinformatics* 35.10. Ed. by John Hancock, pp. 1766–1767. ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/bty863.

Lin, Yeqing et al. (2024). *Out of Many, One: Designing and Scaffolding Proteins at the Scale of the Structural Universe with Genie 2*. Version Number: 1. DOI: 10.48550/ARXIV.2405.15489.

Lin, Zeming et al. (Mar. 17, 2023). "Evolutionary-scale prediction of atomic-level protein structure with a language model". In: *Science* 379.6637, pp. 1123–1130. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.ade2574.

Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou (Dec. 2008). "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. 2008 Eighth IEEE International Conference on Data Mining (ICDM). Pisa, Italy: IEEE, pp. 413–422. ISBN: 978-0-7695-3502-9. DOI: 10.1109/ICDM.2008.17.

Liu, Hongbin et al. (Jan. 2026). "Exploring functional insights into the human gut microbiome via the structural proteome". In: *Cell Host & Microbe* 34.1, 167–185.e9. ISSN: 19313128. DOI: 10.1016/j.chom.2025.11.001.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605.

Maestro, Beatriz and Jesús Sanz (June 14, 2016). "Choline Binding Proteins from Streptococcus pneumoniae: A Dual Role as Enzybiotics and Targets for the Design of New Antimicrobials". In: *Antibiotics* 5.2, p. 21. ISSN: 2079-6382. DOI: 10.3390/antibiotics5020021.

Mahant, Shweta et al. (Oct. 2021). "The Synergistic Role of Tip α, Nucleolin and Ras in Helicobacter pylori Infection Regulates the Cell Fate Towards Inflammation or Apoptosis". In: *Current Microbiology* 78.10, pp. 3720–3732. ISSN: 0343-8651, 1432-0991. DOI: 10.1007/s00284-021-02626-2.

Mahtha, Sanjeet Kumar, Sureshkumar Venkadesan, and Debasisa Mohanty (Jan. 6, 2026). "Comparative evaluation of the prediction accuracy of AlphaFold and ESMFold for monomeric and dimeric proteins". In: *NAR Genomics and Bioinformatics* 8.1, lqag002. ISSN: 2631-9268. DOI: 10.1093/nargab/lqag002.

Maier, James A. et al. (Aug. 11, 2015). "ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB". In: *Journal of Chemical Theory and Computation* 11.8, pp. 3696–3713. ISSN: 1549-9618, 1549-9626. DOI: 10.1021/acs.jctc.5b00255.

Manjunatha Kini, R (Dec. 2003). "Excitement ahead: structure, function and mechanism of snake venom phospholipase A2 enzymes". In: *Toxicon* 42.8, pp. 827–840. ISSN: 00410101. DOI: 10.1016/j.toxicon.2003.11.002.

Martinez, Diego et al. (May 2008). "Genome sequencing and analysis of the biomass-degrading fungus Trichoderma reesei (syn. Hypocrea jecorina)". In: *Nature Biotechnology* 26.5, pp. 553–560. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt1403.

Martinez, Zachary A., Richard M. Murray, and Matt W. Thomson (Oct. 27, 2023). *TRILL: Orchestrating Modular Deep-Learning Workflows for Democratized, Scalable Protein Analysis and Engineering*. Preprint, bioRxiv. DOI: 10.1101/2023.10.24.563881.

Mazmanian, Sarkis K. et al. (July 2005). "An Immunomodulatory Molecule of Symbiotic Bacteria Directs Maturation of the Host Immune System". In: *Cell* 122.1, pp. 107–118. ISSN: 00928674. DOI: 10.1016/j.cell.2005.05.007.

McClain, Mark S., Bradley J. Voss, and Timothy L. Cover (June 30, 2020). "Lipoprotein Processing and Sorting in Helicobacter pylori". In: *mBio* 11.3. Ed. by Steven J. Norris, e00911–20. ISSN: 2161-2129, 2150-7511. DOI: 10.1128/mBio.00911-20.

McClain, Mark S. et al. (May 2, 2024). "Fatty acids of *Helicobacter pylori* lipoproteins CagT and Lpp20". In: *Microbiology Spectrum* 12.5. Ed. by Stacey D. Gilk, e00470–24. ISSN: 2165-0497. DOI: 10.1128/spectrum.00470-24.

McInnes, Leland, John Healy, and James Melville (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. Version Number: 3. DOI: 10.48550/ARXIV.1802.03426.

