Exploring the Trinity of Protein Science: Structure, Stability, and Function Through the Lens of Machine Learning

Humood Alanzi

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington

2024

Committee:

David A. C. Beck, Chair

Joseph Hellerstein

Program Authorized to Offer Degree:

Chemical Engineering

University of Washington

Abstract

Exploring the Trinity of Protein Science: Structure, Function, and Stability Through the Lens of Machine Learning

Humood Alanzi

Chair of the Supervisory Committee:
David A. C. Beck
Department of Chemical Engineering

Machine learning and deep learning are revolutionizing protein science by enabling the prediction of complex, emergent biophysical properties. This thesis presents two novel computational models that leverage these technologies to predict protein thermostability and function, illustrating how they can serve as powerful hypothesis generators within iterative "design, build, test, and learn" cycles. Chapter 2 details NOMELT, a generative model trained as a neural machine translator between mesophilic and thermophilic protein domains, which uses a vast new dataset of homologous protein pairs to enhance the stability of generated thermophilic sequences. Chapter 3 introduces the PairProphet pipeline, which integrates diverse sequence and structural data to predict functional similarities between protein pairs with high accuracy, highlighting the importance of sequence-based features and the potential limitations of current structural analysis techniques. The thesis suggests that integrating ecological information and

pangenomic analyses could further enhance the predictive power of these models, pointing to these approaches as promising areas for future research. This work contributes to a deeper understanding of protein behaviors under diverse environmental conditions and suggests pathways to more effectively design proteins for therapeutic and industrial applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

It is challenging to fully express my gratitude to the numerous individuals and organizations that have supported me throughout my master's journey and the composition of this thesis. Nevertheless, I am deeply thankful to every friend, committee member, collaborator, co-author, and mentor who has been part of this incredible journey. I cannot name you all and I apologize for my inadequacy!

My research would not have been possible without the guidance and mentorship of Professor Beck, who has been an exceptional human being and Principal Investigator. I am grateful to my committee member, Professor Hellerstein, for their invaluable guidance, patience, and advice. I also extend my thanks to Dr. Evan Komp, not only for being an erudite collaborator but also an extraordinary friend. Finally, many thanks to my collaborators Logan Roberts, Ryan Francis, Christian Phillips, Chau Vuong, Amin Mosallanejad, and Marlo Zorman.

The support of my family—both nuclear and extended—has been a tremendous blessing. I could not have reached this point without their endless love, encouragement, kindness, prayers, and faith. Special thanks go to my father, brother, and sister for their unwavering support.

I am also thankful for the resources provided by the University of Washington's Hyak computer cluster and its IT team, which were instrumental in facilitating this research. The projects presented in this thesis were funded by the 2023-2024 Herbold Fellowship and Kuwait's Ministry of Higher Education. I acknowledge that the findings and conclusions of this thesis do not necessarily reflect the views of these funding agencies.

Lastly, I recognize that the university where I conducted this work is situated on the ancestral lands of the Coast Salish peoples. This land touches the shared waters of all tribes and bands within the Suquamish, Tulalip, and Muckleshoot Nations.

DEDICATION

To My Mother.

You are dearly missed.

Chapter 1. INTRODUCTION

Proteins, alongside nucleotides (DNA and RNA) and fatty acids (lipids, membranes, etc.), constitute the fundamental building blocks of life. Within the cells, which are comprised approximately via 70% water and 30% chemicals, proteins notably account for about half of the dry weight, emphasizing their critical role.[1] Classically, these macromolecules are often described as the "workhorses" of the cell as they facilitate catalysis, transport, scaffolding, and gene regulation, among many other functions. The functionality of a protein is intimately tied with the protein's three-dimensional conformation (known as fold or native structure), the understanding of which remains one of the principle aims of molecular biology and related disciplines.

Protein folding—the process where a linear chain of amino acids self-organizes into a functional three-dimensional structure—has been a cornerstone problem in molecular biology for nearly a century. First articulated by Mirsky and Pauling in the 1930s[2], our understanding has evolved from simple models of amino acid interactions to sophisticated computer-based protein design.[1]As this multidisciplinary research program (protein design) advanced, this set of open questions and challenges on the exact mechanistic understanding of the folding reaction was dubbed the "protein folding problem."[4] From a biophysical perspective, protein folding is a reversible and spontaneous chemical reaction in which the unfolded protein chain achieves equilibrium without breaking or forming covalent bonds, except for disulfide bonds. This process is governed by a balance of

[1] For a more in-depth retrospective on the history of *de novo* protein design, I highly recommend Korendovych, et al. comprehensive review.[3] It does not include the history of early condensed matter physics and its contributions to protein science, but it has an authoritative account of the field from the 1980s to early 2019. It periodizes the history to three major waves: I) manual protein design using physical models II) computational design guided by fundamental physicochemical principles III) fragment-based and bioinformatically informed computational protein design. It is, in my opinion, we are entering a new wave as discussed in the introduction.

entropic and enthalpic forces. Entropic forces largely derive from the protein's interactions with solvent molecules, primarily water, where the increase in entropy is a driving force as the solvent organizes around the protein structure. Enthalpic forces include intermolecular interactions not only within the amino acid chain but also with nearby solvent molecules, involving hydrogen bonding and van der Waals forces. These interactions are deeply influenced by the physicochemical conditions of the cellular environment—such as temperature, pressure, pH, and the presence of ions, osmolytes, and other solutes.[5] This cellular milieu is rich and crowded, which can significantly affect the protein folding pathways in surprising ways. However, the totality of these factors typically results in Gibbs free energy differences ranging from 5 to 15 kcal/mol (20 to 60 kJ/mol) between the native and unfolded states, observed across various molecular conditions and organismal habitat.[6,7] This journey is not merely a path to a static 'native fold' but a dynamic equilibrium where the protein may sample various conformations before stabilizing in its functional form. Thus, in a broader context, there exists a complex dynamic between protein structure, stability, and function.

$$\Delta\Delta G_{Fold} = \Delta G_{Native} - \Delta G_{Unfolded} = (\Delta H - T\Delta S)_{Native} - (\Delta H - T\Delta S)_{Unfolded} \quad (1.1)$$

Understanding the precise dynamics of protein folding is crucial not only for theoretical science but also for practical applications. Proper folding is essential for proteins to perform their diverse functions effectively. Conversely, misfolding can lead to serious diseases, underscoring the high stakes involved. For instance, misfolded proteins are implicated in various neurodegenerative disorders like Alzheimer's and Parkinson's, where aberrant folding patterns lead to toxic aggregations that disrupt cellular function.[8,9] Similarly, cystic fibrosis results from misfolding of

the CFTR protein, which affects chloride channel function, illustrating how a single misfolding event can disrupt a critical physiological process.[10] In the realm of oncology, the misfolding of tumor suppressor proteins, such as p53, can prevent them from effectively inhibiting cancerous cell growth, thereby contributing to tumor development.[8] These examples highlight the imperative to understand and predict protein folding pathways accurately, as slight deviations in folding can have profound function implications; achieving this control can lead to groundbreaking advances in medicine and biotechnology, from disease treatment to the development of superior enzymes.

The influence of a single mutation has long raised a fundamental question in protein science: Is all necessary information about a protein encoded in its sequence? This perspective, referred to as the sequence-structure-function paradigm, traditionally championed by luminaries like Nobel laureate Sydney Brenner, who famously suggested that specifying the amino acid sequence ought to be sufficient for correct folding, has shaped decades of research.[2] However, this view, while foundational, often overlooks the complex interactions of environmental factors and dynamic structural changes (metamorphic proteins, allosterity, moonlighting, intrinsically disordered regions (IDRs), etc.) that are critical for protein functionality.[7,9,12–15] The advent of modern computational tools, particularly those leveraging deep learning, has prompted a significant shift, challenging us to rethink protein structure and function. These tools are not merely reshaping our understanding; they are revealing the limits of our previous models and opening up new avenues for exploring protein science that were once thought to be beyond our reach.

[2] The exact quote taken from Phillip Ball's book *how life works* is "all you had to do was to specify the amino acid sequence and the folding would look after itself."[11] This view is also often associated by another luminary Christian B. Anfinsen. However, recent research effort indicate that he left the possibility open that proteins can fold switch from one ground state fold to another responding to different environmental signals.[12]

In particular, the year 2020 marked a seminal moment in computational biology with the introduction of AlphaFold2 by Google's DeepMind at the Critical Assessment of Protein Structure Prediction (CASP) 14 competition. This competition, seen as the "world championship" of protein structure prediction, tests algorithms' ability to predict 3D structures from amino acid sequences. AlphaFold2 didn't just surpass other competitors; it matched, and in some cases, exceeded the quality of experimental determinations, heralding what has been dubbed the "AlphaFold moment" in protein science.[16] Yet, this remarkable achievement does not signify the end of the protein folding problem but rather heralds a new era where neural networks and statistical learning are integral to the field.[17,18] This shift has not solved all existing questions but has transformed the landscape of what is possible in protein science. AlphaFold2's predictive capabilities have opened new avenues in various biological and biomedical fields, ranging from structural biology to drug discovery. However, designing functional proteins, such as enzymes and antibodies, poses additional challenges that go beyond structural prediction. Enzymes require precise modeling of active sites and their dynamics to catalyze reactions effectively[19,20], while antibodies need accurate prediction of loop-rich regions that determine binding specificity.[21]

Ultimately, deep learning, while in many ways still in its infancy, with the explosive development that AlphaFold2, has cemented itself as an indispensable tool in the tool kit of the protein scientist. However, these methods are not without limitations. To name a few, these models require vast high-quality datasets, and that is an emerging bottleneck as a lot of these state-of-the-art models already covered many of the known protein universe.[22,23] Moreover, the context behind the data generation, be it experiment or computational, matters as these models act as ‘differentiable programs’ that learn complex statistical patterns from said data, and we can see how that can act as both a blessing and a curse.[24] For example, AlphaFold2 given its training data

(MSA, PDB file, etc.) has trouble predicting single-domain globular structures characterized by NMR.[17,25] It also has limited success with IDRs, fold-switching proteins, etc., as mentioned previously.[17] In other words, these models serve as excellent high-quality hypothesis generators for experiment, their theoretical biases, however, should be taken into account; they are not first principle biophysical models.[5] Despite that, these models are helping us learn more and more about the complex world of proteins. Techniques like hallucination-based approaches and diffusion-based generative models, as discussed by Chu et al., are pushing the frontiers of de novo protein design.[26,27] These methods allow not only for the creation of stable structures but also for embedding functional motifs into proteins, paving the way for novel applications in biotechnology and medicine. Moreover, techniques like AF-Cluster are pioneering in their ability to predict multiple conformations of proteins, addressing the need to understand proteins in all their functional states.[28] Future advancements in computational biology must strive to build causal models that not only predict but also explain protein behaviors in diverse conditions, moving closer to a comprehensive understanding of protein function and design.

This thesis is just a minuscule contribution to the vast and rapidly developing field of computational protein science. In chapter 2, I will be discussing protein thermal stability as if it were a translation task, I will be discussing in detail the generation and validation of the corpus that will ultimately train a neural machine translator (NMT) to translate a protein sequence that folds at low temperature to one that folds at higher temperatures. From there, I will move on to chapter 3, where I will discuss some of the lesson learned from chapter 2 to produce a first-pass functional pairwise screening pipeline for proteins.

Chapter 2. TRANSLATING PROTEINS FROM LOW-TEMPERATURE TO HIGH-TEMPERATURE

Note: some content from this chapter has been published as Komp et al., 2023.[29] Other content from this chapter may appear in co-authors' dissertations and are currently forthcoming in scientific journals.[30]

2.1 INTRODUCTION AND BACKGROUND

Proteins are vital even outside of its function within cells. These miraculous macromolecules have a variety of applications. Indeed, they play pivotal roles in many industries. In the biochemical industry, enzymes such as lipases and proteases catalyze the breakdown of oily stains and soil in laundry detergents.[31] Similarly, in the pharmaceutical sector, proteins serve as foundational precursor for antibiotics and therapeutic agents.[32,33] Beyond these, proteins contribute to bioremediation[34–36], green chemistry[37], food science[38], and biofuels[39], underscoring their versatility (see Table 2.1 for a summary of protein applications). These applications derive from proteins' unique properties obtained from natural evolution. As enzymes, they offer regio-, stereo-, and enantio-selective catalysis while tolerating structurally diverse substrates. Moreover, they operate under mild conditions, enhancing reaction rates up to 10^{12} times compared to non-catalyzed reactions.[19] Environmentally, the byproducts of protein-based reactions require minimal cleanup before disposal into municipal sewers, offering an eco-friendly alternative as we shift away from fossil fuel reliance.[37]

Table 2.1. Sample protein applications in industry

Protein	Type/Source	Application
Protease[40]	Enzyme. *Klebsiella aerogenes, Bacillus acidipullulyticus, Bacillus subtilis*	Brewing, baking goods, protein processing, distilled spirits, laundry and dishwashing detergents, lens cleaners, leather and fur, chemicals
Lipases and Esterases[38]	Enzyme. Phospholidases, pregastric esterases, phosphatases	Cleaners, leather and fur, dairy, chemicals
Laccases[36]	Enzyme. *Myceliophthora thermophila*	Bioremediation, baking goods, wine
Strictosidine synthase[33]	Enzyme. *R. Serpentina*	Intermediate chemical for antibiotic production
Glucose oxidase[38]	Enzyme. Many sources	Food flavoring and shelf life improvement

However, proteins face technical limitations that hinder their broader industrial application. As discussed in the introduction, protein folding is crucial for maintaining functional integrity under physiological conditions relevant to an organism's ecological niche. Yet, perturbations such as temperature increases can lead to denaturation, where proteins lose their functional conformation—a process intimately tied to their evolutionary adaptation to specific cellular environments.[7,19] This limitation often clashes with industrial conditions, which typically involve extreme temperatures, pH levels, and organic solvents to enhance reaction and transport kinetics. Despite their catalytic specificity being a major advantage, it is paradoxically a constraint under these harsh conditions.[19] In other words, proteins advantage of being highly adapted to fit their natural environment also presents a limitation. Thus, the quest for high-temperature proteins (HTPs) that retain stability and high activity under such conditions is therefore imperative.[3]

To obtain HTPs, we have to think through protein thermal stability. Protein thermal stability can be conceptualized through two interrelated perspectives: thermodynamic stability and kinetic

[3] The other significant technical challenge, not covered in this thesis, involves cofactors and their recycling. Many proteins require cofactors to function, and in industrial settings, ensuring a steady supply of these cofactors remains a bottleneck.[37]

thermal stability.[4] As outlined in chapter 1, thermodynamic stability refers to the prevalence of native folded states over unfolded ones, governed by Gibbs free energy differences between these states. Kinetic stability, on the other hand, describes the rate at which proteins transition between states and the duration proteins can function before irreversible denaturation occurs. Despite extensive research, the precise determinants of how temperature impacts protein stability based on amino acid sequences remain elusive.[41] However, we can go out and inspect nature for inspiration.

Natural evolution is driven by the reciprocal interplay between genes, organisms, and the environment, resulting in a process known as adaptation.[42,43] This dynamic process has enabled life on Earth to thrive in diverse ecological niches, including those considered extreme.[44,45] Among the various ecological niches, the thermal niche has garnered significant interest from the biotech industry and biological research in general, as it offers valuable insights into thermal adaptations.[46–48] For instance, the archaeon *S. acidocaldarius* thrives in environments with pH 3 and temperatures as high as 80 °C. Investigating how such organisms ensure protein stability and function under these conditions can illuminate strategies for designing HTPs that retain their stability and activity in industrial settings.

Extremophiles are organisms adapted to conditions at or near the limits of what is biologically tenable, such as extreme radiation, pressure, pH, salinity, desiccation, and, notably for this discussion, temperature.[46–48] Thermophiles, for instance, are microbes thriving in temperatures ranging from 40 °C to about 98 °C, contrasted with mesophiles, which prefer temperatures between 20 °C and 40 °C. The study of these organisms, particularly their proteins, provides invaluable insights into protein thermal stability, given their analogous yet distinct proteomes.[49] Studies on protein thermal stability encompass a broad range of investigations, typically conducted at two levels: the folding/function level[41,50] or the combined system level[41,51,52]. Within these levels, researchers employ three major approaches to uncover the molecular basis of thermal stability.

[4] Some authors try to make a firm distinction between these two modes of studying protein thermal stability as the literature often conflates these two modes of analysis. Some authors refer to thermodynamic protein stability as thermal tolerance and kinetic stability as thermostability. Thermal tolerance is the thermodynamic stability of the protein — its ability to refold after elevated temperature has dissipated. Thermostability refers to the ability to avoid irreversible denaturation, or kinetic stability. Unfortunately, while there have been studies that taken up this admittedly useful conceptual distinctions, the confounding of these two modes remain in many studies, unfortunately.[19]

These approaches are: 1) comparative analysis between the sequences and structures of orthologous mesophilic and thermophilic pairs[49,53–55], 2) comparison of a mesophilic protein with its engineered thermostable mutant[56,57], and 3) data-driven algorithmic approaches.[58–61] Each approach has its own advantages and limitations. The first approach has been traditionally used in the field but often overlooks the evolutionary history of the protein. The second approach, although it excludes unrelated molecular adaptations, relies on context- specific protein engineering methods such as synthetic chemical modification, directed evolution, and various forms of computational protein design (rational, combinatorial, and data-driven), which can be costly and labor-intensive.[19,62,63] The third approach, which is more recent, leverages advances in machine learning (ML) and bioinformatics to successfully tackle these questions using computational tools. However, this approach heavily depends on the quality and information density of the data.[64] If the data lacks evolutionary information, the model may not generalize properly. Furthermore, data-driven approaches that aim to incorporate rational design often exhibit biases towards proteins at ambient temperatures.[65,66]

Despite the different approaches and levels of investigation, these studies converge on several factors that contribute to the thermosensitivity of proteins. These factors include amino acid composition and properties[6,49,53,67,68], protein structural features[6,47,54,55,69,70], and protein length.[71] However, it is important to note that all these factors, while influencing thermal stability, indicate that the modes by which proteins achieve stability are not independent but rather strongly linked to the protein's evolutionary history.[72] In the subsequent sections of this chapter, I will explore the gaps in our current understanding of protein thermostability, discuss various computational models derived from extant data, and introduce a pipeline that generates the largest corpus of context-dependent paired protein homologs across different temperature ranges. Lastly, I will present a novel protein model, NOMELT (Neural Optimized Machine Enabling Learned Thermostabilization), designed to induce thermally stable variations in proteins in a context-dependent manner.

2.2 CREATING THE LARGEST THERMOSTABILITY CORPUS

In the previous section, we discussed various methods for measuring protein thermal stability, highlighting the need for substantial, high-quality data to train robust deep learning models. To construct a comprehensive dataset for this purpose, we must first determine the type of stability to measure—whether thermodynamic or kinetic—and identify gaps in existing datasets.

Current datasets on thermostability, as shown in Table 2.2, are often limited in size and diversity. For instance, databases like FireProtDB focus on single point mutations, providing high-quality insights into stability changes due to specific mutations but lacking the variety found across the broader protein sequence space.[73] Other datasets utilize melting temperature (T_M) or the organism's optimal growth temperature (OGT) to infer stability trends.[74–77] While these measures offer a broader view, they introduce noise and may not accurately reflect the stability characteristics of individual proteins across different conditions.[55,78,79] Furthermore, existing datasets often overlook the contextual dependencies of protein stability. For example, the biggest current dataset of paired protein homologs includes only 1.6k unique examples and lacks comprehensive representation across evolutionary variations.[6] Such limitations stress the necessity of a context-dependent approach, pairing homologs across a range of temperatures and providing multiple evolutionary examples for each.