McNutt, Andrew T. et al. (Mar. 2, 2025). "GNINA 1.3: the next increment in molecular docking with deep learning". In: *Journal of Cheminformatics* 17.1, p. 28. ISSN: 1758-2946. DOI: 10.1186/s13321-025-00973-x.

Mirarchi, Antonio, Toni Giorgino, and Gianni De Fabritiis (Nov. 28, 2024). "mdCATH: A Large-Scale MD Dataset for Data-Driven Computational Biophysics". In: *Scientific Data* 11.1, p. 1299. ISSN: 2052-4463. DOI: 10.1038/s41597-024-04140-z.

Mirdita, Milot et al. (June 2022). "ColabFold: making protein folding accessible to all". In: *Nature Methods* 19.6, pp. 679–682. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-022-01488-1.

Mitternacht, Simon (Feb. 18, 2016). "FreeSASA: An open source C library for solvent accessible surface area calculations". In: *F1000Research* 5, p. 189. ISSN: 2046-1402. DOI: 10.12688/f1000research.7931.1.

Mo, Ran et al. (June 28, 2022). "Evolutionary Principles of Bacterial Signaling Capacity and Complexity". In: *mBio* 13.3. Ed. by Qijing Zhang, e00764–22. ISSN: 2150-7511. DOI: 10.1128/mbio.00764-22.

Morningstar-Wright, Lindsay et al. (Feb. 14, 2022). "The TNF-Alpha Inducing Protein is Associated With Gastric Inflammation and Hyperplasia in a Murine Model of Helicobacter pylori Infection". In: *Frontiers in Pharmacology* 13, p. 817237. ISSN: 1663-9812. DOI: 10.3389/fphar.2022.817237.

Munsamy, Geraldene et al. (May 5, 2024). *Conditional language models enable the efficient design of proficient enzymes*. DOI: 10.1101/2024.05.03.592223.

Mutti, Giacomo, Eduard Ocaña-Pallarès, and Toni Gabaldón (July 1, 2025). "Newly Developed Structure-Based Methods Do Not Outperform Standard Sequence-Based Methods for Large-Scale Phylogenomics". In: *Molecular Biology and Evolution* 42.7. Ed. by Belinda Chang, msaf149. ISSN: 0737-4038, 1537-1719. DOI: 10.1093/molbev/msaf149.

Naikare, Hemant et al. (Oct. 2006). "Major Role for FeoB in *Campylobacter jejuni* Ferrous Iron Acquisition, Gut Colonization, and Intracellular Survival". In: *Infection and Immunity* 74.10, pp. 5433–5444. ISSN: 0019-9567, 1098-5522. DOI: 10.1128/IAI.00052-06.

Nijkamp, Erik et al. (Nov. 2023). "ProGen2: Exploring the boundaries of protein language models". In: *Cell Systems* 14.11, 968–978.e3. ISSN: 24054712. DOI: 10.1016/j.cels.2023.10.002.

Novakova, Renata et al. (Feb. 23, 2022). "A New Family of Transcriptional Regulators Activating Biosynthetic Gene Clusters for Secondary Metabolites". In: *International Journal of Molecular Sciences* 23.5, p. 2455. ISSN: 1422-0067. DOI: 10.3390/ijms23052455.

O'Boyle, Noel M et al. (Dec. 2011). "Open Babel: An open chemical toolbox". In: *Journal of Cheminformatics* 3.1, p. 33. ISSN: 1758-2946. DOI: 10.1186/1758-2946-3-33.

O'Shea-Wheller, Thomas A. and Katie I. Murray (Mar. 3, 2026). "Deep learning in biology faces a transferability crisis". In: *PLOS Biology* 24.3, e3003656. ISSN: 1545-7885. DOI: 10.1371/journal.pbio.3003656.

Ottesen, Elizabeth A. and Jared R. Leadbetter (May 15, 2011). "Formyltetrahydrofolate Synthetase Gene Diversity in the Guts of Higher Termites with Different Diets and Lifestyles". In: *Applied and Environmental Microbiology* 77.10, pp. 3461–3467. ISSN: 0099-2240, 1098-5336. DOI: 10.1128/AEM.02657-10.

Outeiral, Carlos and Charlotte M. Deane (Feb. 23, 2024). “Codon language embeddings provide strong signals for use in protein engineering”. In: *Nature Machine Intelligence* 6.2, pp. 170–179. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00791-0.

Parks, Donovan H et al. (Jan. 6, 2026). “GTDB release 10: a complete and systematic taxonomy for 715 230 bacterial and 17 245 archaeal genomes”. In: *Nucleic Acids Research* 54 (D1), pp. D743–D754. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkaf1040.

Passaro, Saro et al. (June 18, 2025). *Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction*. DOI: 10.1101/2025.06.14.659707.

Paszke, Adam et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Version Number: 1. DOI: 10.48550/ARXIV.1912.01703.

Pavlopoulos, Georgios A. et al. (Oct. 19, 2023). “Unraveling the functional dark matter through global metagenomics”. In: *Nature* 622.7983, pp. 594–602. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06583-7.

Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Peek, Richard M. and Martin J. Blaser (Jan. 1, 2002). “Helicobacter pylori and gastrointestinal tract adenocarcinomas”. In: *Nature Reviews Cancer* 2.1, pp. 28–37. ISSN: 1474-175X, 1474-1768. DOI: 10.1038/nrc703.

Plovier, Hubert et al. (Jan. 2017). “A purified membrane protein from Akkermansia muciniphila or the pasteurized bacterium improves metabolism in obese and diabetic mice”. In: *Nature Medicine* 23.1, pp. 107–113. ISSN: 1078-8956, 1546-170X. DOI: 10.1038/nm.4236.

Portero, Luciano Raúl et al. (Jan. 2019). “Photolyases and Cryptochromes in UV -resistant Bacteria from High-altitude Andean Lakes”. In: *Photochemistry and Photobiology* 95.1, pp. 315–330. ISSN: 0031-8655, 1751-1097. DOI: 10.1111/php.13061.

Prihoda, David et al. (Nov. 28, 2025). *Ovo, an Open-Source Ecosystem for De Novo Protein Design*. DOI: 10.1101/2025.11.27.691041.

Principal Component Analysis (2002). Springer Series in Statistics. New York: Springer-Verlag. ISBN: 978-0-387-95442-4. DOI: 10.1007/b98835.

Pudžiuvelytė, Ieva et al. (Mar. 29, 2024). “TemStaPro: protein thermostability prediction using sequence representations from protein language models”. In: *Bioinformatics* 40.4. Ed. by Lenore Cowen, btae157. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btae157.

Quinn, Robert A. et al. (Mar. 5, 2020). “Global chemical effects of the microbiome include new bile-acid conjugations”. In: *Nature* 579.7797, pp. 123–129. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-020-2047-9.

Ragsdale, Stephen W. and Elizabeth Pierce (Dec. 2008). "Acetogenesis and the Wood–Ljungdahl pathway of CO2 fixation". In: *Biochimica et Biophysica Acta (BBA) - Proteins and Proteomics* 1784.12, pp. 1873–1898. ISSN: 15709639. DOI: 10.1016/j.bbapap.2008.08.012.

Rajbhandari, Samyam et al. (Nov. 2020). "ZeRO: Memory optimizations Toward Training Trillion Parameter Models". In: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. Atlanta, GA, USA: IEEE, pp. 1–16. ISBN: 978-1-7281-9998-6. DOI: 10.1109/SC41405.2020.00024.

Ramos, Juan L. et al. (June 2005). "The TetR Family of Transcriptional Repressors". In: *Microbiology and Molecular Biology Reviews* 69.2, pp. 326–356. ISSN: 1092-2172, 1098-5557. DOI: 10.1128/MMBR.69.2.326-356.2005.

Richardson, Lorna et al. (Jan. 6, 2023). "MGnify: the microbiome sequence data analysis resource in 2023". In: *Nucleic Acids Research* 51 (D1), pp. D753–D759. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkac1080.

Rimal, Bipin et al. (Feb. 22, 2024). "Bile salt hydrolase catalyses formation of amine-conjugated bile acids". In: *Nature* 626.8000, pp. 859–863. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06990-w.

Rives, Alexander et al. (Apr. 13, 2021). "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15, e2016239118. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.2016239118.

Rocklin, Gabriel J. et al. (July 14, 2017). "Global analysis of protein folding using massively parallel design, synthesis, and testing". In: *Science* 357.6347, pp. 168–175. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aan0693.