Table 2.2. Selected thermostability datasets

Name	Size	Size of Protein Pairs	Diversity	Label
Meltome Atlas[78]	48k	-	Diverse proteins, 13 organisms	T_M
ProThermDB[80]	32k	-	Diverse proteins and single point mutation	T_M
FireProtDB[73]	16k	-	Single point mutations	T_M
Engqvist et al.[81]	5.5 M	-	Diverse enzymes, multiple examples across evolution	OGT
TemStaPro[82]	2.5 M	-	Diverse enzymes, multiple examples across evolution	OGT
Hait et al.[6]	700	1.6k	Diverse protein pairs	OGT

To overcome these challenges and expand the available data for deep learning applications, we have adopted the organism's OGT as a preliminary proxy for thermodynamic stability. This decision allows us to leverage larger datasets while acknowledging the inherent context-specific nature of protein stabilization mechanisms. Recent advances in deep learning have demonstrated the potential of large models to discern complex patterns in protein behavior, including binding and structural predictions.[24,66,83–86] By focusing on thermodynamic stability and aiming for generalization, we seek to create a dataset that not only addresses the current gaps but also enhances our understanding of protein stability in diverse biological contexts.

In light of the successes seen in applying computational models to various biological challenges, translating these advancements to context-dependent thermal stability design

necessitates the creation of a largest thermostability sequence dataset. This section introduces learn2thermDB, a dataset comprising 69 million protein pairs, each with up to 250 amino acids, derived from 4739 mesophilic organisms (with an OGT of less than 40 °C) and 289 thermophilic organisms (with OGTs ranging from 40°C to 98 °C). To construct this dataset, we first paired mesophilic and thermophilic organisms based on evolutionary distance, then identified homologous proteins within these paired taxa. Even with a more stringent thermophilic threshold of 60 °C, our dataset retains 9 million pairs, marking a significant increase over existing databases (4-fold increase).

The dataset's vast scale includes individual proteins that appear in multiple pairings, ensuring redundancy across evolutionary space. Moreover, organisms without direct pairings contribute their proteomes, labeled by OGT, adding another 23 million proteins from mesophiles and 1 million from thermophiles for further analysis. The breadth of organisms and proteins covered in this dataset is detailed in Table 2.3, spanning a wide range of prokaryotes across the taxonomic tree as depicted in Figure 2.2.1-left and covering extensive known protein space shown in Figure 2.2.1-right for comparison.

Table 2.3. Summary table of learn2thermDB primary entities. Organism and protein pairs were identified via local alignment, thus not all taxa/proteins participate in pairing, and those that do may occur multiple times.

Number of organisms (in pairs)	Number of proteins (in pairs)	Number of organism pairs	Number of protein pairs
9,536 (5,028)	24 M (4 M)	150k	69 M

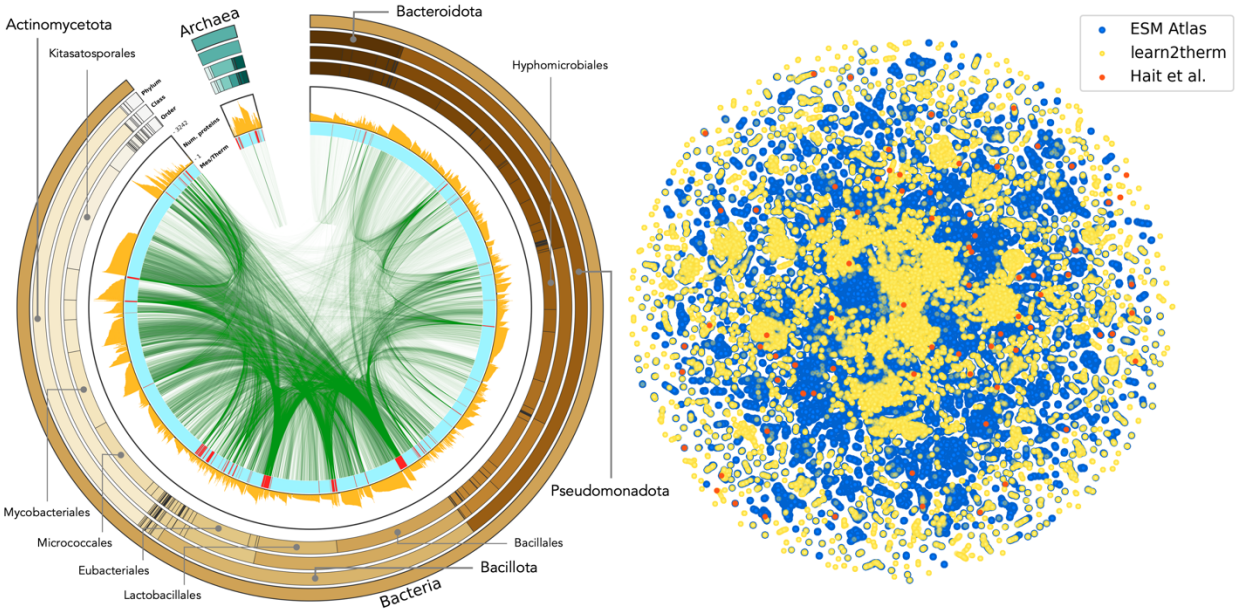

Figure 2.2.1. Coverage of taxonomic and protein space by the homologous protein pairs (N=69 million) within the dataset. Left) The dataset's NCBI taxonomic breakdown is illustrated.[87] The outer ring represents the super kingdom, phylum, class, and order, with wedge sizes reflecting the count of organisms within each classification that include at least one protein in a learn2therm protein pair. Prominent phyla and classes are labeled. The inner ring displays a histogram of proteins involved in pairs per organism, colored to distinguish mesophilic organisms in blue and thermophilic organisms in red. Connections at the center show taxonomic pairs that contribute to protein pairings. Right) A two-dimensional t-SNE mapping of sample protein space based on Evolutionary Scale Model (ESM) embeddings showcases the distribution of proteins.[22] Proteins in pairs from our dataset are shown in yellow, the existing largest dataset of temperature-crossing protein pairs in orange, and a high-confidence structural subset from the ESM Atlas in blue. The sample sizes are proportionate to the relative sizes of our proteins compared to the Atlas and the reference dataset. For mapping details, refer to the Appendix A.6.

2.2.1 *Methods Behind the Corpus*

A summary of the overall data pipeline to generate learn2thermDB is illustrated in Figure 2.2.2, and elaborated in detail in this subsection.

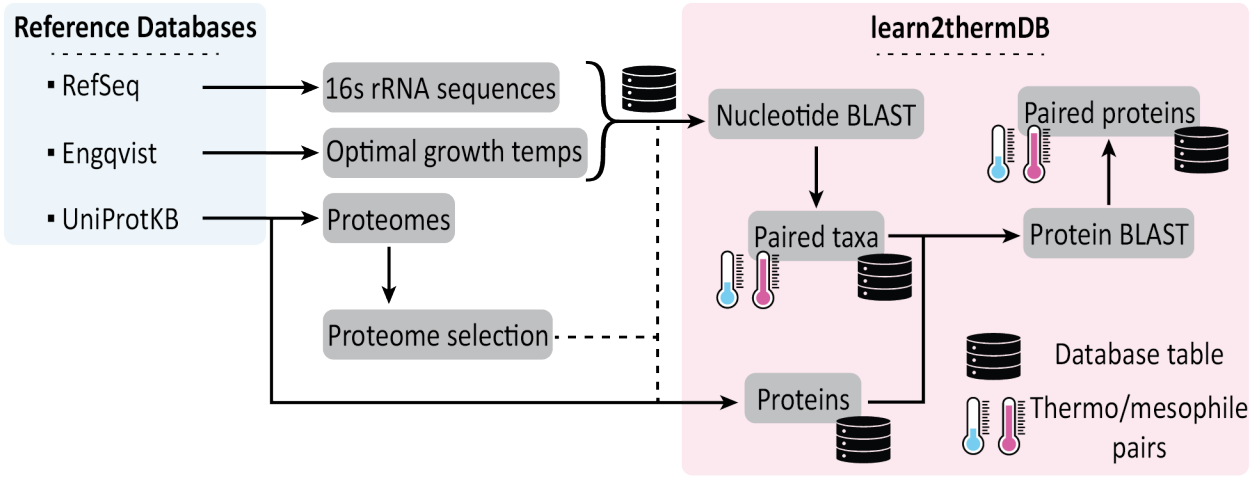

Figure 2.2.2. Pipeline for generating learn2thermDB by labeling homologous protein pairs across varying temperatures. Initial raw data sources included RefSeq16S rRNA sequences[88] and OGT labels from Engqvist[61,81], alongside UniProtKB proteome metadata[89] and protein sequences. From the metadata, a single representative proteome was selected for well-studied organisms, while ensuring inclusion of data from lesser-studied taxa. Only proteins from selected proteomes with corresponding OGT labels were retained. The search for protein pairs began by identifying related organism pairs through alignment of 16s rRNA sequences, followed by sequence alignment to find protein pairs. The final database comprises tables detailing taxa, mesophilic/thermophilic taxa pairs, proteins, and mesophilic/thermophilic protein pairs.

Archaeal and bacterial 16s rRNA sequences associated with NCBI taxonomy identifiers (taxids) were extracted from NCBI BioProjects 33175 and 33317 using the Entrez API.[90–92] We retained only complete sequences, ranging from 1300 to 1600 bases. In instances where multiple sequences were linked to the same taxid with greater than 98% identity (isoforms), only the longest sequence was preserved. OGTs, curated by Engqvist[81], were acquired and associated with the corresponding species taxid. When multiple OGTs existed for a single species, the average OGT was calculated. Protein sequences along with their respective proteome metadata were downloaded from UniProtKB[89] to ensure a comprehensive and diverse dataset. This metadata facilitated the identification of redundant entries and prioritization of proteomes based on UniProt's labels. For organisms represented by multiple proteomes, selection was based on a hierarchy established by UniProt: "Reference and representative proteome", "Reference proteome", and "Representative proteome". Proteomes labeled as "Redundant" or "Excluded" were disregarded. Proteins were

included only if they corresponded to an organism within our taxa table and belonged to the designated priority proteome. This filtering ensured a balanced representation across the dataset, avoiding overrepresentation from model organisms.

Organisms were categorized as thermophilic if their OGT was above 40 °C, with the provision to adjust this threshold to 60 °C based on specific research needs. 16s rRNA sequences of categorized organisms were aligned using a local alignment BLASTn to identify evolutionary relationships necessary for subsequent protein pairings.[93–96] Pairs of taxa with greater than 81% identity and 98.5% coverage were selected for protein homology searches. This step was crucial in narrowing the potential search space from trillions to a manageable number. For the final protein pairing, DIAMOND[97] was employed due to its efficiency and sensitivity compared to traditional BLAST. It facilitated the alignment of protein sequences, each less than 250 amino acids, across the identified taxa pairs. This process significantly reduced computational requirements while maintaining the integrity and depth of the analysis. The homologous protein pairs, along with relevant metadata such as alignment identity and coverage, were compiled into the learn2thermDB. This database now serves as a foundational resource for exploring protein thermal stability across a diverse range of taxa.

The dataset's relational nature is depicted through various visualizations, including taxonomic breakdowns and protein distribution by temperature, outlined in Figure 2.2.3-top and Figure 2.2.3-bottom, respectively. These visual aids spotlight the diversity and scale of this dataset. For detailed methodologies, including specific alignment parameters and definitions of key metrics, refer to Appendix A.2.

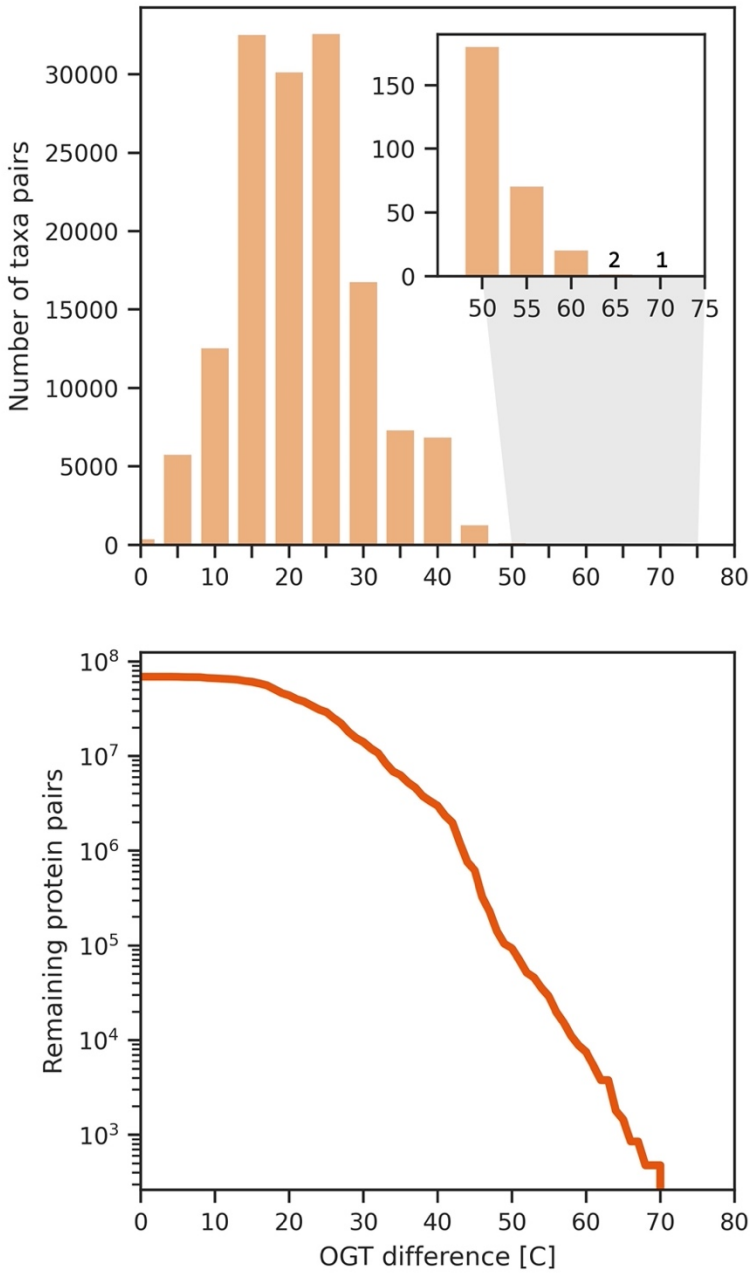

Figure 2.2.3. The distribution of learn2thermDB as a function of difference in OGT between entities. In the top, shown is a histogram of difference in OGT between pairs of organisms. The distribution is left-skewed towards 0, but there is still a good amount of data, i.e., pairs, with temperature difference >30 °C. In the bottom, shown is the count of protein pairs remaining as minimum OGT difference is increased. We still maintain 14 M protein pairs with OGT >30 °C.

The development of learn2thermDB, from the initial data extraction to downstream validation steps (detailed in the Validation of protein pairs section), was meticulously tracked using Data Version Control (DVC).[98] This approach ensures a historical record of data modifications as the code evolved and parameters were adjusted, such as maximum protein length and minimum alignment metrics. The pipeline is designed for reproducibility and can be re-executed with a single command, pending appropriate configuration of the environment and computing cluster.

The data tables generated during this process—including taxa, proteins, taxa_pairs, and pairs—were organized into a relational database using DuckDB.[99] This setup enhances data accessibility and facilitates efficient querying and filtering. A selection of tunable parameters critical to the dataset construction is documented in Table 2.4. Sample selection of tunable parameters used to generate learn2thermDB, with specific values applied to the current dataset outlined. Moreover, the computational carbon footprint was conscientiously evaluated using CodeCarbon[100], estimating the carbon cost of compute-intensive steps at 11.5 kg, with the total methodological development cost approximated at 70 kg. A representation of the database schema, detailing the relationships between the four main tables: 'taxa', 'proteins', 'taxa_pairs', and 'pairs', is depicted in Figure 2.2.4. For a more elaborate schema and detailed descriptions of each field, please see Appendix A.3.

Table 2.4. Sample selection of tunable parameters used to generate learn2thermDB

Parameter(s)	Description	Published Value
OGT Threshold	Binary threshold to split organisms into mesophiles and thermophiles.	40 °C
16S Alignment Coverage	Minimum alignment coverage (both strands) of 16s rRNA sequence to be considered a taxa pair	98.5%
16S Alignment %ID	Minimum alignment gap compressed percent identity of 16s rRNA sequence to be considered a taxa pair	81%
Protein Alignment Coverage	Minimum alignment coverage (both strands) of protein sequence to be considered a protein pair	75%
Protein Alignment E-value	Maximum E-value of protein alignment to be considered a protein pair	1e-4

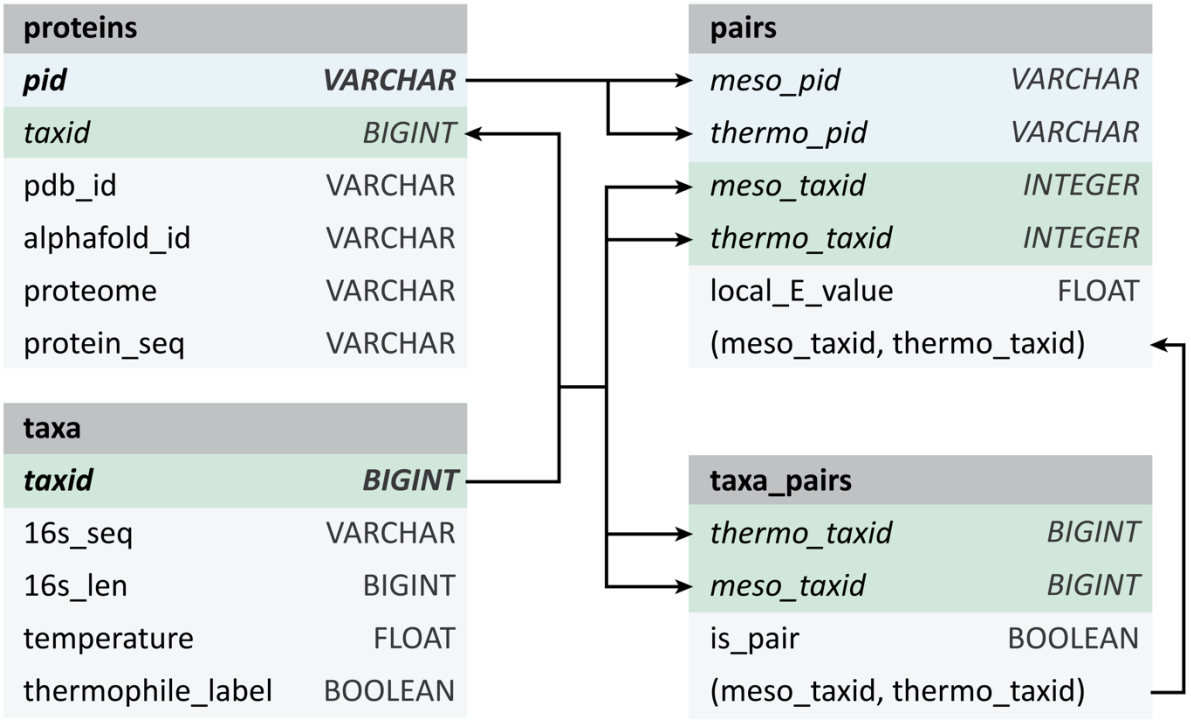

Figure 2.2.4. Condensed schema for learn2thermDB. The taxa table contains NCBI taxonomic information, corresponding 16S rRNA sequences, and OGT label. The protein table contains the amino acid sequence and cross-links to external databases, such as UniProt. Finally, the taxa_pairs and pairs table contain metrics of the local alignments for 16s rRNA sequences and protein sequences, respectively.