Rost, Burkhard (Feb. 1999). "Twilight zone of protein sequence alignments". In: *Protein Engineering, Design and Selection* 12.2, pp. 85–94. ISSN: 1741-0134, 1741-0126. DOI: 10.1093/protein/12.2.85.

Round, June L. and Sarkis K. Mazmanian (July 6, 2010). "Inducible Foxp3$^+$ regulatory T-cell development by a commensal bacterium of the intestinal microbiota". In: *Proceedings of the National Academy of Sciences* 107.27, pp. 12204–12209. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0909122107.

Sancar, Aziz (June 1, 2003). "Structure and Function of DNA Photolyase and Cryptochrome Blue-Light Photoreceptors". In: *Chemical Reviews* 103.6, pp. 2203–2238. ISSN: 0009-2665, 1520-6890. DOI: 10.1021/cr0204348.

Sanger, F. et al. (Feb. 1977). "Nucleotide sequence of bacteriophage φX174 DNA". In: *Nature* 265.5596, pp. 687–695. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/265687a0.

Santos-Aberturas, Javier and Natalia Vior (Feb. 2, 2022). "Beyond Soil-Dwelling Actinobacteria: Fantastic Antibiotics and Where to Find Them". In: *Antibiotics* 11.2, p. 195. ISSN: 2079-6382. DOI: 10.3390/antibiotics11020195.

Sarkar, Arpan, Kumaresh Krishnan, and Sean R. Eddy (June 5, 2024). *Protein sequence domain annotation using a language model*. DOI: 10.1101/2024.06.04.596712.

Schaeffer, R. Dustin et al. (Feb. 28, 2024). "ECOD domain classification of 48 whole proteomes from AlphaFold Structure Database using DPAM2". In: *PLOS Computational Biology* 20.2. Ed. by Roland L. Dunbrack, e1011586. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1011586.

Schütze, Konstantin et al. (Nov. 17, 2022). "Nearest neighbor search on embeddings rapidly identifies distant protein relations". In: *Frontiers in Bioinformatics* 2, p. 1033775. ISSN: 2673-7647. DOI: 10.3389/fbinf.2022.1033775.

Shenoy, Aditi et al. (Jan. 2, 2024). "M-Ionic: prediction of metal-ion-binding sites from sequence using residue embeddings". In: *Bioinformatics* 40.1. Ed. by Xin Gao, btad782. ISSN: 1367-4803, 1367-4811. DOI: 10.1093/bioinformatics/btad782.

Shi, Yu et al. (2026). *lightgbm: Light Gradient Boosting Machine*.

Sigrist, Christian J. A. et al. (Jan. 2010). "PROSITE, a protein domain database for functional characterization and annotation". In: *Nucleic Acids Research* 38 (suppl_-1), pp. D161–D166. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkp885.

Smillie, Chris S. et al. (Dec. 2011). "Ecology drives a global network of gene exchange connecting the human microbiome". In: *Nature* 480.7376, pp. 241–244. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature10571.

Soares, Eduardo et al. (2024). *A Large Encoder-Decoder Family of Foundation Models For Chemical Language*. Version Number: 1. DOI: 10.48550/ARXIV.2407.20267.

Stebbins, C. Erec and Jorge E. Galán (Aug. 2001). "Structural mimicry in bacterial virulence". In: *Nature* 412.6848, pp. 701–705. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/35089000.

Steinegger, Martin and Johannes Söding (Nov. 2017). "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets". In: *Nature Biotechnology* 35.11, pp. 1026–1028. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt.3988.

Stirpe, Fiorenzo (June 2013). "Ribosome-inactivating proteins: From toxins to useful proteins". In: *Toxicon* 67, pp. 12–16. ISSN: 00410101. DOI: 10.1016/j.toxicon.2013.02.005.

Stock, Ann M., Victoria L. Robinson, and Paul N. Goudreau (June 2000). "Two-Component Signal Transduction". In: *Annual Review of Biochemistry* 69.1, pp. 183–215. ISSN: 0066-4154, 1545-4509. DOI: 10.1146/annurev.biochem.69.1.183.

Su, Jin et al. (Oct. 2, 2023). *SaProt: Protein Language Modeling with Structure-aware Vocabulary*. DOI: 10.1101/2023.10.01.560349.

Subramanian, Arjuna M. et al. (Dec. 23, 2023). *Unexplored regions of the protein sequence-structure map revealed at scale by a library of foldtuned language models*. DOI: 10.1101/2023.12.22.573145.

Suganuma, Masami et al. (July 2008). "TNF-α-inducing protein, a carcinogenic factor secreted from *H. pylori* , enters gastric cancer cells". In: *International Journal of Cancer* 123.1, pp. 117–122. ISSN: 0020-7136, 1097-0215. DOI: 10.1002/ijc.23484.

Suganuma, Masami et al. (Mar. 1, 2021). "Role of TNF-α-Inducing Protein Secreted by Helicobacter pylori as a Tumor Promoter in Gastric Cancer and Emerging Preventive Strategies". In: *Toxins* 13.3, p. 181. ISSN: 2072-6651. DOI: 10.3390/toxins13030181.

Suryanarayanan, Parthasarathy et al. (2024). *Multi-view biomedical foundation models for molecule-target and property prediction*. Version Number: 4. DOI: 10.48550/ARXIV.2410.19704.

Takeuchi, Hiroaki et al. (Nov. 17, 2021). "*Helicobacter pylori* protein that binds to and activates platelet specifically reacts with sera of *H. pylori* -associated chronic immune thrombocytopenia". In: *Platelets* 32.8, pp. 1120–1123. ISSN: 0953-7104, 1369-1635. DOI: 10.1080/09537104.2021.1945570.

Tam, James et al. (Nov. 16, 2015). "Antimicrobial Peptides from Plants". In: *Pharmaceuticals* 8.4, pp. 711–757. ISSN: 1424-8247. DOI: 10.3390/ph8040711.

Tartici, Alp, Gowri Nayar, and Russ B Altman (June 2, 2025). "Pool PaRTI: a PageRank-based pooling method for identifying critical residues and enhancing protein sequence representations". In: *Bioinformatics* 41.6. Ed. by Xin Gao, btaf330. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btaf330.

Taylor, Barry L. and Igor B. Zhulin (June 1999). "PAS Domains: Internal Sensors of Oxygen, Redox Potential, and Light". In: *Microbiology and Molecular Biology Reviews* 63.2, pp. 479–506. ISSN: 1092-2172, 1098-5557. DOI: 10.1128/MMBR.63.2.479-506.1999.

Teufel, Felix et al. (July 2022). "SignalP 6.0 predicts all five types of signal peptides using protein language models". In: *Nature Biotechnology* 40.7, pp. 1023–1025. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/s41587-021-01156-3.

Tierney, Braden T. et al. (Aug. 2019). "The Landscape of Genetic Content in the Gut and Oral Human Microbiome". In: *Cell Host & Microbe* 26.2, 283–295.e8. ISSN: 19313128. DOI: 10.1016/j.chom.2019.07.008.

Torres, Marcelo D. T. et al. (Nov. 4, 2021). "Mining for encrypted peptide antibiotics in the human proteome". In: *Nature Biomedical Engineering* 6.1, pp. 67–75. ISSN: 2157-846X. DOI: 10.1038/s41551-021-00801-1.

Tosi, Tommaso et al. (May 19, 2009). "Structures of the tumor necrosis factor α inducing protein Tipα: A novel virulence factor from *Helicobacter pylori*". In: *FEBS Letters* 583.10, pp. 1581–1585. ISSN: 0014-5793, 1873-3468. DOI: 10.1016/j.febslet.2009.04.033.

Tsuge, Hideaki et al. (Oct. 2009). "Structural basis for the Helicobacter pylori-carcinogenic TNF-α-inducing protein". In: *Biochemical and Biophysical Research Communications* 388.2, pp. 193–198. ISSN: 0006291X. DOI: 10.1016/j.bbrc.2009.07.121.

Vallese, Francesca et al. (Dec. 2017). "Helicobacter pylori antigenic Lpp20 is a structural homologue of Tipα and promotes epithelial-mesenchymal transition". In: *Biochimica et Biophysica Acta (BBA) - General Subjects* 1861.12, pp. 3263–3271. ISSN: 03044165. DOI: 10.1016/j.bbagen.2017.09.017.

Van Kempen, Michel et al. (Feb. 2024). "Fast and accurate protein structure search with Foldseek". In: *Nature Biotechnology* 42.2, pp. 243–246. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/s41587-023-01773-0.

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. Version Number: 7. DOI: 10.48550/ARXIV.1706.03762.