2.2.2 *Validation of the Protein Pairs Across Temperature*

To ensure the integrity and actual homology of the protein pairs identified across different temperatures in learn2thermDB, several validation steps were implemented. Initially, proteins were cross-referenced with existing published datasets to verify the accuracy of the assigned optimal growth temperature (OGT) labels. Further validation involved the use of remote homology Hidden Markov Model (HMM) searches, which facilitated the comparison of protein pairs by applying established protein family/domain labels.[101–103] Both structural and sequence alignments were performed, with outcomes benchmarked against those from the largest available dataset of

protein pairs. Additionally, a growth temperature predictor model was trained to discern the underlying data signal in the labeling process, ensuring that the dataset's integrity is maintained across different temperature regimes. The specifics of these validation steps are elaborated in the following paragraphs.

To ascertain whether OGT could serve as a reliable proxy for protein melting temperature (T_M)—a key measure of thermostability—the dataset's proteins were compared against established melting temperature data. Proteins from our dataset were aligned with wild-type proteins from FireProtDB[73] and the Meltome atlas[78], requiring greater than 99% coverage and identity. This comparison involved 4,640 proteins that had both internal OGT labels and external T_M labels. The analysis revealed a Spearman's correlation of 0.85 between OGT and T_M, with a p-value of virtually zero, indicating a strong and statistically significant correlation. Further statistical testing using a binomial test assessed whether T_M was consistently higher than OGT. The results, yielding a P-value of 2.68e-19, confirmed that T_M values are generally higher, supporting the use of OGT as a surrogate measure of thermostability. A parity plot illustrating the relationship between OGT and T_M is provided in Figure 2.2.5. This visual comparison underscores the trend that an organism's protein melting temperatures generally align with its growth temperatures, validating our approach in utilizing OGT as an effective stand-in for direct thermostability measures.

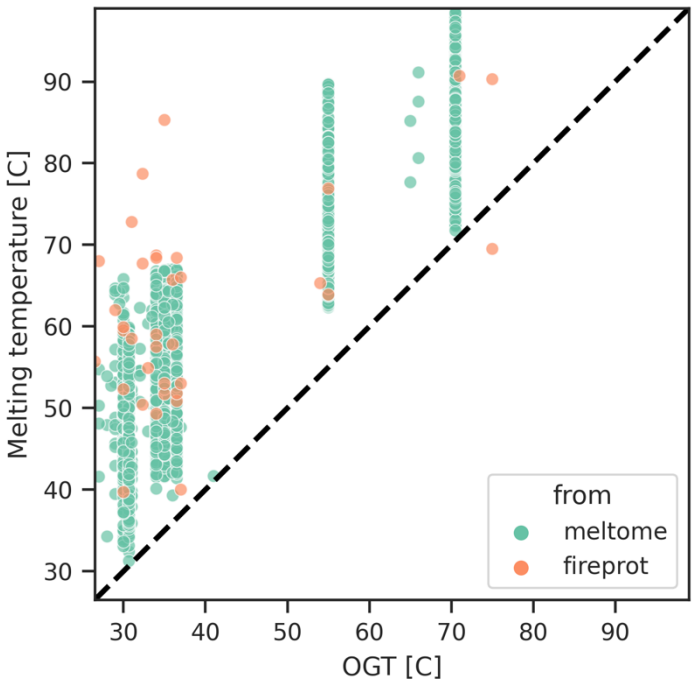

Figure 2.2.5. A parity plot of OGT as a function of melting point temperature (T_M) using data from two third-party databases. The dashed black line corresponds to identity, and almost all examples, fall on the side of T_M > OGT. Melting point temperature being greater than OGT is substantiated, with a Spearman's of 0.85 (P value 0.0), and a binomial test of >99% chance of passing with alternative P-value 2.68e-19.

In benchmarking our dataset, we compared it to the largest existing dataset of functional protein pairs across temperature, comprising 1.6k pairs, curated by Hait et al. from PDB structures.[6] To perform this comparison, we aligned Hait's dataset using DIAMOND and then assessed the alignment metrics against those obtained from our dataset. Our data exhibited alignment scores that were statistically similar or superior to this baseline, as detailed in Table 2.5. Percent identity and normalized bit scores from these alignments are depicted in Figure 2.2.6, showing both our results and the baseline for comparison.

Table 2.5. Contrastive statistical analysis of learn2thermDB and Hait's, et al. dataset

Score	Statistical Test	Probability (Cov >75)	Probability (Cov >95)
Normalized Percent Identity	Left T	1.75e-6	6.06e-155
Normalized Bit Score	Left T	1.94e-8	9.47e-117
Pfam Jaccard	Left T	0.99	3.24e-13
Structure P score	Bionomial P < 0.001	<0.001	<0.001

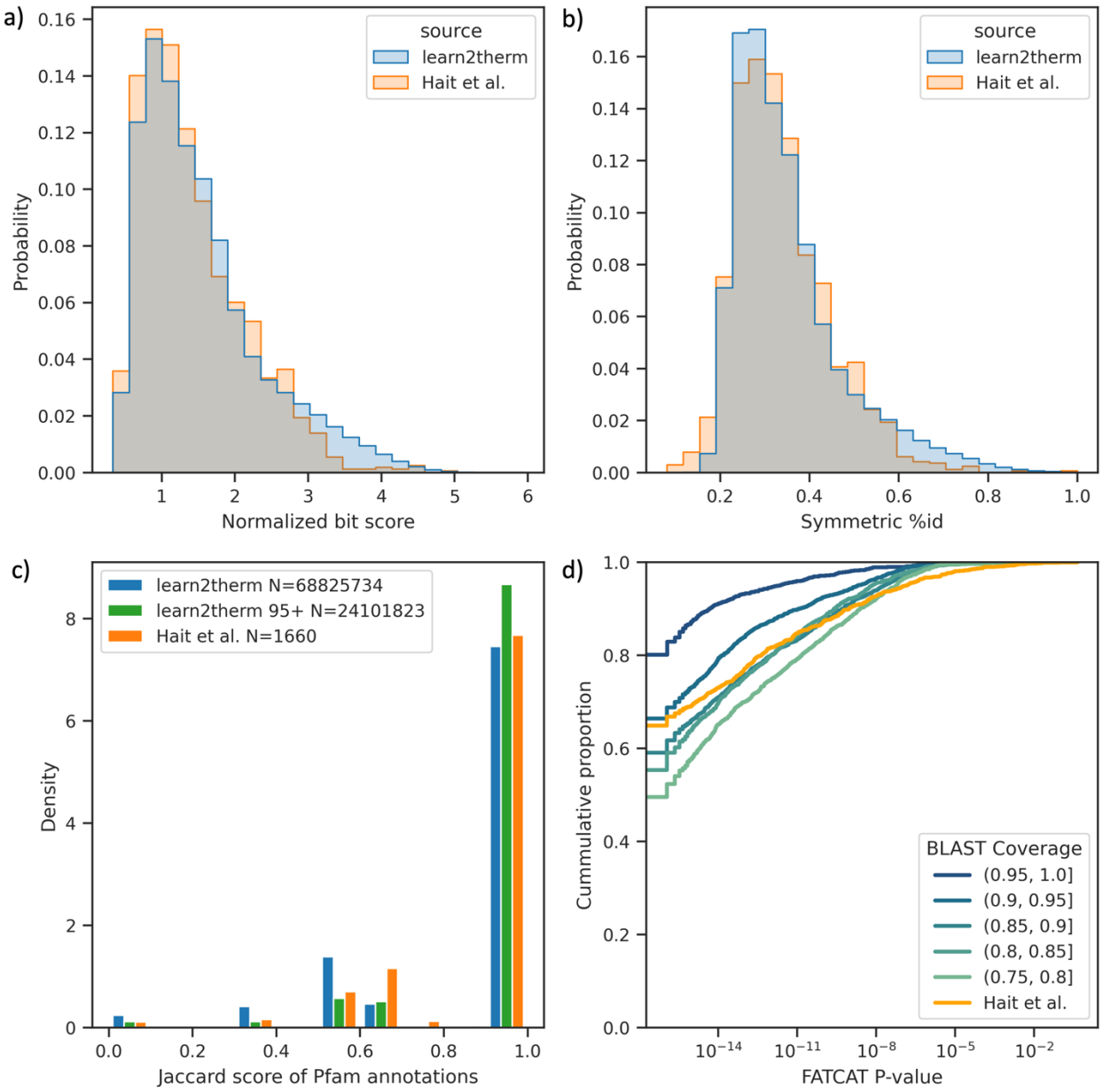

Figure 2.2.6. Comparison of protein pair quality between our dataset and Hait et al.'s dataset of 1,660 protein pairs.[6] (a) Empirical distribution of local alignment homology, presented as a bit score normalized to the average length of both protein strands. Our data shows a statistically significant rightward shift in scores, with a t-test probability 1.94e−8. (b) Similar to (a), but for percent identity. Our data again demonstrates a rightward shift in scores, with a t-test probability of 1.75e−6. (c) Empirical distribution of Jaccard scores for Pfam annotations, comparing our dataset (blue) to the baseline dataset (orange). Generally, our full dataset exhibits more annotation mismatches. However, when considering only the 25 million protein pairs with

BLAST coverage greater than 95%, the Pfam annotations become statistically indistinguishable from those of the baseline, with a t-test probability of 3.24e−13. (d) Cumulative distribution of FATCAT structural alignment P-values, for bins in BLAST coverage sampled uniformly from our dataset and compared to baseline structural alignments. Pairs with even low coverage are statistically more likely to exhibit P-values less than one in a thousand, with binomial confidence exceeding 99%.

To further validate our dataset, we annotated proteins using Pfam (version 35.0) for both learn2thermDB and Hait's dataset, retaining matches with an E-value below 1e-10. This annotation covered 86.1% of learn2thermDB proteins and 99.8% of Hait's proteins, indicating that our dataset includes novel proteins not extensively represented in Pfam. The quality of protein pairs was evaluated based on the Jaccard score of Pfam accession annotations, considering only those pairs where at least one protein was annotated. While there were more annotation mismatches in learn2thermDB, the annotation quality of pairs with over 95% sequence alignment coverage matched that of the baseline, supported by a t-test probability of 3.24e-13. The distributions of these scores for both datasets are compared in Figure 2.2.6-c.

Structural alignments were performed using FATCAT2[104] on the PDB structures of baseline pairs[105], selecting Chain A for alignment when available. This flexible alignment algorithm provides a scaled P-value indicating the likelihood of the alignment score occurring by chance, with values close to 0.0 signifying quality structural overlap. For learn2thermDB, where PDB structures were not available, predicted structures from AlphaFold2 were used.[106] Due to computational constraints, only a subset of 10,000 pairs was analyzed, selected uniformly from five bins of sequence alignment coverage ranging from 75% to 100%. The cumulative distribution of FATCAT2 probability scores for each subset is presented in Figure 2.2.6-d, demonstrating that alignments in learn2thermDB are statistically less likely to occur by chance than those in the baseline. A binomial test showed that even pairs with less than 80% sequence alignment coverage were comparable to the baseline, with those having higher coverage typically achieving better structural alignment scores, affirming the high quality of our dataset with over 99.9% confidence.

2.3 DATA SIGNAL IN LEARN2THERMDB

To assess the utility of our dataset for training deep learning models, we implemented a classification task to distinguish between mesophilic and thermophilic proteins based solely on their amino acid sequences. Previous studies have demonstrated the feasibility of learning such distinctions.[82,107–110] Our preprocessing pipeline included binarization of proteins by OGT (<30 °C as mesophilic and ≥60°C as thermophilic), class balancing, deduplication of similar sequences, and splitting according to the NCBI taxonomy of host species. This resulted in a division into training and test sets comprising 290,000 and 28,000 proteins, respectively (detailed preprocessing steps are provided in Appendix A.5). We evaluated the performance of TemStaPro, a recent predictive model, on our test set of 28,000 proteins.[82] TemStaPro, an ensemble of neural networks layered on ProtT5XL embeddings, categorizes temperature classes in 5 °C increments and checks consistency across its ensembles. On our dataset, 95.5% of predictions were consistent, with an accuracy of 91% in classifying the correct temperature range, significantly outperforming the null model's 61% accuracy. The distribution of these predictions is illustrated in Figure 2.3.1.

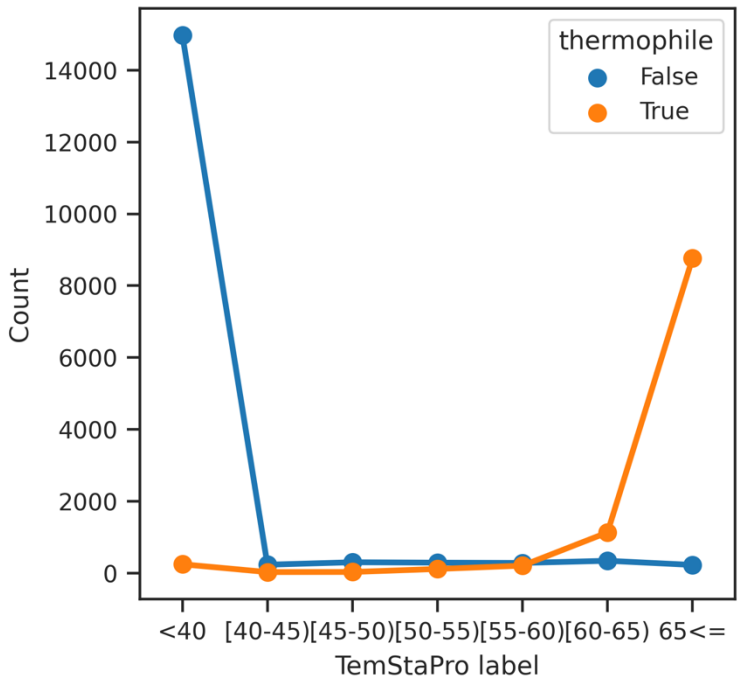

Figure 2.3.1. Distribution of Predicted Temperature Classes by TemStaPro[82] for Thermophilic and Non-Thermophilic Proteins. This figure displays the classification of proteins

by the TemStaPro model into temperature bins, specifically for proteins categorized as thermophilic (≥60 °C, orange) and non-thermophilic (<30 °C, blue). The x-axis represents the temperature bins used by TemStaPro, while the y-axis shows the count of proteins in each category. The model demonstrates strong performance with few predictions in intermediate ranges where no data exist, achieving 91% accuracy in distinguishing actual temperature classes.

Considering the potential for data overlap, as TemStaPro was trained on UniParc proteins which might include sequences from our dataset, we conducted an additional test to rule out data leakage. We fine-tuned ProteinBERT, another Large Protein Language Model (LPLM), on our development set.[111] The model's training parameters and architecture details are available in Appendix A.5, and its performance on a held-out test set is summarized in Table 2.6. This model also demonstrated high predictive accuracy, validating the presence of significant biological signal in our dataset.

Table 2.6. Summary of performance of the test set on the fine-tuned LPLM

Balanced Accuracy	F1 Score	Matthew's Correlation	Confusion Matrix
92.5%	0.91	0.85	[16069, 1459, 671, 10296]

The results confirm not only that our dataset supports basic machine learning tasks like thermophilic classification but also possesses sufficient resolution for advanced models like transformers to differentiate between low and high temperature proteins. Additionally, by incorporating homologous pairing, our data can facilitate the training of a sequence-to-sequence translation model. To minimize the carbon footprint, we leveraged ProtT541, a foundational protein encoder-decoder transformer model trained in a self-supervised manner to reconstruct protein sequences. ProtT541 is particularly suited for causal language generation tasks right out of the box.[111]

2.4 TRANSLATING THERMOSTABILITY VARIANCE FROM LOW TO HIGH

2.4.1 *Data Preparation for the Themostability Translator Training*

The learn2thermDB dataset underwent filtering based on results from the OGT predictor, retaining only those protein pairs where the mesophilic temperature was ≤40 °C and the thermophilic temperature was ≥60 °C. Further refinement ensured that only pairs with greater than 95% sequence alignment coverage and a sequence length discrepancy of no more than 10% were preserved. This selection process yielded 4.7 million high-quality protein pairs. Given the redundancy and high sequence similarity among homologs, and the need for a translator that functions effectively across diverse proteins, a conventional random data split would risk validation and test set leakage, thus skewing model evaluation. Proteins sequences, unlike data in typical machine learning tasks like image processing, are not independently and identically distributed due to their evolutionary relationships. Therefore, we employed a more sophisticated splitting strategy using the source organisms to prevent the model from trivially learning sequence-temperature correlations based on organismal data. The dataset was clustered using MMSeqs2 at 50% identity through a cascading connected component method, creating 85,000 clusters (detailed parameters in Appendix B.1).[112] This clustering helped ensure that train and test sets were not evolutionarily biased towards each other, a common challenge in molecular datasets. The mean E-value for test sequences aligned against training sequences was 1.6e-4, as determined by BLAST alignments, aligning with splits reported in other studies[113] (see Appendix A.2 for BLAST parameters).

To further ensure robust model evaluation, the validation and test sets were each confined to include only a single random protein from each cluster, mitigating the risk of overrepresentation by large clusters. Due to the high computational costs of model validation and BEAM searches, only 1,000 sequences from each set were ultimately retained. These selected sequence pairs are designated for scoring the model's performance in the next section. The comprehensive dataset, however, was utilized to train a final model variant for broader downstream engineering applications discussed in section 2.5.

2.4.2 *NOMELT Training*

The hyperparameters for training the NOMELT (Neural Optimized Machine Enabling Learned Thermostabilization) model, both for test evaluation and the final model training, are detailed in Table 2.7. These parameters were selected following an exploration phase where a few key hyperparameters were varied in low throughput single-point tests due to computational constraints. Tests included adjustments in label smoothing, reweighing examples for the loss function by inverse MMSeqs cluster size, and selectively freezing early layers in the model. While most tested hyperparameters had minimal impact, adjustments in the clustering parameters significantly influenced outcomes. Freezing the first 20% of layers slightly worsened the loss (from 1.77 to 1.81) but reduced computational costs, proving a worthwhile trade-off. The final model settings reflect a balance between performance and efficiency, with further details on the specific impacts of these choices depicted in the cross-entropy validation loss in Figure 2.4.1. The entire data preparation and model training process was meticulously documented using Data Version Control (DVC), and a complete list of adjustable hyperparameters, along with the PyTorch configuration of the T5 model, is available in Appendix B.2. The training regimen required approximately nine hours on eight NVIDIA A40 GPUs configured with DeepSpeed ZeRO-3, resulting in an estimated carbon footprint of 4.8 kg CO_2 equivalent, based on the Washington State energy grid's specifications.

Table 2.7. Summary of hyperparameters for NOMELT. *Note: Epochs were determined by the early stopping point of the Eval Model

Hyperparameter	Value	
	Eval Model	Full Model
	Data Hyperparameters	
Meso/Thermo OGT Window	40 °C/60 °C	40 °C/60 °C
Minimum Sequence Coverage, Both Strands	95%	95%
Max Sequence Length Difference	10%	10%
MMSeqs2 %ID Threshold	50%	-
Data size (train/val/test)	4 M/1000/1000	4.7 M/0/0
	Model Hyperparameters	
Model Parameter Count	2.8 B	2.8 B
Frozen Layers/Trainable Layers Per Encoder and Decoder, from Bottom	4/20	4/20
Effective Batch Size	1120	1120
Max Learning Rate	1e-4	1e-4
Learning Rate Schedule	Linear, 10% Warm up for 1 epoch	Linear, 10% Warm up for 1 epoch
Parameter Precision	bf16	bf16
Optimizer	Adam W (0.9, 0.9999)	Adam W (0.9, 0.9999)
Loss	CrossEntropy	CrossEntropy
Label Smoothing Factor	0.001	0.001
Dropout Rate	10%	10%
Early Stopping Threshold	0.01	-
Early Stopping Patience	4	-
Epochs	0.18	0.18*

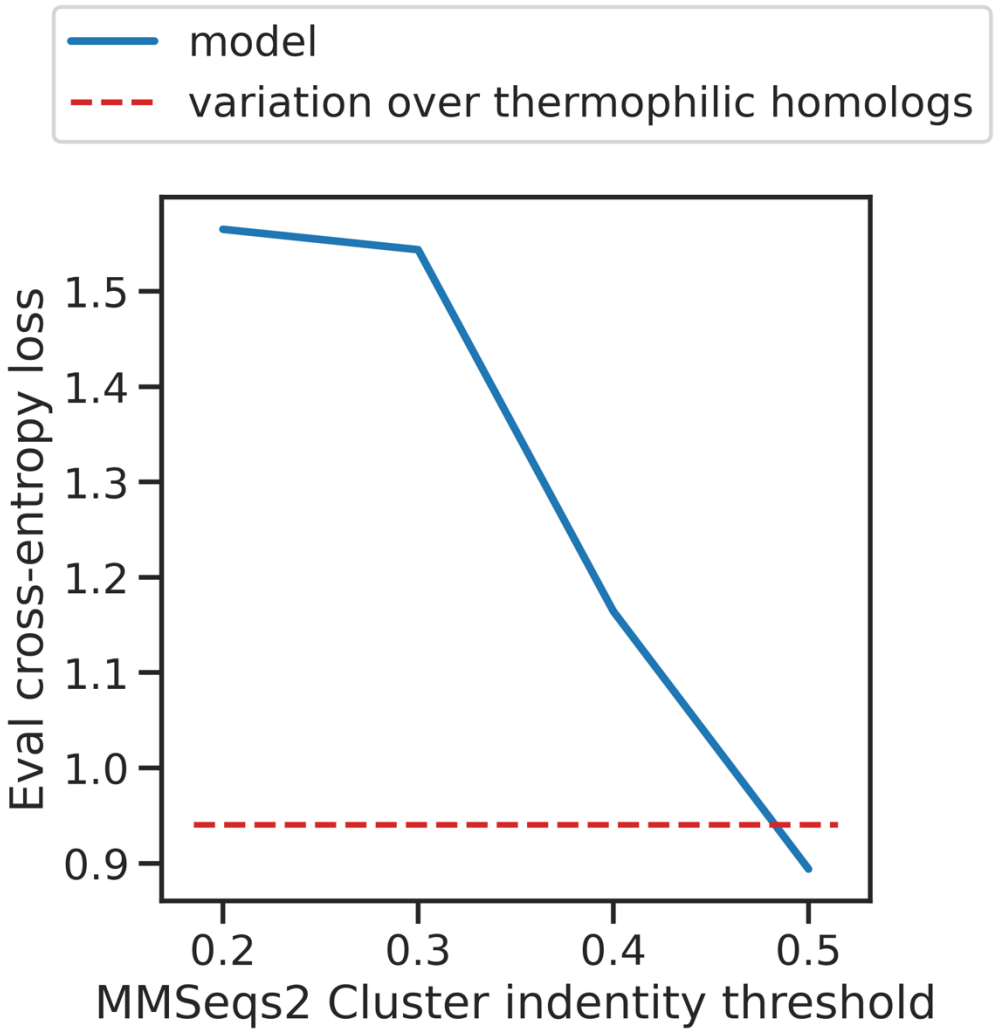

Figure 2.4.1. Validation set loss of trained models plotted against the MMSeqs2 clustering percent identity threshold, shown in blue. Model performance reaches parity with natural variation only when the identity threshold is reduced to 50%. This suggests that while the model does not need an MSA of high-temperature homologs for variation, it does require similar examples in the training set to achieve optimal performance

The NOMELT model's training objective is to generate a thermophilic homolog from an input mesophilic protein sequence, constructing the amino acid sequence from the N-terminus to the C-terminus. Given the variability within protein families—where functional similarity can persist despite as low as 20% sequence identity—and the potential for residues along the protein to exhibit diverse amino acid distributions, including gaps and insertions, the model does not aim for exact replication of thermophilic sequences.[114] Instead, it learns from a dataset where each

mesophilic or thermophilic sequence may be paired with multiple homologs, as illustrated in Figure 2.4.2. This figure displays the probability density of sequence occurrences in the training set, noting that while many proteins form unique pairs, thermophilic sequences often align with multiple mesophilic counterparts, reflecting significant and desired redundancy. Specifically, each thermophilic sequence is paired with an average of 103.6 mesophilic sequences, and inversely, each mesophilic sequence corresponds to approximately 7.1 thermophilic sequences. Such redundancy enriches the model's learning across evolutionary variations, enhancing its ability to generalize across diverse protein forms.

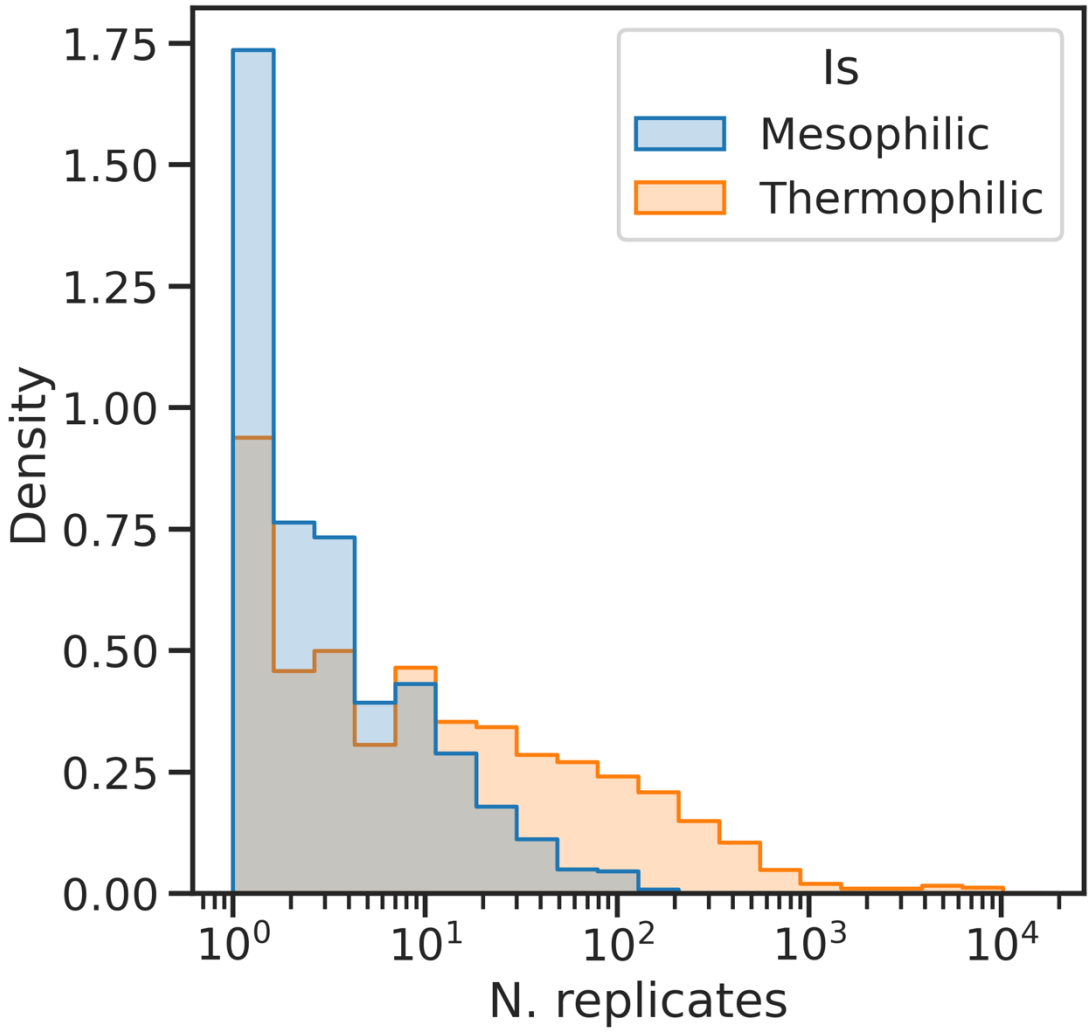

Figure 2.4.2. Density of replicate counts for mesophilic and thermophilic sequences in the training set. While the most common number of replicates for both input and target sequences is one, most proteins appear multiple times, each paired with a different homolog. The distribution

of thermophilic replicates is right-skewed, reflecting the identification of homologous pairs from a larger pool of mesophilic sequences compared to a smaller set of thermophilic ones

To assess the model's efficacy in generating thermophilic counterparts, we utilized a rigorously curated test set. This set comprised protein clusters defined by 50% sequence identity, ensuring that sequences similar to those in the training set were excluded from testing to prevent data leakage. Additionally, to mitigate bias towards overrepresented protein families, only one sequence per cluster was included in the test set, as detailed in the previous subsection. Several metrics were calculated to evaluate the model's performance, which are summarized in Table 2.8. The model's test loss was computed using a causal approach with teacher forcing applied to prior amino acids. All other metrics, including accuracy and sequence fidelity, were derived from comparing the model's outputs against sequences generated through a BEAM search over autoregressive sequence translations.[115] This process produced a single predicted sequence for each input.

Table 2.8. Summary table for NOMELT to recapitulate the test set. “Resiude” type was computed via amino acid basis, “Sequence” was computed using the full protein sequence, and finally “Structure” was found via comparing ESM-predicted structure. *Note: this does not utilize the NOMELT model, but instead natural variation over thermophilic homologs

Test Metric	Value	Type
Cross Entropy Loss	0.90	Residue
MSA Cross Entropy Loss	0.94	Natural*, Residue
Transcription Error Rate	47%	Sequence
Sequence Identity	43%	Sequence
Bits Per Resiudes, BLOSSUM62	2.4	Sequence
Jenson-Shannon Secondary Structure	0.01	Structure
FATCAT Structural Alignment P-value	0.01	Structure

The results indicate that the model typically generates sequences that vary by ±2.3% in length relative to the actual thermophilic sequences. The transcription error rate stood at 47%, meaning that in 53% of cases, the model correctly placed the exact amino acid in the appropriate position. It is important to note that this error rate does not account for homologous or analogous amino acids, nor does it consider gaps and deletions; rather, it is normalized against the total number of residues in the test set rather than the number of sequences. To further contextualize the model's performance, we leveraged known natural variations to qualitatively benchmark against the test loss values. This comparison helps to ground the model's capabilities in translating mesophilic sequences to their thermophilic counterparts within the context of observed natural protein variability.

Moreover, to evaluate the model's capacity to replicate thermophilic sequences, a Multiple Sequence Alignment (MSA) was constructed for each thermophilic target using jackhmmer.[116] This analysis revealed an average cross-entropy of 0.94 across the natural variation of thermophilic residues, indicating the typical uncertainty when predicting amino acids independently based on natural distributions. Remarkably, the model's categorical cross-entropy loss on test data registered at 0.90, suggesting a slight advantage in predicting accurate amino acid positions over random sampling from observed distributions in thermophilic homologs. This demonstrates that the model effectively learns from a single mesophilic input sequence without needing an MSA of other thermophilic counterparts.

For a holistic evaluation of the model's output, each generated sequence was aligned to its corresponding true thermophilic sequence. These alignments exhibited a wide range of identities, from as low as 4.5% to 100%, with an average identity of 43% and bits-per-residue value of 2.4 based on the BLOSUM62 scoring matrix.[117] Notably, only 19% of generated sequences deviated more in identity from both their mesophilic and thermophilic counterparts than these counterparts did from each other. Additionally, secondary structures of the generated and actual thermophilic sequences were analyzed using pyDSSP following ESMFold-predicted structures.[22,118] The Jenson-Shannon divergence for distributions across helix, strand, and loop structures averaged at 0.01, underscoring the model's fidelity in replicating expected secondary structures.

Lastly, 3D structural alignments performed using FATCAT on ESMFold-predicted structures for both generated and actual thermophilic sequences yielded a mean P-value of 0.01 and a maximum of 0.1.[104] This indicates a structural closeness between the generated sequences and their thermophilic targets, affirming the model's capability not just at a sequence level but also in preserving the complex three-dimensional structure essential for functional activity.

2.5 Results and Discission of the NOMELT model

2.5.1 *NOMELT Recapitulates Known Thermophilic Amino Acid Propensities*

Protein thermostability researchers have long endeavored to decipher the mechanisms behind the high temperature stability of thermophilic proteins, recognizing that while stability is highly sequence-dependent, certain amino acid distributions are consistently altered in thermophiles compared to mesophiles.[47,77] Our model replicates this shift in amino acid propensities, aligning closely with established observations.[67] As illustrated in Figure 2.5.1, the amino acid prevalence in model-generated sequences mirrors the shifts reported in literature, though deviations in magnitude for some amino acids are noted due to our broader dataset of 16 proteomes.[67] Notably, the model exhibits a significant propensity to substitute Valine with Isoleucine, where the latter appears more frequently than in the test set. This substitution, while deviating from the expected literature values, underlines the model's nuanced understanding of amino acid replacements conducive to thermal stability. Further development of interpretative deep learning techniques would prove fruitful to enable us to probe exactly the model's understanding or lack there of.

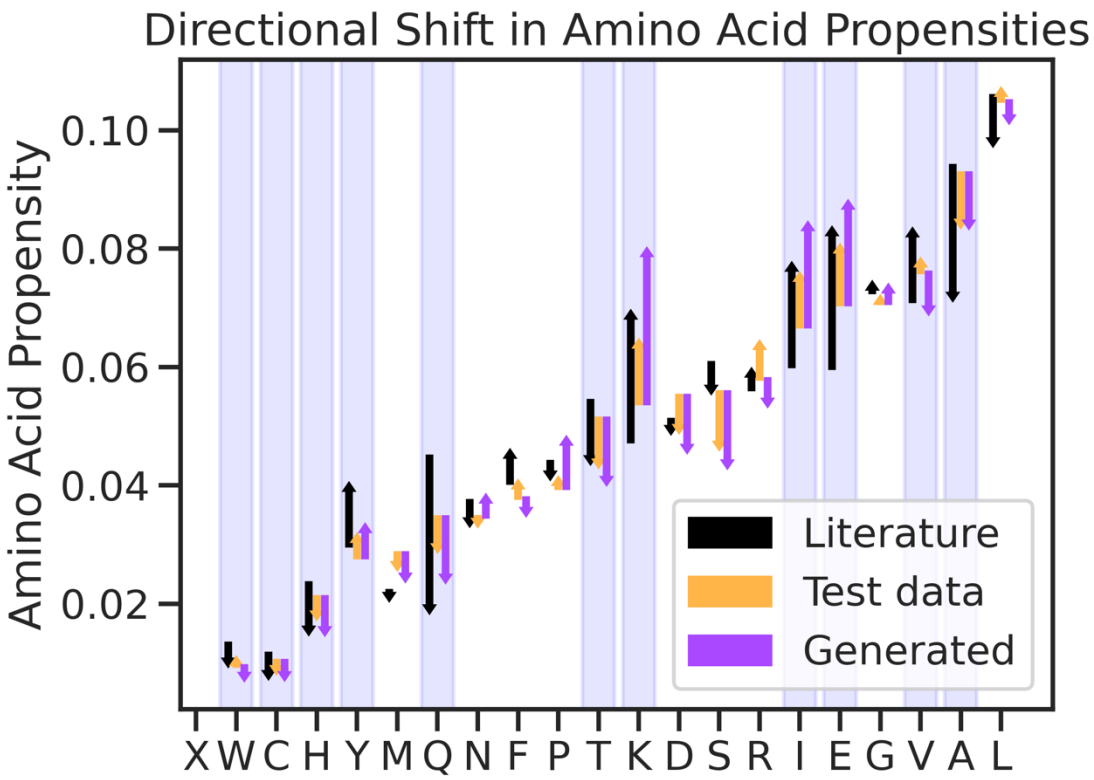

Figure 2.5.1. Change in amino acid frequencies between mesophilic and thermophilic proteins. Data from 16 proteomes in literature[67] are shown in black, test set data in orange, and model-generated sequences in purple. Statistically significant shifts, highlighted in blue, generally align in direction and magnitude with shifts identified in reference proteomes. The model-generated sequences closely replicate the observed distribution.

Furthermore, the role of disulfide bonds in enhancing thermophilic stability, particularly in archaea, is well-acknowledged.[119,120] We evaluated the model's predictions for cysteine placements necessary for disulfide bond formation. The results, significantly supported by a T-test (P-value=8.5e-22), show that the model preferentially predicts cysteine residues that contribute to disulfide bonding rather than those that do not, suggesting a deep learning-driven recognition of their structural importance, as depicted in Figure 2.5.2. The positions for these bonds were estimated using a 7.5Å distance criterion between alpha carbons, based on the predicted structure by ESMFold.[121]

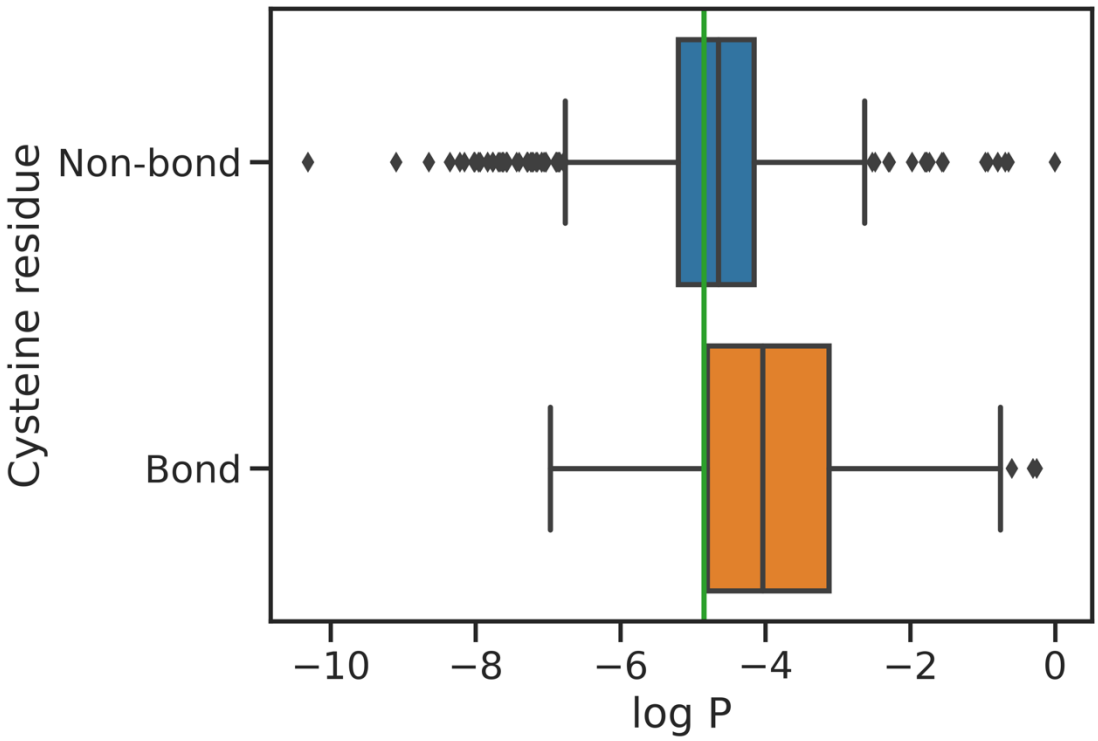

Figure 2.5.2. Model-predicted log likelihood of cysteine residues at positions likely (or unlikely) to form a disulfide bond with another cysteine (Cα distance < 7.5Å).[121] In green, the log likelihood assuming a random uniform distribution of amino acids. For positions where cysteine does not form a bond, the model shows minimal bias, variably predicting cysteine or other amino acids. Conversely, at positions likely to form a disulfide bond, the model demonstrates a strong bias towards predicting cysteine.