Venter, J. Craig et al. (Apr. 2, 2004). "Environmental Genome Shotgun Sequencing of the Sargasso Sea". In: *Science* 304.5667, pp. 66–74. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1093857.

Vernikos, Georgios S. and Julian Parkhill (Sept. 15, 2006). "Interpolated variable order motifs for identification of horizontally acquired DNA: revisiting the *Salmonella* pathogenicity islands". In: *Bioinformatics* 22.18, pp. 2196–2203. ISSN: 1367-4811, 1367-4803. DOI: 10.1093/bioinformatics/btl369.

Waman, Vaishali P et al. (Jan. 6, 2025). "CATH v4.4: major expansion of CATH by experimental and predicted structural data". In: *Nucleic Acids Research* 53 (D1), pp. D348–D355. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkae1087.

Wang, Ercheng et al. (Aug. 28, 2019). "End-Point Binding Free Energy Calculation with MM/PBSA and MM/GBSA: Strategies and Applications in Drug Design". In: *Chemical Reviews* 119.16, pp. 9478–9508. ISSN: 0009-2665, 1520-6890. DOI: 10.1021/acs.chemrev.9b00055.

Wang, Jiawei et al. (July 2, 2020). "PaCRISPR: a server for predicting and visualizing anti-CRISPR proteins". In: *Nucleic Acids Research* 48 (W1), W348–W357. ISSN: 0305-1048, 1362-4962. DOI: 10.1093/nar/gkaa432.

Wang, Zechen et al. (Mar. 20, 2025). "Robust enzyme discovery and engineering with deep learning using CataPro". In: *Nature Communications* 16.1, p. 2736. ISSN: 2041-1723. DOI: 10.1038/s41467-025-58038-4.

Watanabe, Tatsuro et al. (June 2010). "Nucleolin as cell surface receptor for tumor necrosis factor-α inducing protein: a carcinogenic factor of Helicobacter pylori". In: *Journal of Cancer Research and Clinical Oncology* 136.6, pp. 911–921. ISSN: 0171-5216, 1432-1335. DOI: 10.1007/s00432-009-0733-y.

Watson, Joseph L. et al. (Aug. 31, 2023). "De novo design of protein structure and function with RFdiffusion". In: *Nature* 620.7976, pp. 1089–1100. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06415-8.

Williams, Christopher J. et al. (Jan. 2018). "MolProbity: More and better reference data for improved all-atom structure validation". In: *Protein Science* 27.1, pp. 293–315. ISSN: 0961-8368, 1469-896X. DOI: 10.1002/pro.3330.

Winstedt, Lena and Claes Von Wachenfeldt (Dec. 2000). "Terminal Oxidases of *Bacillus subtilis* Strain 168: One Quinol Oxidase, Cytochrome aa_3 or Cytochrome *bd* , Is Required for Aerobic Growth". In: *Journal of Bacteriology* 182.23, pp. 6557–6564. ISSN: 0021-9193, 1098-5530. DOI: 10.1128/JB.182.23.6557-6564.2000.

Wohlwend, Jeremy et al. (Nov. 20, 2024). *Boltz-1 Democratizing Biomolecular Interaction Modeling*. DOI: 10.1101/2024.11.19.624167.

Wolf, Thomas et al. (2020). "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online: Association for Computational Linguistics, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

Xu, Xiaopeng et al. (Apr. 2026). "ProteinMCP : An agentic AI framework for autonomous protein engineering". In: *Protein Science* 35.4, e70547. ISSN: 0961-8368, 1469-896X. DOI: 10.1002/pro.70547.

Yadahalli, Shilpa and Chandra S. Verma (Oct. 16, 2020). *Predicting Cell-Penetrating Peptides: Building and Interpreting Random Forest based prediction Models*. DOI: 10.1101/2020.10.15.341149.

Yu, Tianhao et al. (Mar. 31, 2023). "Enzyme function prediction using contrastive learning". In: *Science* 379.6639, pp. 1358–1363. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.adf2465.

Zhu, Feng et al. (Feb. 9, 2018). "The Plant Ribosome-Inactivating Proteins Play Important Roles in Defense against Pathogens and Insect Pest Attacks". In: *Frontiers in Plant Science* 9, p. 146. ISSN: 1664-462X. DOI: 10.3389/fpls.2018.00146.