In addition to amino acid distribution and structural predictions, the model's ability to enhance thermophilic traits was directly tested. Using the mAF-min method[122], adjusted for sequence length, we compared the estimated stability of model-generated thermophilic sequences against their ground truth thermophilic counterparts and the original mesophilic sequences. Both model-predicted and actual thermophilic sequences displayed statistically significant increases in stability, reinforcing the model's capability to imbue mesophilic proteins with thermophilic properties (see Figure 2.5.3).

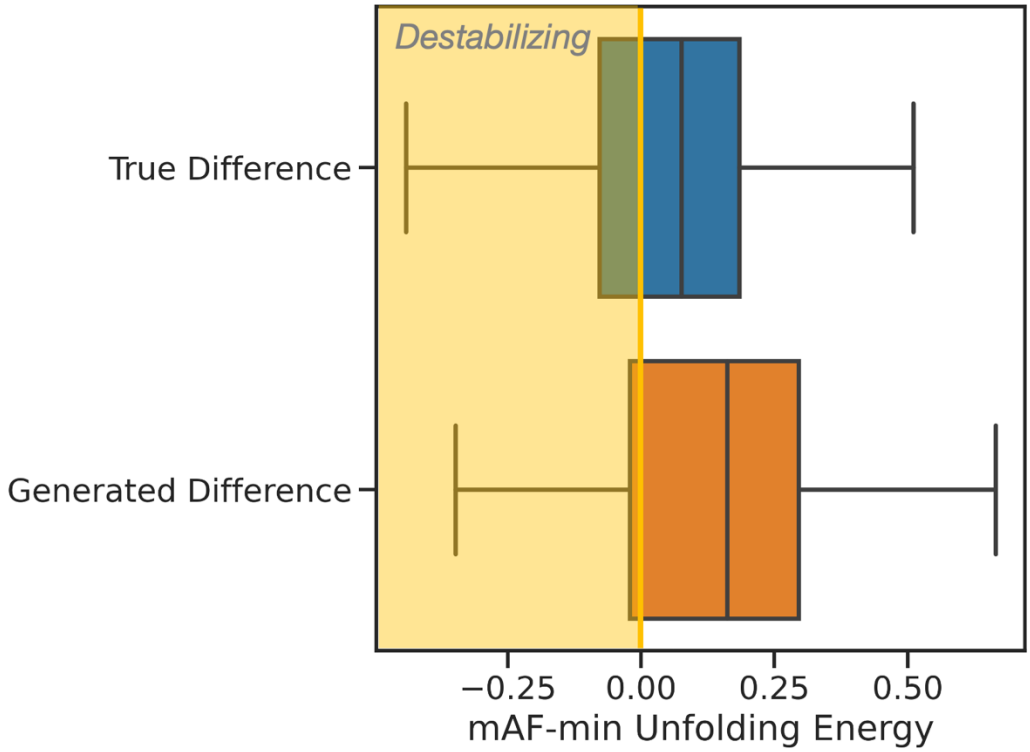

Figure 2.5.3. Shift in stability from mesophilic to thermophilic proteins, evaluated using the mAF-min method. Ground truth thermophilic sequences were stabilizing on average, with 56% of examples in the dataset exhibiting over 95% confidence in high folding free energy change. Model-generated sequences on the test set also displayed increased stability, with 72% achieving the >95% confidence interval.

While the model successfully captures and applies thermophilic stability traits, it must be noted that the natural thermophilic proteomes it learned from contain evolutionary adaptations that extend beyond mere thermal stability. Consequently, some mutations suggested by the model might not directly contribute to thermal stabilization but are part of broader adaptive traits. This complexity highlights the challenges and opportunities in leveraging deep learning for understanding and designing proteins capable of withstanding extreme temperatures. A pangenomic and ecologically-informed investigation would be useful areas to explore to attempt to tackle these challenges.

2.5.2 *Engineering Thermally Stable Variants and Validating via Dynamics*

The NOMELT model, detailed in the 'NOMELT training' section, represents a novel sequence-to-sequence approach specifically developed to translate protein sequences from ambient to high-temperature regimes. This generative model is designed not only to fill a unique niche in protein engineering but also to provide a generalized solution for designing proteins with enhanced thermal stability.

The effectiveness of NOMELT was tested using the Engrailed Homeodomain (En-HD) as a case study.[123] En-HD, a well-studied eukaryotic transcription factor, is known for its rapid simulation properties in molecular dynamics, facilitating the estimation of thermal stability. It has previously been engineered to achieve a melting temperature exceeding 98 °C from a wild type temperature of 56 °C.[124] Figure 2.5.4 illustrates the structure of En-HD, which has no close homologs in the NOMELT training dataset, as confirmed by a BLASTp search that returned the best hit with an E-value of 1.6 and 40% coverage (detailed BLAST parameters are available in Appendix A.2).

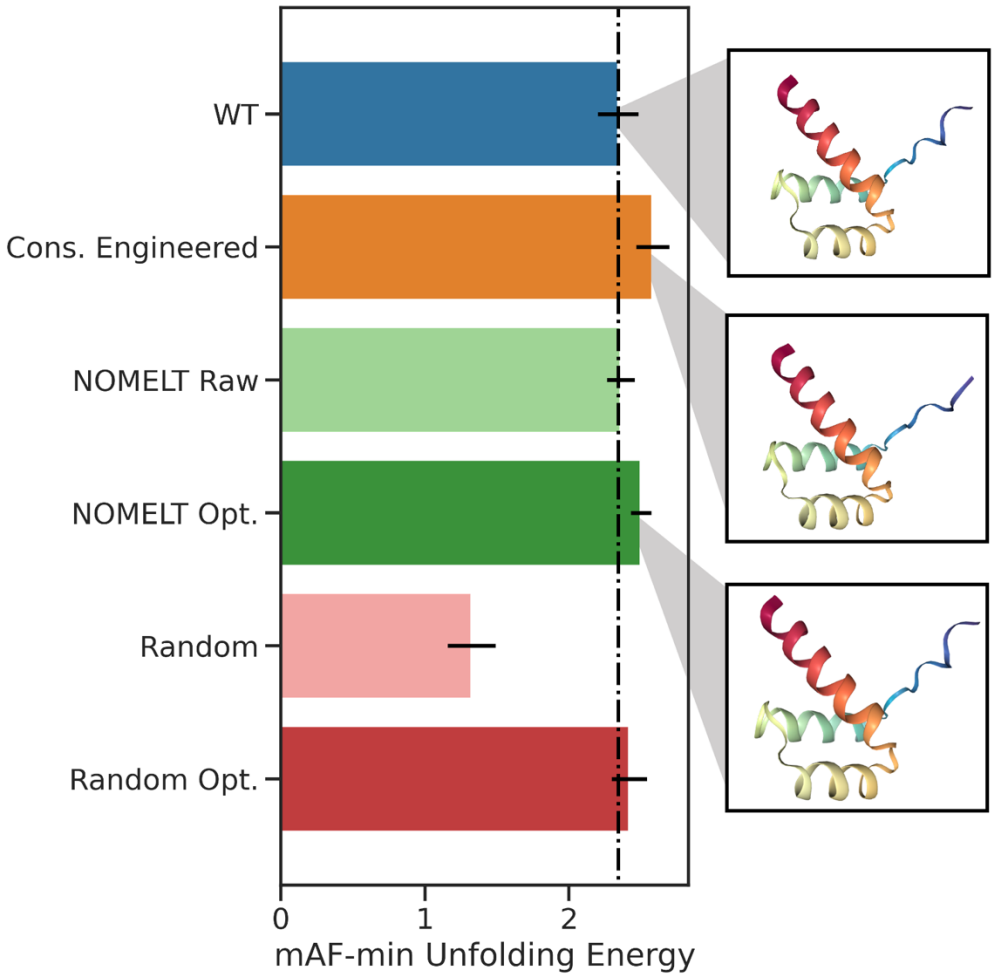

Figure 2.5.4. Comparison of protein stability across various designs using the mAF-min method's unfolding free energy change. Error bars represent the 95% confidence interval for AlphaFold structure ensembles. The wild type's score is marked by a vertical black line. A previously engineered variant (orange), optimized via consensus across many homologs, exhibits increased stability (P-value 5.9e-18). The initial output from the NOMELT model (light green), which includes 14 mutations with some insertions based on a single wild-type input, shows neither stabilizing nor destabilizing effects. However, after exploring only 100 examples from a possible 2^{14} mutation permutations using NSGA-II (dark green), a variant statistically more stable than the wild type is achieved (P-value 9.5e-14). Conversely, introducing the same number of mutations randomly (light red) proves extremely destabilizing, and equivalent exploration in a random mutation space does not enhance protein stability over the wild-type

When fed En-HD as a mesophilic input, NOMELT, utilizing a BEAM search with a temperature setting of 1.0 across 10 beams, suggested 14 mutations (including insertions and deletions) relative to the wild type. These mutations were aligned using a BLOSUM62 Smith-Waterman alignment, details of which are provided in Appendix B. The model's raw output yielded a stability score (mAF-min) of 2.36, comparable to the wild type's score of 2.34, indicating no significant increase in estimated stability. In contrast, a previously engineered variant of En-HD, optimized through consensus methods over multiple homologs, demonstrated a higher stability score of 2.58.[124]

This result highlights NOMELT's role as a source of potential variations rather than as a definitive tool for single-pass protein engineering. The variations introduced by NOMELT, derived from natural evolutionary adaptations, may not always directly enhance thermal stability but contribute to a broader understanding and exploration of protein design possibilities. Consequently, a library of 16,384 potential variations was generated based on 14 binary mutation options, providing a vast resource for further exploration and optimization in pursuit of thermally stable proteins.

The inferred improvements in protein stability provided by NOMELT, based on the mAF-min estimates, necessitate further validation given the reliance on the accuracy of this method. To more rigorously assess these results, Replica Exchange Molecular Dynamics (REMD) simulations were employed, a technique extensively used to study the thermal stability of proteins such as the En-HD.[125,126] For this purpose, five replicates of 1-microsecond simulations were conducted at various temperatures for the wild type En-HD, the NOMELT-optimized variant, and a previously engineered variant designed to withstand temperatures up to 98 °C.[125] Each simulation's root mean square deviation (RMSD) was recorded, and the results were averaged to yield a distribution of five time-averaged RMSD values per protein per temperature. Notably, the initial structure for the NOMELT variant, used prior to equilibration, was predicted by AlphaFold2.

Figure 2.5.5 illustrates the average RMSD relative to the baseline temperature of 298 K across varying temperatures. The data reveal that the literature-engineered variant, referred to as

'UVF', maintains a consistent structural alignment with its initial configuration up to 370K.[127] In contrast, the wild-type protein begins to deviate significantly from its starting structure at its melting temperature of 325K, as evidenced by an expansion in the distribution of RMSD values and a marked increase in magnitude. While the NOMELT variant does not achieve the exceptional stability of UVF—which was the result of extensive human and computational efforts—it notably retains its structural integrity up to 340K before showing similar signs of instability.

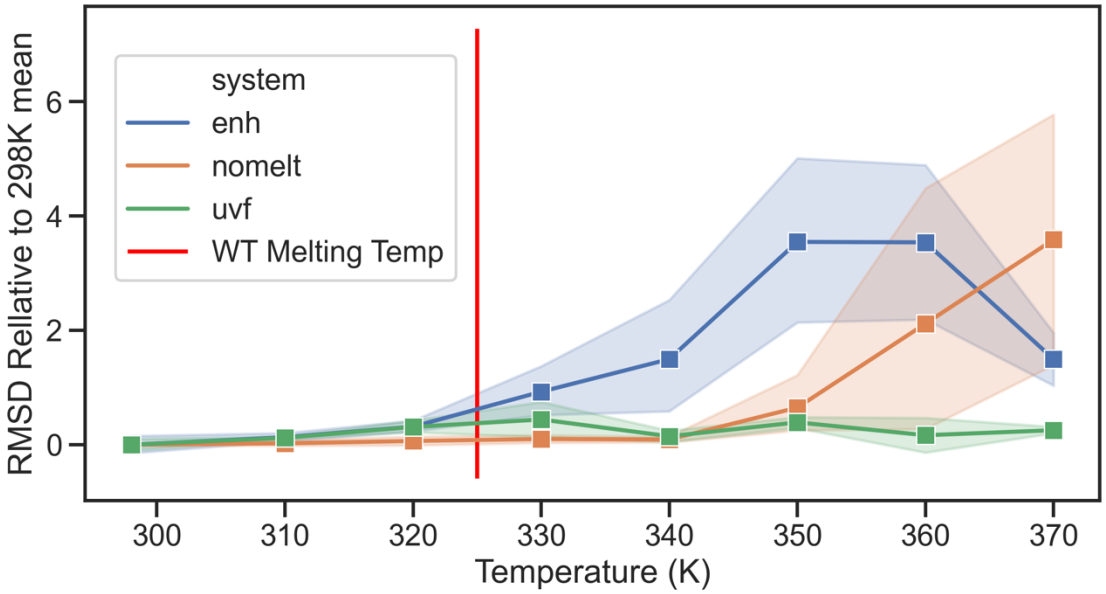

Figure 2.5.5. RMSD trends over 1 microsecond dynamics simulations of En-HD variants, relative to the average RMSD during the 296K simulations. The distributions are based on five independent simulations. The UVF variant, with an experimentally determined melting temperature of over 98 °C, maintains close adherence to its initial structure across all temperatures. In contrast, the wild-type protein diverges from its starting structure near its melting temperature. Meanwhile, the NOMELT variant preserves its structural integrity up to an additional 10K

These REMD results not only validate the mAF-min derived predictions but also highlight the potential of NOMELT as a tool for generating viable thermophilic variants with enhanced stability. The findings underscore the utility of NOMELT in facilitating the initial stages of protein engineering, particularly for less resource-intensive applications. Further exploration, especially involving enzymatic proteins, is anticipated as a promising direction for future research.

2.5.3 *NOMELT as a Zero-shot Estimator and Validating via Experimental Data*

In the absence of in vitro validation of engineered variants, we leveraged machine learning to predict general performance and aid in filtering directed evolution (DE) mutagenesis libraries.[128] Although this approach does not use the generative aspects of the NOMELT model, it provides practical insights by evaluating existing library variants. In this context, NOMELT was tested as a zero-shot predictor, where the wild-type sequence was input into the model's encoder, and a variant sequence (including insertions and deletions) was scored by the decoder using the output logits. We applied NOMELT to rank variants of LovD and LipA, comparing model scores with their experimentally determined melting temperatures.[129,130]

For LovD, NOMELT exhibited remarkable predictive accuracy without prior training on melting temperature prediction or exposure to the protein sequences, as confirmed by BLASTp searches (parameters detailed in Appendix B). However, while NOMELT effectively ranked high-temperature variants (>65 °C) for LipA, it overestimated the stability of wild-type variants. This discrepancy likely stems from the model's training to recognize and prioritize sequences similar to its learned thermophilic distribution, which may inadvertently conflate factors beyond mere thermal stability (Figure 2.5.6).

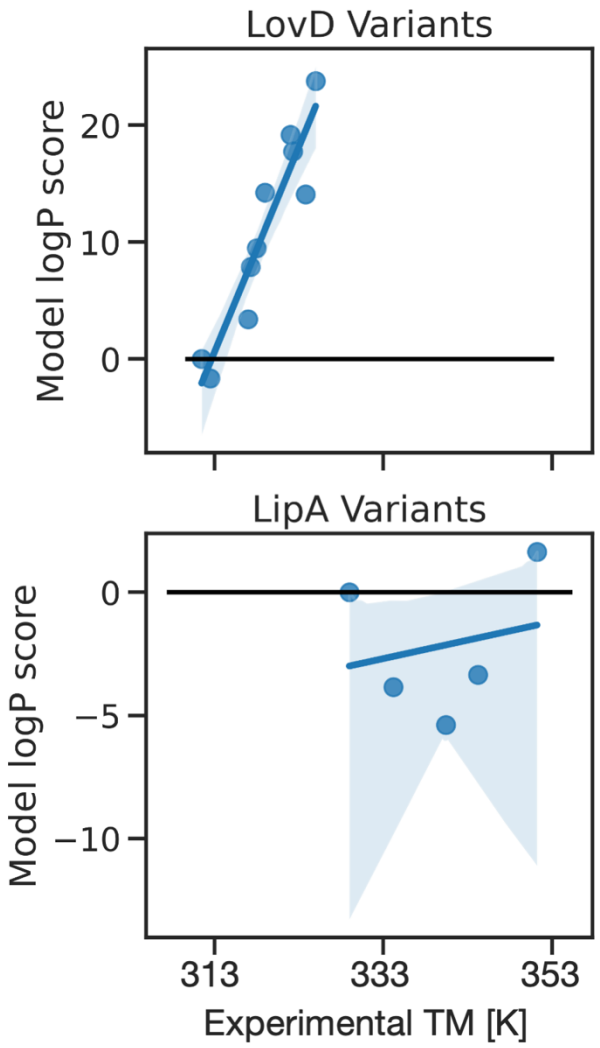

Figure 2.5.6. Experimental melting temperatures of LovD and LipA variants versus NOMELT log probability scores as per logit equation. The model effectively ranks LovD melting temperatures, evidenced by a Spearman's R of 0.94 (P-value 5.5e-5). Although less accurate, the model qualitatively ranks the three LipA variants with the highest temperatures

Further, NOMELT's capabilities were benchmarked using the Lipase A deep mutational scan (DMS) dataset from ProteinGym, focused specifically on thermal stability, measuring catalytic half activation temperature.[131,132] Here, NOMELT achieved a Spearman's correlation of 0.32 across 2,172 variants, a performance on par with ESM zero-shot models trained on vastly larger and more diverse datasets. Notably, NOMELT outperforms other single-sequence models, suggesting that its training on thermophilic translation provides a more tailored approach for estimating thermal stability than broader self-supervised learning methods. Despite the inherent

noise in the DMS assay and minimal changes in stability from single mutations, which often render variants statistically indistinguishable, NOMELT successfully distinguished 64% of discernible protein variants, a significant improvement over random predictions (Figure 2.5.7).

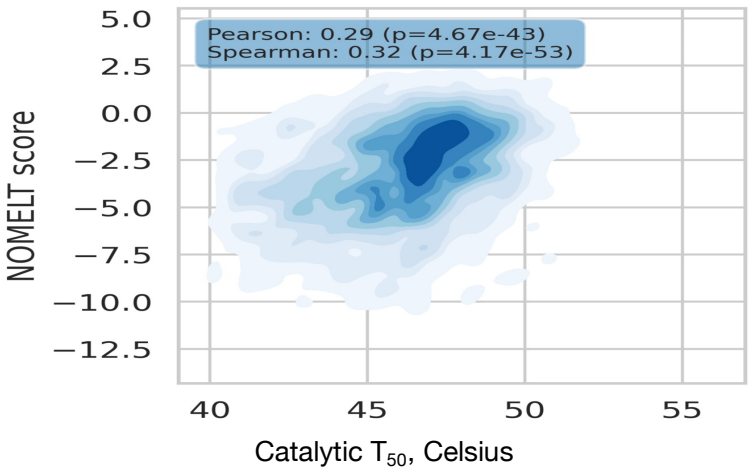

Figure 2.5.7. Parity of zero-shot prediction for single-point DMS variants affecting the catalytic T_{50} in *Bacillus subtilis* LipA.

While NOMELT may not precisely pinpoint the global optimum in a variant library, especially in DMS libraries where changes are subtle, its ability to reliably rank variants according to stability highlights its utility in practical protein engineering contexts. This demonstrates not only the model's robustness but also stresses the potential of machine learning to augment traditional biochemical methods by providing actionable insights into protein stability.

2.6 Summary and Conclusions

This chapter detailed NOMELT, a novel generative model designed to translate mesophilic protein sequences into their thermophilic counterparts, aimed at enhancing high temperature stability. A substantial dataset, learn2thermDB, was curated to train this model, encompassing a broad spectrum of protein space yet revealing certain gaps due to stringent filtering and reliance on available homologous pairs.

NOMELT demonstrated proficiency in generating sequences that not only mimic but occasionally surpass the stability of natural thermophilic variations, as evidenced by both

predictive estimations and molecular dynamics simulations. Particularly, the model's utility was highlighted through its application to the Engrailed Homeodomain (En-HD), where it introduced mutations that potentially increase thermal stability, pending experimental validation.

Further, the methodology could evolve from reliance on homologous pairing to using a more direct approach of conditioning the model on organism growth temperatures and metagenomic data. This would allow the model to generate temperature-adapted protein variants based on a broader spectrum of evolutionary information, potentially enhancing the model’s generative capabilities across a wider array of protein families.

The next step involves rethinking the training approach to accommodate these broader data inputs, possibly by developing a new model that integrates MSAs directly into the training process. Such a model would not only predict temperature-adapted protein sequences but also do so based on a comprehensive understanding of evolutionary and environmental adaptations, thus significantly advancing the field of protein design for high-temperature applications.

Chapter 3. STREAMLINING PROTEIN PAIR FUNCTIONAL SCREENING

3.1 INTRODUCTION

In Chapter 2, we demonstrated the significant role of careful data curation in analyzing complex phenomena such as protein thermostability. Building from this, Chapter 3 explores the application of similar computational techniques to screen protein pairs for analogous functionality. Over the last two decades, high-throughput sequencing technologies have dramatically accelerated protein discovery.[14] Yet, the functional characterization of these proteins remains an arduous and resource-intensive task, leaving the functions of many proteins unknown.[133] Consequently, there is a high demand for computational methods that can rapidly and accurately assign functions to both new and previously unannotated proteins. While several tools exist for screening and annotating these sequences, they often suffer from being decentralized, computationally intensive, and not user-friendly, which limits their practical use by researchers conducting high-throughput screenings. This chapter introduces PairProphet, a "first-pass" model that leverages existing proteome data to streamline the functional screening of protein pairs, significantly reducing the time required for experimental functional screening.

As discussed in the Chapter 1, protein function determination is inherently complex, necessitating an understanding of a multidimensional, nonlinear, rugged energy landscape that governs the probabilities of protein folding states and the energy barriers between them.[134] A biophysical model that directly links protein sequence to function remains elusive. Moreover, the concept of 'function' itself is multifaceted—unlike 3D structure, it cannot be captured by a single metric. Functions often involve interrelated properties such as catalytic activity and thermostability, and their mechanisms can be obscured by the vast size of sequence space.[13,15] As highlighted in Chapter 2, even focusing on a single property like enhanced thermostability requires extensive scrutiny across various protein families, significant experimental resources, and a variety metrics to identify top performers.[17,18] This approach, however, only scratches the surface of protein function, leaving aside the crucial aspect of protein dynamics which are vital to understanding function. Moreover, while homology modeling—through comparison of primary

amino acid sequences and 3D structures—helps approximate functional similarities, it falls short of providing definitive functional labelling.[135,136] The most reliable method remains observing proteins within their operational environment.[137,138] Nonetheless, the techniques discussed in Chapter 2, if employed properly, can facilitate the inference of functionality.

3.2 METHODS

3.2.1 *OMA Database*

To develop an effective 'first-pass' pairwise functional screener, it is important to utilize a dataset with reliable ground-truth labels. This dataset enables the training of a machine learning model capable of classifying whether two sequences are functionally related. Although the relationship between sequence and function is not straightforward, as discussed earlier, evolutionary insights can facilitate our search. The OMA (Orthologous MAtrix) database, a cornerstone in comparative genetics and phylogenetics, serves as our primary data source.[139,140] OMA defines orthology as a relationship between gene pairs from different species that diverged following a speciation event and thus share a common ancestor. This evolutionary relationship suggests that orthologs, which are often functionally conserved, can be instrumental in inferring species trees and, most importantly, predicting functional analogies.[141]

The OMA pipeline begins with an all-against-all Smith-Waterman alignment to identify homologs across all genomes, utilizing amino acid sequences. This method is preferred over DNA searches, which are less sensitive and offer a shorter evolutionary look-back time compared to protein or translated DNA alignments. After identifying significant homologous pairs, the algorithm computes similarity scores and evolutionary distances, incorporating estimation uncertainties and stringent criteria for evolutionary distance. Identified orthologs are then categorized into graphs representing one-to-one, one-to-many, many-to-one, and many-to-many relationships, which reflect various evolutionary groupings.

For our study, we sourced data from the "All.Jul2023" release of the OMA database, containing 2,851 full genomes and 22,092,112 proteins. We specifically targeted prokaryotic proteomes with protein sequences under 250 amino acids—a typical length for a functional protein domain. This

selection not only aligns with our functional focus but also reduces the computational demands of subsequent analyses. After this refinement, our dataset included approximately 2.596 x 10^8 proteins.

To train our model, we needed examples of both functional relationships and non-relationships. Using DuckDB's capabilities, we randomly shuffled the pairs of orthologs, maintaining indices to track their origins.[99] True orthologs were labeled with 'clean_#', where '#' denotes a numerical identifier from the original OMA database, and non-pairs were tagged as 'bad_#'. Furthermore, to manage protein isoforms effectively, we ensured each unique UniProt ID corresponded to a distinct protein sequence, determined by the longest amino acid sequence. This filtering process yielded roughly 1 million orthologs, balanced in a 70/30 ratio of 'good' to 'bad' examples. The entire pipeline is illustrated in Figure 3.2.1.

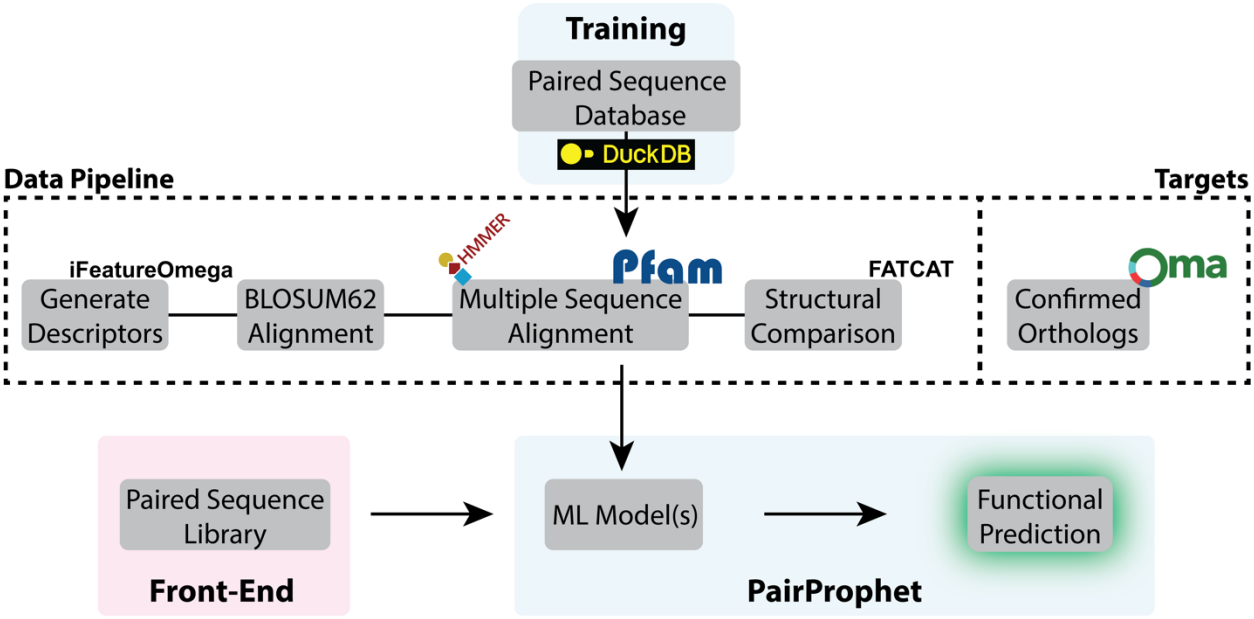

Figure 3.2.1. Pipeline for generating PairProphet

It is crucial to differentiate between the machine learning (ML) pipeline used for training the PairProphet model and the front-end user interface where researchers interact with the software. The core of the ML pipeline incorporates a suite of well-established homology algorithms, essential for predicting the functionality of protein pairs. Among these, global alignment is the only algorithm mandatory for model training, as it generates critical features for our model.

Additional algorithms, such as HMMER and FATCAT2, are optional; they are available through the user interface and can be activated based on the user's computational resources and the need for a more exhaustive analysis of potential functionality in protein pairs. Consequently, the PairProphet system offers various permutations of ML models, each incorporating different sets of homology algorithm features, thus allowing users to tailor the analysis to meet specific research requirements.

3.2.2 *Global Alignment*

The foundational component of our pipeline is the implementation of a BLOSSUM62 global alignment, detailed further in Appendix C, which includes alignment metrics and equations. This step is indispensable for both the training of our machine learning models and their deployment through the user interface. Global alignment, unlike other homology algorithms that are optional, is mandated because of its robust ability to detect evolutionary relationships across entire sequences, which is crucial for our modified OMA dataset where distinguishing between true functional pairs and challenging non-pairs is vital.[135] Global alignment methods, by aligning every residue in a sequence from start to finish, offer a comprehensive view of evolutionary and functional relationships. This comprehensive approach is particularly effective in scenarios where sequences may have diverged significantly but still retain faint evolutionary echoes across their full lengths, which might be missed by local alignment methods that focus on the most similar sequence segments.

Studies suggest that sequences with more than 40% identity generally share functional similarities, as evidenced by corresponding Enzyme Commission (E.C.) numbers. Nevertheless, there are notable exceptions where minimal amino acid differences—often in key active sites—can result in significant changes in enzyme activity. Such nuances illustrate the inadequacies of relying solely on significant local similarities, which may overlook critical functional divergences.[135] Moreover, global alignment is favored in our application due to its sensitivity to conserved residues that are often crucial for the structural and functional integrity of proteins. By considering the entirety of sequences, global alignment helps in identifying these conserved regions, which are vital for accurate function prediction. The algorithm's ability to identify both conserved domains and subtle variations across the full length of sequences provides a more

reliable basis for functional inference compared to local alignment methods, which might give undue weight to highly similar but functionally irrelevant segments. By employing global alignment, our model ensures a more accurate analysis of protein functionality, leveraging evolutionary data to predict functional relationships with higher confidence.

3.2.3 *HMMER and Pfam*

The second homology algorithm employed in our pipeline is pyhmmer, a Cython interface for the widely-used HMMER3 software, which leverages Hidden Markov Models (HMMs) rather than direct pairwise alignments.[102,103] Unlike simple pairwise methods, HMMER utilizes multiple sequence alignments (MSAs) of homologous protein families. These MSAs reveal evolutionarily conserved patterns at specific sites within the protein domains. Certain key residues are highly conserved, indicating essential functional or structural roles, while other positions may allow substitutions that preserve physicochemical properties like hydrophobicity, charge, or size. Additionally, some positions exhibit evolutionary neutrality with considerable variability, and various sites differently accommodate insertions and deletions. These patterns of conservation and variation are not only pivotal for constructing phylogenetic trees but also enhance the sensitivity of functional homology inference. Notably, the integration of MSAs into the AlphaFold2 algorithm was instrumental in achieving its groundbreaking accuracy in protein structure prediction, showcasing the power of this approach.[84]

For functional annotation, we utilize the Pfam 35.0 database, a well-known and well-used repository of protein families and domains.[101] Our method involves extracting the accession IDs from the protein pairs, which represent distinct protein domains. These IDs are then aggregated into a set, with the Jaccard score of this set serving as a functional score. This score is subsequently used as a feature in our ML model, similar to our approach for validating the learn2thermDB as discussed in section 2.2.2.

3.2.4 *FATCAT2*

Given the OMA's focus primarily on protein sequences, incorporating structural features is crucial for accurate functional determination. To integrate these structural elements, we accessed corresponding PDB structures via the UniProt IDs, utilizing the Structure Integration with Function, Taxonomy, and Sequence (SIFTS) project.[142,143] For pairs where both proteins have available 3D structures, we employed the FATCAT2 algorithm, a flexible alignment tool that accommodates the intrinsic flexibility of proteins.[104]

FATCAT2 aligns substructural elements, allowing variations in linker orientations and scoring alignments based on the RMSD of these substructures while penalizing significant discrepancies. This method is particularly effective in identifying functional alignments in cases where amino acid differences cause conformational changes but maintain overall structural similarity. By focusing on flexibility, FATCAT2 can uncover functional relationships that rigid alignment methods might miss, highlighting subtle but critical adaptations.

3.2.5 *iFeatureOmegaCLI*

Lastly, we employed iFeatureOmegaCLI, a comprehensive platform designed for generating, analyzing, and visualizing over 180 representations of biological sequences.[144] Our focus was on extracting a broad array of features from amino acid sequences, which differ fundamentally from the homology-based features previously described. These features include amino acid distributions, composition, and order, among others, which have proven to be invaluable for sequence-based models.

We successfully extracted 18 distinct features using this tool; for a detailed list and definitions of these features, refer to Appendix C.2. Each feature was selected based on its demonstrated ability to enhance the performance of sequence-models, providing a strong foundation for our machine learning algorithms.

3.2.6 *Training of Random Forest Classifier*

We selected the Random Forest classifier as our primary ML architecture due to its robustness and flexibility in handling complex and high-dimensional datasets. Random Forest is an ensemble learning technique that constructs multiple decision trees during training and outputs the mode of the classifications (for categorical output) or the mean prediction (for continuous output) across trees. This methodology not only improves predictive accuracy but also helps in controlling over-fitting, making it ideal for our diverse feature set.

For model training, we implemented an 85:15 random split between training and testing datasets to ensure a good evaluation of the model's performance. Hyperparameters were carefully optimized to enhance the model's efficacy. Specifically, the configuration of the Random Forest included setting the number of trees to 200, the maximum number of features considered for splitting a node to 50% of the total features (max_features = 0.5), and the minimum number of samples required to split a node to 17 (min_samples_split = 17). All other parameters were kept at their default settings as provided by scikit-learn. This training regimen adhered to best practices in machine learning, emphasizing rigorous validation and testing to prevent overfitting and ensure generalizability. The model's performance was subsequently assessed through standard metrics such as accuracy, precision, recall, and the F1 score, providing a comprehensive overview of its predictive capabilities.

All code used in this study, along with scripts that automatically download the necessary data, are publicly available on GitHub. It can be found in the following: https://github.com/learn2therm/PairProphet. This repository ensures that other researchers can replicate our findings and extend the analysis as needed, promoting an open and collaborative scientific environment.

3.3 Results and Discussion

The PairProphet pipeline facilitated the generation of eight distinct models, detailed in Table 3.9, utilizing all features discussed in the methods section. The rationale behind offering multiple models was to enable users to select the desired balance between analytical sensitivity and

computational cost. Our analysis revealed diminishing returns in model performance enhancements beyond the inclusion of HMMER, which significantly boosted accuracy—a result anticipated given the proven success of MSA techniques in structure and function prediction. However, these techniques often exhibit a bias toward structural rather than functional attributes.

Table 3.9. PairProphet (June Version) ML model summary

Feature	F1 score
BLAST	0.87
BLAST + iFeatureOmegaCLI	0.94
BLAST + HMMER	0.994
BLAST + FATCAT2	0.94
BLAST + HMMER + iFeatureOmegaCLI	0.995
BLAST + HMMER + FATCAT2	0.95
BLAST + FATCAT2 + iFeatureOmegaCLI	0.994
BLAST + FATCAT2 + iFeatureOmegaCLI + HMMER	0.995

Interestingly, analyses incorporating FATCAT2 and structural data did not significantly enhance the classifier's performance. This finding suggests potential avenues for future research, such as refining the selection process for chains from PDB files. Additionally, while iFeatureOmegaCLI contributed minimally, it demonstrated that sequence information was more effective than structural information in predicting pairwise protein functionality.

Looking ahead, we plan to refine PairProphet by implementing a more sophisticated approach to sampling and generating negative non-pairs from the OMA dataset, with careful consideration of protein isoforms. Further, we aim to validate the functional prediction capabilities of PairProphet using DsRed and its dysfunctional mutants as test cases.[145] Moreover, plans are underway to make PairProphet available online, enhancing accessibility for the broader scientific community.

3.4 SUMMARY AND CONCLUSIONS

This chapter introduced and detailed the PairProphet pipeline, an approach designed to streamline the functional screening of protein pairs. By leveraging a suite of computational techniques and databases, including the OMA database, HMMER, FATCAT2, and iFeatureOmegaCLI, we developed a robust machine learning model that predicts functional similarities between protein pairs with high accuracy.

The implementation of the Random Forest classifier allowed us to integrate and analyze a diverse set of features derived from both sequence and structural data. The model's performance was validated, demonstrating its capability to handle complex biological data effectively. Among the different features and algorithms tested, the inclusion of HMMER significantly enhanced the model's accuracy due to its use of multiple sequence alignments, which are crucial for identifying functional homology. However, our results also indicated that while sequence-based features provided substantial predictive power, structural features, especially those derived from FATCAT2, were less influential. This insight points to the complex nature of protein functionality and highlights the potential need for refined structural analysis techniques in future iterations of the pipeline.

Going forward, the PairProphet pipeline will undergo further refinements to improve its predictive accuracy and efficiency. This includes a more sophisticated approach to generating negative non-pairs and integrating a wider array of protein isoforms. Additionally, by expanding our validation efforts using DsRed and its mutants, we aim to further substantiate the model's applicability to real-world biological problems. Lastly, the planned online deployment of PairProphet will facilitate broader access and contribute to the scientific community, enabling researchers worldwide to benefit from this tool in their functional proteomic studies.

Chapter 4. CONCLUSIONS AND FUTURE WORK

This thesis has demonstrated the significant potential of machine learning and deep learning in advancing our understanding of complex biophysical properties. Chapter 2 explored the application of these methodologies to predict protein thermostability, showcasing the novel sequence-to-sequence translator, NOMELT. This model notably enhances the design of proteins to withstand high temperatures, emphasizing the utility of machine learning in generating actionable insights into protein behavior under various functional demands.

Chapter 3 introduced the PairProphet pipeline, an approach that streamlines the functional screening of protein pairs by leveraging computational techniques and databases such as OMA, HMMER, FATCAT2, and iFeatureOmegaCLI. The implementation of a Random Forest classifier, integrating diverse features from sequence and structural data, accentuated the model's high accuracy in predicting functional similarities between protein pairs. This chapter not only highlighted the capabilities of machine learning in handling complex biological data but also identified areas for further methodological enhancements, particularly in structural feature analysis.

However, these powerful predictive tools—while transforming our approach to understanding protein functions and stability—should be treated as hypothesis generators within iterative and empirical "design, build, test, and learn" cycles. This iterative process aims to refine these models, enabling a deeper integration of theoretical frameworks that undergird biological phenomena.

The limitations of traditional homology modeling and the insights gained from this research suggest a crucial trajectory for future work: the development of basic methods in bioinformatics that transcend traditional confines. Future computational models in protein science should integrate noisy, yet information-rich metagenomic data to create robust models capable of universal application.

In conclusion, this thesis illuminates the potential of merging advanced machine learning techniques with extensive biological data to enhance the modeling of biological systems. While the contributions of this work are modest, it foreshadows the crucial role that ecological information and pangenomic analyses play in advancing protein science. These approaches are pivotal because they have been shown to effectively predict protein function, thereby enriching our theoretical understanding and practical capabilities. By integrating these broader perspectives on ecological and microenvironmental contexts, future research can more effectively bridge the theoretical models with the complex realities of biological systems. This integration is essential for significant advancements in our ability to predict and manipulate protein behavior in diverse environmental conditions.

BIBLIOGRAPHY

(1) Alberts, B. *Molecular Biology of the Cell*, 6th ed.; W.W. Norton & Company: New York, 2017. https://doi.org/10.1201/9781315735368.

(2) Mirsky, A. E.; Pauling, L. On the Structure of Native, Denatured, and Coagulated Proteins. *Proc. Natl. Acad. Sci. U. S. A.* **1936**, *22* (7), 439–447.

(3) Korendovych, I. V.; DeGrado, W. F. *De Novo* Protein Design, a Retrospective. *Q. Rev. Biophys.* **2020**, *53*, e3. https://doi.org/10.1017/S0033583519000131.

(4) Wuyun, Q.; Chen, Y.; Shen, Y.; Cao, Y.; Hu, G.; Cui, W.; Gao, J.; Zheng, W. Recent Progress of Protein Tertiary Structure Prediction. *Molecules* **2024**, *29* (4), 832. https://doi.org/10.3390/molecules29040832.

(5) Chen, S.-J.; Hassan, M.; Jernigan, R. L.; Jia, K.; Kihara, D.; Kloczkowski, A.; Kotelnikov, S.; Kozakov, D.; Liang, J.; Liwo, A.; Matysiak, S.; Meller, J.; Micheletti, C.; Mitchell, J. C.; Mondal, S.; Nussinov, R.; Okazaki, K.; Padhorny, D.; Skolnick, J.; Sosnick, T. R.; Stan, G.; Vakser, I.; Zou, X.; Rose, G. D. Protein Folds vs. Protein Folding: Differing Questions, Different Challenges. *Proc. Natl. Acad. Sci.* **2023**, *120* (1), e2214423119. https://doi.org/10.1073/pnas.2214423119.

(6) Hait, S.; Mallik, S.; Basu, S.; Kundu, S. Finding the Generalized Molecular Principles of Protein Thermal Stability. *Proteins Struct. Funct. Bioinforma.* **2020**, *88* (6), 788–808. https://doi.org/10.1002/prot.25866.

(7) Tokuriki, N.; Tawfik, D. S. Stability Effects of Mutations and Protein Evolvability. *Curr. Opin. Struct. Biol.* **2009**, *19* (5), 596–604. https://doi.org/10.1016/j.sbi.2009.08.003.

(8) Fersht, A. *Structure and Mechanism in Protein Science: A Guide to Enzyme Catalysis and Protein Folding*, 1st edition.; W. H. Freeman: New York, 1998.

(9) Trivedi, R.; Nagarajaram, H. A. Intrinsically Disordered Proteins: An Overview. *Int. J. Mol. Sci.* **2022**, *23* (22), 14050. https://doi.org/10.3390/ijms232214050.

(10) Liou, T. G. The Clinical Biology of Cystic Fibrosis Transmembrane Regulator Protein. *Chest* **2019**, *155* (3), 605–616. https://doi.org/10.1016/j.chest.2018.10.006.

(11) Ball, P. *How Life Works: A User's Guide to the New Biology*, 1st ed.; University of Chicago Press, 2023.

(12) Vila, J. A. Metamorphic Proteins in Light of Anfinsen's Dogma. *J. Phys. Chem. Lett.* **2020**, *11* (13), 4998–4999. https://doi.org/10.1021/acs.jpclett.0c01414.

(13) Notin, P.; Rollins, N.; Gal, Y.; Sander, C.; Marks, D. Machine Learning for Functional Protein Design. *Nat. Biotechnol.* **2024**, *42* (2), 216–228. https://doi.org/10.1038/s41587-024-02127-0.

(14) Jeffery, C. J. Current Successes and Remaining Challenges in Protein Function Prediction. *Front. Bioinforma.* **2023**, *3*, 1222182. https://doi.org/10.3389/fbinf.2023.1222182.

(15) Porter, L. L. Fluid Protein Fold Space and Its Implications. *BioEssays* **2023**, *45* (9), 2300057. https://doi.org/10.1002/bies.202300057.

(16) Chica, R. A.; Ferruz, N. What Does It Take for an 'AlphaFold Moment' in Functional Protein Engineering and Design? *Nat. Biotechnol.* **2024**, *42* (2), 173–174. https://doi.org/10.1038/s41587-023-02120-z.

(17) Lane, T. J. Protein Structure Prediction Has Reached the Single-Structure Frontier. *Nat. Methods* **2023**, *20* (2), 170–173. https://doi.org/10.1038/s41592-022-01760-4.

(18) Doerr, A. Protein Design: The Experts Speak. *Nat. Biotechnol.* **2024**, *42* (2), 175–178. https://doi.org/10.1038/s41587-023-02111-0.
(19) Polizzi, K. M.; Bommarius, A. S.; Broering, J. M.; Chaparro-Riggers, J. F. Stability of Biocatalysts. *Curr. Opin. Chem. Biol.* **2007**, *11* (2), 220–225. https://doi.org/10.1016/j.cbpa.2007.01.685.
(20) Yang, J.; Li, F.-Z.; Arnold, F. H. Opportunities and Challenges for Machine Learning-Assisted Enzyme Engineering. *ACS Cent. Sci.* **2024**, *10* (2), 226–241. https://doi.org/10.1021/acscentsci.3c01275.
(21) Chiu, M. L.; Goulet, D. R.; Teplyakov, A.; Gilliland, G. L. Antibody Structure and Function: The Basis for Engineering Therapeutics. *Antibodies* **2019**, *8* (4), 55. https://doi.org/10.3390/antib8040055.
(22) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-Scale Prediction of Atomic-Level Protein Structure with a Language Model. *Science* **2023**, *379* (6637), 1123–1130. https://doi.org/10.1126/science.ade2574.
(23) Asnicar, F.; Thomas, A. M.; Passerini, A.; Waldron, L.; Segata, N. Machine Learning for Microbiologists. *Nat. Rev. Microbiol.* **2023**, 1–15. https://doi.org/10.1038/s41579-023-00984-1.
(24) AlQuraishi, M.; Sorger, P. K. Differentiable Biology: Using Deep Learning for Biophysics-Based and Data-Driven Modeling of Molecular Mechanisms. *Nat. Methods* **2021**, *18* (10), 1169–1180. https://doi.org/10.1038/s41592-021-01283-4.
(25) Akdel, M.; Pires, D. E. V.; Pardo, E. P.; Jänes, J.; Zalevsky, A. O.; Mészáros, B.; Bryant, P.; Good, L. L.; Laskowski, R. A.; Pozzati, G.; Shenoy, A.; Zhu, W.; Kundrotas, P.; Serra, V. R.; Rodrigues, C. H. M.; Dunham, A. S.; Burke, D.; Borkakoti, N.; Velankar, S.; Frost, A.; Basquin, J.; Lindorff-Larsen, K.; Bateman, A.; Kajava, A. V.; Valencia, A.; Ovchinnikov, S.; Durairaj, J.; Ascher, D. B.; Thornton, J. M.; Davey, N. E.; Stein, A.; Elofsson, A.; Croll, T. I.; Beltrao, P. A Structural Biology Community Assessment of AlphaFold2 Applications. *Nat. Struct. Mol. Biol.* **2022**, *29* (11), 1056–1067. https://doi.org/10.1038/s41594-022-00849-w.
(26) Jeliazkov, J. R.; Alamo, D. del; Karpiak, J. D. ESMFold Hallucinates Native-Like Protein Sequences. bioRxiv May 24, 2023, p 2023.05.23.541774. https://doi.org/10.1101/2023.05.23.541774.
(27) Chu, A. E.; Lu, T.; Huang, P.-S. Sparks of Function by de Novo Protein Design. *Nat. Biotechnol.* **2024**, *42* (2), 203–215. https://doi.org/10.1038/s41587-024-02133-2.
(28) Wayment-Steele, H. K.; Ojoawo, A.; Otten, R.; Apitz, J. M.; Pitsawong, W.; Hömberger, M.; Ovchinnikov, S.; Colwell, L.; Kern, D. Predicting Multiple Conformations via Sequence Clustering and AlphaFold2. *Nature* **2024**, *625* (7996), 832–839. https://doi.org/10.1038/s41586-023-06832-9.
(29) Komp, E.; Alanzi, H. H.; Francis, R.; Vuong, C.; Roberts, L.; Mossallanejad, A.; Beck, D. A. C. Homologous Pairs of Low and High Temperature Originating Proteins Spanning the Known Prokaryotic Universe. *Sci. Data* **2023**, *10* (1), 682. https://doi.org/10.1038/s41597-023-02553-w.
(30) Komp, E.; Phillips, C.; Alanzi, H. N.; Zorman, M.; Beck, D. A. C. A Learnable Transition from Low Temperature to High Temperature Proteins with Neural Machine Translation. bioRxiv February 8, 2024, p 2024.02.06.579188. https://doi.org/10.1101/2024.02.06.579188.

(31) Olsen, H. S.; Falholt, P. The Role of Enzymes in Modern Detergency. *J. Surfactants Deterg.* **1998**, *1* (4), 555–567. https://doi.org/10.1007/s11743-998-0058-7.
(32) Bruggink, A.; Roy, P. D. Industrial Synthesis of Semisynthetic Antibiotics. In *Synthesis of β-Lactam Antibiotics: Chemistry, Biocatalysis & Process Integration*; Bruggink, A., Ed.; Springer Netherlands: Dordrecht, 2001; pp 12–54. https://doi.org/10.1007/978-94-010-0850-1_1.
(33) Tatsis, E. C.; Carqueijeiro, I.; Dugé de Bernonville, T.; Franke, J.; Dang, T.-T. T.; Oudin, A.; Lanoue, A.; Lafontaine, F.; Stavrinides, A. K.; Clastre, M.; Courdavault, V.; O'Connor, S. E. A Three Enzyme System to Generate the Strychnos Alkaloid Scaffold from a Central Biosynthetic Intermediate. *Nat. Commun.* **2017**, *8* (1), 316. https://doi.org/10.1038/s41467-017-00154-x.
(34) Wackett, L. P.; Hershberger, C. D. *Biocatalysis and Biodegradation: Microbial Transformation of Organic Compounds. Ch. 10: Microbial Biotechnology: Chemical Production and Bioremediation*; ASM Press: Washington, D.C, 2001.
(35) Wood, T. K. Molecular Approaches in Bioremediation. *Curr. Opin. Biotechnol.* **2008**, *19* (6), 572–578. https://doi.org/10.1016/j.copbio.2008.10.003.
(36) Peixoto, R. S.; Vermelho, A. B.; Rosado, A. S. Petroleum-Degrading Enzymes: Bioremediation and New Prospects. *Enzyme Res.* **2011**, *2011*, 1–7. https://doi.org/10.4061/2011/475193.
(37) Sheldon, R. A.; Woodley, J. M. Role of Biocatalysis in Sustainable Chemistry. *Chem. Rev.* **2018**, *118* (2), 801–838. https://doi.org/10.1021/acs.chemrev.7b00203.
(38) Raveendran, S.; Parameswaran, B.; Ummalyma, S. B.; Abraham, A.; Mathew, A. K.; Madhavan, A.; Rebello, S.; Pandey, A. Applications of Microbial Enzymes in Food Industry. *Food Technol. Biotechnol.* **2018**, *56* (1), 16–30. https://doi.org/10.17113/ftb.56.01.18.5491.
(39) Singhania, R. R.; Ruiz, H. A.; Awasthi, M. K.; Dong, C.-D.; Chen, C.-W.; Patel, A. K. Challenges in Cellulase Bioprocess for Biofuel Applications. *Renew. Sustain. Energy Rev.* **2021**, *151*, 111622. https://doi.org/10.1016/j.rser.2021.111622.
(40) Vojcic, L.; Pitzler, C.; Körfer, G.; Jakob, F.; Ronny Martinez, null; Maurer, K.-H.; Schwaneberg, U. Advances in Protease Engineering for Laundry Detergents. *New Biotechnol.* **2015**, *32* (6), 629–634. https://doi.org/10.1016/j.nbt.2014.12.010.
(41) Leuenberger, P.; Ganscha, S.; Kahraman, A.; Cappelletti, V.; Boersema, P. J.; von Mering, C.; Claassen, M.; Picotti, P. Cell-Wide Analysis of Protein Thermal Unfolding Reveals Determinants of Thermostability. *Science* **2017**, *355* (6327), eaai7825. https://doi.org/10.1126/science.aai7825.
(42) Lewontin, R. *The Triple Helix: Gene, Organism, and Environment*; Harvard University Press: Cambridge, MA, 2002.
(43) Levins, R. *The Dialectical Biologist*; Harvard University Press: Cambridge, Mass, 1985.
(44) Rothschild, L. J.; Mancinelli, R. L. Life in Extreme Environments. *Nature* **2001**, *409* (6823), 1092–1101. https://doi.org/10.1038/35059215.
(45) Cowan, D.; Ramond, J.-B.; Makhalanyane, T.; De Maayer, P. Metagenomics of Extreme Environments. *Curr. Opin. Microbiol.* **2015**, *25*, 97–102. https://doi.org/10.1016/j.mib.2015.05.005.
(46) Mehta, R.; Singhal, P.; Singh, H.; Damle, D.; Sharma, A. K. Insight into Thermophiles and Their Wide-Spectrum Applications. *3 Biotech* **2016**, *6* (1), 81. https://doi.org/10.1007/s13205-016-0368-z.

(47) Pucci, F.; Rooman, M. Physical and Molecular Bases of Protein Thermal Stability and Cold Adaptation. *Curr. Opin. Struct. Biol.* **2017**, *42*, 117–128. https://doi.org/10.1016/j.sbi.2016.12.007.

(48) Angelin, J.; Kavitha, M. Chapter 5 - Molecular Mechanisms behind the Cold and Hot Adaptation in Extremozymes. In *Extremozymes and Their Industrial Applications*; Arora, N. K., Agnihotri, S., Mishra, J., Eds.; Academic Press, 2022; pp 141–176. https://doi.org/10.1016/B978-0-323-90274-8.00013-7.

(49) Zeldovich, K. B.; Berezovsky, I. N.; Shakhnovich, E. I. Protein and DNA Sequence Determinants of Thermophilic Adaptation. *PLoS Comput. Biol.* **2007**, *3* (1), e5. https://doi.org/10.1371/journal.pcbi.0030005.

(50) Sawle, L.; Ghosh, K. How Do Thermophilic Proteins and Proteomes Withstand High Temperature? *Biophys. J.* **2011**, *101* (1), 217–227. https://doi.org/10.1016/j.bpj.2011.05.059.

(51) Venev, S. V.; Zeldovich, K. B. Thermophilic Adaptation in Prokaryotes Is Constrained by Metabolic Costs of Proteostasis. *Mol. Biol. Evol.* **2018**, *35* (1), 211–224. https://doi.org/10.1093/molbev/msx282.

(52) Li, G.; Hu, Y.; Jan Zrimec; Luo, H.; Wang, H.; Zelezniak, A.; Ji, B.; Nielsen, J. Bayesian Genome Scale Modelling Identifies Thermal Determinants of Yeast Metabolism. *Nat. Commun.* **2021**, *12* (1), 190. https://doi.org/10.1038/s41467-020-20338-2.

(53) Berezovsky, I. N.; Zeldovich, K. B.; Shakhnovich, E. I. Positive and Negative Design in Stability and Thermal Adaptation of Natural Proteins. *PLOS Comput. Biol.* **2007**, *3* (3), e52. https://doi.org/10.1371/journal.pcbi.0030052.

(54) Karshikoff, A.; Nilsson, L.; Ladenstein, R. Rigidity versus Flexibility: The Dilemma of Understanding Protein Thermal Stability. *FEBS J.* **2015**, *282* (20), 3899–3917. https://doi.org/10.1111/febs.13343.

(55) Quezada, A. G.; Díaz-Salazar, A. J.; Cabrera, N.; Pérez-Montfort, R.; Piñeiro, Á.; Costas, M. Interplay between Protein Thermal Flexibility and Kinetic Stability. *Structure* **2017**, *25* (1), 167–179. https://doi.org/10.1016/j.str.2016.11.018.

(56) Rutherford, K.; Bennion, B. J.; Parson, W. W.; Daggett, V. The 108M Polymorph of Human Catechol O-Methyltransferase Is Prone to Deformation at Physiological Temperatures. *Biochemistry* **2006**, *45* (7), 2178–2188. https://doi.org/10.1021/bi051988i.

(57) Merkley, E. D.; Parson, W. W.; Daggett, V. Temperature Dependence of the Flexibility of Thermophilic and Mesophilic Flavoenzymes of the Nitroreductase Fold. *Protein Eng. Des. Sel. PEDS* **2010**, *23* (5), 327–336. https://doi.org/10.1093/protein/gzp090.

(58) Tang, H.; Cao, R.-Z.; Wang, W.; Liu, T.-S.; Wang, L.-M.; He, C.-M. A Two-Step Discriminated Method to Identify Thermophilic Proteins. *Int. J. Biomath.* **2017**, *10* (04), 1750050. https://doi.org/10.1142/S1793524517500504.

(59) Feng, C.; Ma, Z.; Yang, D.; Li, X.; Zhang, J.; Li, Y. A Method for Prediction of Thermophilic Protein Based on Reduced Amino Acids and Mixed Features. *Front. Bioeng. Biotechnol.* **2020**, *8*, 285. https://doi.org/10.3389/fbioe.2020.00285.

(60) Charoenkwan, P.; Chotpatiwetchkul, W.; Lee, V. S.; Nantasenamat, C.; Shoombuatong, W. A Novel Sequence-Based Predictor for Identifying and Characterizing Thermophilic Proteins Using Estimated Propensity Scores of Dipeptides. *Sci. Rep.* **2021**, *11* (1), 23782. https://doi.org/10.1038/s41598-021-03293-w.

(61) Li, G.; Buric, F.; Zrimec, J.; Viknander, S.; Nielsen, J.; Zelezniak, A.; Engqvist, M. K. M. Learning Deep Representations of Enzyme Thermal Adaptation. *Protein Sci.* **2022**, *31* (12), e4480. https://doi.org/10.1002/pro.4480.

(62) Packer, M. S.; Liu, D. R. Methods for the Directed Evolution of Proteins. *Nat. Rev. Genet.* **2015**, *16* (7), 379–394. https://doi.org/10.1038/nrg3927.

(63) Pikkemaat, M. G.; Linssen, A. B. M.; Berendsen, H. J. C.; Janssen, D. B. Molecular Dynamics Simulations as a Tool for Improving Protein Stability. *Protein Eng. Des. Sel.* **2002**, *15* (3), 185–192. https://doi.org/10.1093/protein/15.3.185.

(64) Greener, J. G.; Kandathil, S. M.; Moffat, L.; Jones, D. T. A Guide to Machine Learning for Biologists. *Nat. Rev. Mol. Cell Biol.* **2022**, *23* (1), 40–55. https://doi.org/10.1038/s41580-021-00407-0.

(65) Zheng, Z.; Deng, Y.; Xue, D.; Zhou, Y.; YE, F.; Gu, Q. Structure-Informed Language Models Are Protein Designers. arXiv February 9, 2023. https://doi.org/10.48550/arXiv.2302.01649.

(66) Dauparas, J.; Anishchenko, I.; Bennett, N.; Bai, H.; Ragotte, R. J.; Milles, L. F.; Wicky, B. I. M.; Courbet, A.; de Haas, R. J.; Bethel, N.; Leung, P. J. Y.; Huddy, T. F.; Pellock, S.; Tischer, D.; Chan, F.; Koepnick, B.; Nguyen, H.; Kang, A.; Sankaran, B.; Bera, A. K.; King, N. P.; Baker, D. Robust Deep Learning–Based Protein Sequence Design Using ProteinMPNN. *Science* **2022**, *378* (6615), 49–56. https://doi.org/10.1126/science.add2187.

(67) Singer, G. A. C.; Hickey, D. A. Thermophilic Prokaryotes Have Characteristic Patterns of Codon Usage, Amino Acid Composition and Nucleotide Content. *Gene* **2003**, *317*, 39–47. https://doi.org/10.1016/S0378-1119(03)00660-7.

(68) Gromiha, M. M.; Nagarajan, R.; Selvaraj, S. Protein Structural Bioinformatics: An Overview. In *Encyclopedia of Bioinformatics and Computational Biology*; Elsevier, 2019; pp 445–459. https://doi.org/10.1016/B978-0-12-809633-8.20278-1.

(69) Szilágyi, A.; Závodszky, P. Structural Differences between Mesophilic, Moderately Thermophilic and Extremely Thermophilic Protein Subunits: Results of a Comprehensive Survey. *Structure* **2000**, *8* (5), 493–504. https://doi.org/10.1016/S0969-2126(00)00133-7.

(70) England, J. L.; Shakhnovich, B. E.; Shakhnovich, E. I. Natural Selection of More Designable Folds: A Mechanism for Thermophilic Adaptation. *Proc. Natl. Acad. Sci.* **2003**, *100* (15), 8727–8731. https://doi.org/10.1073/pnas.1530713100.

(71) Sawle, L.; Ghosh, K. How Do Thermophilic Proteins and Proteomes Withstand High Temperature? *Biophys. J.* **2011**, *101* (1), 217–227. https://doi.org/10.1016/j.bpj.2011.05.059.

(72) Berezovsky, I. N.; Shakhnovich, E. I. Physics and Evolution of Thermophilic Adaptation. *Proc. Natl. Acad. Sci.* **2005**, *102* (36), 12742–12747. https://doi.org/10.1073/pnas.0503890102.

(73) Stourac, J.; Dubrava, J.; Musil, M.; Horackova, J.; Damborsky, J.; Mazurenko, S.; Bednar, D. FireProtDB: Database of Manually Curated Protein Stability Data. *Nucleic Acids Res.* **2021**, *49* (D1), D319–D324. https://doi.org/10.1093/nar/gkaa981.

(74) Gromiha, M. M.; Oobatake, M.; Sarai, A. Important Amino Acid Properties for Enhanced Thermostability from Mesophilic to Thermophilic Proteins. *Biophys. Chem.* **1999**, *82* (1), 51–67. https://doi.org/10.1016/S0301-4622(99)00103-9.

(75) Miotto, M.; Olimpieri, P. P.; Di Rienzo, L.; Ambrosetti, F.; Corsi, P.; Lepore, R.; Tartaglia, G. G.; Milanetti, E. Insights on Protein Thermal Stability: A Graph

Representation of Molecular Interactions. *Bioinformatics* **2018**, *35* (15), 2569–2577. https://doi.org/10.1093/bioinformatics/bty1011.

(76) Dehouck, Y.; Folch, B.; Rooman, M. Revisiting the Correlation between Proteins' Thermoresistance and Organisms' Thermophilicity. *Protein Eng. Des. Sel.* **2008**, *21* (4), 275–278. https://doi.org/10.1093/protein/gzn001.

(77) Ahmed, Z.; Zulfiqar, H.; Tang, L.; Lin, H. A Statistical Analysis of the Sequence and Structure of Thermophilic and Non-Thermophilic Proteins. *Int. J. Mol. Sci.* **2022**, *23* (17). https://doi.org/10.3390/ijms231710116.

(78) Jarzab, A.; Kurzawa, N.; Hopf, T.; Moerch, M.; Zecha, J.; Leijten, N.; Bian, Y.; Musiol, E.; Maschberger, M.; Stoehr, G.; Becher, I.; Daly, C.; Samaras, P.; Mergner, J.; Spanier, B.; Angelov, A.; Werner, T.; Bantscheff, M.; Wilhelm, M.; Klingenspor, M.; Lemeer, S.; Liebl, W.; Hahne, H.; Savitski, M. M.; Kuster, B. Meltome Atlas—Thermal Proteome Stability across the Tree of Life. *Nat. Methods* **2020**, *17* (5), 495–503. https://doi.org/10.1038/s41592-020-0801-4.

(79) Pucci, F.; Rooman, M. Improved Insights into Protein Thermal Stability: From the Molecular to the Structurome Scale. *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.* **2016**, *374* (2080), 20160141. https://doi.org/10.1098/rsta.2016.0141.

(80) Nikam, R.; Kulandaisamy, A.; Harini, K.; Sharma, D.; Gromiha, M. M. ProThermDB: Thermodynamic Database for Proteins and Mutants Revisited after 15 Years. *Nucleic Acids Res.* **2021**, *49* (D1), D420–D424. https://doi.org/10.1093/nar/gkaa1035.

(81) Engqvist, M. K. M. Correlating Enzyme Annotations with a Large Set of Microbial Growth Temperatures Reveals Metabolic Adaptations to Growth at Diverse Temperatures. *BMC Microbiol.* **2018**, *18* (1), 177. https://doi.org/10.1186/s12866-018-1320-7.

(82) Pudžiuvelytė, I.; Olechnovič, K.; Godliauskaite, E.; Sermokas, K.; Urbaitis, T.; Gasiunas, G.; Kazlauskas, D. TemStaPro: Protein Thermostability Prediction Using Sequence Representations from Protein Language Models. bioRxiv April 26, 2023, p 2023.03.27.534365. https://doi.org/10.1101/2023.03.27.534365.

(83) Jung, F.; Frey, K.; Zimmer, D.; Mühlhaus, T. DeepSTABp: A Deep Learning Approach for the Prediction of Thermal Protein Stability. *Int. J. Mol. Sci.* **2023**, *24* (8), 7444. https://doi.org/10.3390/ijms24087444.

(84) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2.

(85) Robert Verkuil; Ori Kabeli; Yilun Du; Basile I. M. Wicky; Lukas F. Milles; Justas Dauparas; David Baker; Sergey Ovchinnikov; Tom Sercu; Alexander Rives. Language Models Generalize beyond Natural Proteins. *bioRxiv* **2022**, 2022.12.21.521521. https://doi.org/10.1101/2022.12.21.521521.

(86) Ananthan Nambiar; Simon Liu; Mark Hopkins; Maeve Heflin; Sergei Maslov; Anna Ritz. Transforming the Language of Life: Transformer Neural Networks for Protein Prediction Tasks. *bioRxiv* **2020**, 2020.06.15.153643. https://doi.org/10.1101/2020.06.15.153643.

(87) NCBI Taxonomy: A Comprehensive Update on Curation, Resources and Tools. https://academic.oup.com/database/article/doi/10.1093/database/baaa062/5881509?login=false.

(88) O'Leary, N. A.; Wright, M. W.; Brister, J. R.; Ciufo, S.; Haddad, D.; McVeigh, R.; Rajput, B.; Robbertse, B.; Smith-White, B.; Ako-Adjei, D.; Astashyn, A.; Badretdin, A.; Bao, Y.; Blinkova, O.; Brover, V.; Chetvernin, V.; Choi, J.; Cox, E.; Ermolaeva, O.; Farrell, C. M.; Goldfarb, T.; Gupta, T.; Haft, D.; Hatcher, E.; Hlavina, W.; Joardar, V. S.; Kodali, V. K.; Li, W.; Maglott, D.; Masterson, P.; McGarvey, K. M.; Murphy, M. R.; O'Neill, K.; Pujar, S.; Rangwala, S. H.; Rausch, D.; Riddick, L. D.; Schoch, C.; Shkeda, A.; Storz, S. S.; Sun, H.; Thibaud-Nissen, F.; Tolstoy, I.; Tully, R. E.; Vatsan, A. R.; Wallin, C.; Webb, D.; Wu, W.; Landrum, M. J.; Kimchi, A.; Tatusova, T.; DiCuccio, M.; Kitts, P.; Murphy, T. D.; Pruitt, K. D. Reference Sequence (RefSeq) Database at NCBI: Current Status, Taxonomic Expansion, and Functional Annotation. *Nucleic Acids Res.* **2016**, *44* (Database issue), D733–D745. https://doi.org/10.1093/nar/gkv1189.

(89) The UniProt Consortium. UniProt: The Universal Protein Knowledgebase in 2023. *Nucleic Acids Res.* **2023**, *51* (D1), D523–D531. https://doi.org/10.1093/nar/gkac1052.

(90) Entrez Direct: E-Utilities on the Unix Command Line. In *Entrez Programming Utilities Help*; National Center for Biotechnology Information, 2023.

(91) *Bacteria (ID 33175) - BioProject - NCBI*. NCBI. Bacterial 16S Ribosomal RNA RefSeq Targeted Loci Project. 2008/12. In: BioProject [Internet]. http://www.ncbi.nlm.nih.gov/bioproject/PRJNA33175 (accessed 2023-05-14).

(92) *Archaea (ID 33317) - BioProject - NCBI*. NCBI. Archaeal 16S Ribosomal RNA RefSeq Targeted Loci Project. 2008/12. In: BioProject [Internet]. https://www.ncbi.nlm.nih.gov/bioproject/33317 (accessed 2023-05-14).

(93) Camacho, C.; Coulouris, G.; Avagyan, V.; Ma, N.; Papadopoulos, J.; Bealer, K.; Madden, T. L. BLAST+: Architecture and Applications. *BMC Bioinformatics* **2009**, *10* (1), 421. https://doi.org/10.1186/1471-2105-10-421.

(94) Yang, B.; Wang, Y.; Qian, P.-Y. Sensitivity and Correlation of Hypervariable Regions in 16S rRNA Genes in Phylogenetic Analysis. *BMC Bioinformatics* **2016**, *17* (1), 135. https://doi.org/10.1186/s12859-016-0992-y.

(95) Schloss, P. D. The Effects of Alignment Quality, Distance Calculation Method, Sequence Filtering, and Region on the Analysis of 16S rRNA Gene-Based Studies. *PLOS Comput. Biol.* **2010**, *6* (7), e1000844. https://doi.org/10.1371/journal.pcbi.1000844.

(96) Kim, M.; Oh, H.-S.; Park, S.-C.; Chun, J. Towards a Taxonomic Coherence between Average Nucleotide Identity and 16S rRNA Gene Sequence Similarity for Species Demarcation of Prokaryotes. *International Journal of Systematic and Evolutionary Microbiology*, 2014, *64*, 346–351. https://doi.org/10.1099/ijs.0.059774-0.

(97) Buchfink, B.; Xie, C.; Huson, D. H. Fast and Sensitive Protein Alignment Using DIAMOND. *Nat. Methods* **2015**, *12* (1), 59–60. https://doi.org/10.1038/nmeth.3176.

(98) *Data Version Control · DVC*. Data Version Control · DVC. https://dvc.org/ (accessed 2023-05-14).

(99) Raasveldt, M.; Mühleisen, H. DuckDB: An Embeddable Analytical Database. In *Proceedings of the 2019 International Conference on Management of Data*; SIGMOD '19; Association for Computing Machinery: New York, NY, USA, 2019; pp 1981–1984. https://doi.org/10.1145/3299869.3320212.

(100) CodeCarbon.Io. https://codecarbon.io/.

(101) Mistry, J.; Chuguransky, S.; Williams, L.; Qureshi, M.; Salazar, G. A.; Sonnhammer, E. L. L.; Tosatto, S. C. E.; Paladin, L.; Raj, S.; Richardson, L. J.; Finn, R. D.; Bateman, A. Pfam: The Protein Families Database in 2021. *Nucleic Acids Res.* **2021**, *49* (D1), D412–D419. https://doi.org/10.1093/nar/gkaa913.

(102) Finn, R. D.; Clements, J.; Eddy, S. R. HMMER Web Server: Interactive Sequence Similarity Searching. *Nucleic Acids Res.* **2011**, *39* (suppl_2), W29–W37. https://doi.org/10.1093/nar/gkr367.

(103) Larralde, M.; Zeller, G. PyHMMER: A Python Library Binding to HMMER for Efficient Sequence Analysis. *Bioinformatics* **2023**, *39* (5), btad214. https://doi.org/10.1093/bioinformatics/btad214.

(104) Li, Z.; Jaroszewski, L.; Iyer, M.; Sedova, M.; Godzik, A. FATCAT 2.0: Towards a Better Understanding of the Structural Diversity of Proteins. *Nucleic Acids Res.* **2020**, *48* (W1), W60–W64. https://doi.org/10.1093/nar/gkaa443.

(105) Burley, S. K.; Berman, H. M.; Kleywegt, G. J.; Markley, J. L.; Nakamura, H.; Velankar, S. Protein Data Bank (PDB): The Single Global Macromolecular Structure Archive. In *Protein Crystallography: Methods and Protocols*; Springer, 2017.

(106) Varadi, M.; Anyango, S.; Deshpande, M.; Nair, S.; Natassia, C.; Yordanova, G.; Yuan, D.; Stroe, O.; Wood, G.; Laydon, A.; Žídek, A.; Green, T.; Tunyasuvunakool, K.; Petersen, S.; Jumper, J.; Clancy, E.; Green, R.; Vora, A.; Lutfi, M.; Figurnov, M.; Cowie, A.; Hobbs, N.; Kohli, P.; Kleywegt, G.; Birney, E.; Hassabis, D.; Velankar, S. AlphaFold Protein Structure Database: Massively Expanding the Structural Coverage of Protein-Sequence Space with High-Accuracy Models. *Nucleic Acids Res.* **2022**, *50* (D1), D439–D444. https://doi.org/10.1093/nar/gkab1061.

(107) Li, G.; Rabe, K. S.; Nielsen, J.; Engqvist, M. K. M. Machine Learning Applied to Predicting Microorganism Growth Temperatures and Enzyme Catalytic Optima. *ACS Synth. Biol.* **2019**, *8* (6), 1411–1420. https://doi.org/10.1021/acssynbio.9b00099.

(108) Yang, Y.; Zhao, J.; Zeng, L.; Vihinen, M. ProTstab2 for Prediction of Protein Thermal Stabilities. *Int. J. Mol. Sci.* **2022**, *23* (18), 10798. https://doi.org/10.3390/ijms231810798.

(109) Wang, X.-F.; Gao, P.; Liu, Y.-F.; Li, H.-F.; Lu, F. Predicting Thermophilic Proteins by Machine Learning. *Curr. Bioinforma.* **2020**, *15* (5), 493–502. https://doi.org/10.2174/1574893615666200207094357.

(110) Zhao, J.; Yan, W.; Yang, Y. DeepTP: A Deep Learning Model for Thermophilic Protein Prediction. *Int. J. Mol. Sci.* **2023**, *24* (3), 2217. https://doi.org/10.3390/ijms24032217.

(111) A. Elnaggar; M. Heinzinger; C. Dallago; G. Rehawi; Y. Wang; L. Jones; T. Gibbs; T. Feher; C. Angerer; M. Steinegger; D. Bhowmik; B. Rost. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44* (10), 7112–7127. https://doi.org/10.1109/TPAMI.2021.3095381.

(112) Steinegger, M.; Söding, J. MMseqs2 Enables Sensitive Protein Sequence Searching for the Analysis of Massive Data Sets. *Nat. Biotechnol.* **2017**, *35* (11), 1026–1028. https://doi.org/10.1038/nbt.3988.

(113) Strodthoff, N.; Wagner, P.; Wenzel, M.; Samek, W. UDSMProt: Universal Deep Sequence Models for Protein Classification. *Bioinformatics* **2020**, *36* (8), 2401–2409. https://doi.org/10.1093/bioinformatics/btaa003.

(114) Whisstock, J. C.; Lesk, A. M. Prediction of Protein Function from Protein Sequence and Structure. *Q. Rev. Biophys.* **2003**, *36* (3), 307–340. https://doi.org/10.1017/s0033583503003901.
(115) Freitag, M.; Al-Onaizan, Y. Beam Search Strategies for Neural Machine Translation. In *Proceedings of the First Workshop on Neural Machine Translation*; Luong, T., Birch, A., Neubig, G., Finch, A., Eds.; Association for Computational Linguistics: Vancouver, 2017; pp 56–60. https://doi.org/10.18653/v1/W17-3207.
(116) Johnson, L. S.; Eddy, S. R.; Portugaly, E. Hidden Markov Model Speed Heuristic and Iterative HMM Search Procedure. *BMC Bioinformatics* **2010**, *11* (1), 431. https://doi.org/10.1186/1471-2105-11-431.
(117) Henikoff, S.; Henikoff, J. G. Amino Acid Substitution Matrices from Protein Blocks. *Proc. Natl. Acad. Sci.* **1992**, *89* (22), 10915–10919. https://doi.org/10.1073/pnas.89.22.10915.
(118) Minami, S. ShintaroMinami/PyDSSP, 2024. https://github.com/ShintaroMinami/PyDSSP (accessed 2024-06-06).
(119) Jorda, J.; Yeates, T. O. Widespread Disulfide Bonding in Proteins from Thermophilic Archaea. *Archaea* **2011**, *2011*, 409156. https://doi.org/10.1155/2011/409156.
(120) Beeby, M.; O'Connor, B. D.; Ryttersgaard, C.; Boutz, D. R.; Perry, L. J.; Yeates, T. O. The Genomics of Disulfide Bonding and Protein Stabilization in Thermophiles. *PLOS Biol.* **2005**, *3* (9), e309. https://doi.org/10.1371/journal.pbio.0030309.
(121) Gao, X.; Dong, X.; Li, X.; Liu, Z.; Liu, H. Prediction of Disulfide Bond Engineering Sites Using a Machine Learning Method. *Sci. Rep.* **2020**, *10* (1), 10330. https://doi.org/10.1038/s41598-020-67230-z.
(122) Peccati, F.; Alunno-Rufini, S.; Jiménez-Osés, G. Accurate Prediction of Enzyme Thermostabilization with Rosetta Using AlphaFold Ensembles. *J. Chem. Inf. Model.* **2023**, *63* (3), 898–909. https://doi.org/10.1021/acs.jcim.2c01083.
(123) Clarke, N. D.; Kissinger, C. R.; Desjarlais, J.; Gilliland, G. L.; Pabo, C. O. Structural Studies of the Engrailed Homeodomain. *Protein Sci. Publ. Protein Soc.* **1994**, *3* (10), 1779–1787.
(124) Tripp, K. W.; Sternke, M.; Majumdar, A.; Barrick, D. Creating a Homeodomain with High Stability and DNA Binding Affinity by Sequence Averaging. *J. Am. Chem. Soc.* **2017**, *139* (14), 5051–5060. https://doi.org/10.1021/jacs.6b11323.
(125) Beck, D. A. C.; Daggett, V. Methods for Molecular Dynamics Simulations of Protein Folding/Unfolding in Solution. *Methods* **2004**, *34* (1), 112–120. https://doi.org/10.1016/j.ymeth.2004.03.008.
(126) McCully, M. E.; Beck, D. A. C.; Daggett, V. Promiscuous Contacts and Heightened Dynamics Increase Thermostability in an Engineered Variant of the Engrailed Homeodomain. *Protein Eng. Des. Sel.* **2013**, *26* (1), 35–45. https://doi.org/10.1093/protein/gzs063.
(127) Mayor, U.; Johnson, C. M.; Daggett, V.; Fersht, A. R. Protein Folding and Unfolding in Microseconds to Nanoseconds by Experiment and Simulation. *Proc. Natl. Acad. Sci.* **2000**, *97* (25), 13518–13522. https://doi.org/10.1073/pnas.250473497.
(128) Wittmann, B. J.; Johnston, K. E.; Wu, Z.; Arnold, F. H. Advances in Machine Learning for Directed Evolution. *Curr. Opin. Struct. Biol.* **2021**, *69*, 11–18. https://doi.org/10.1016/j.sbi.2021.01.008.

(129) Ahmad, S.; Kamal, M. Z.; Sankaranarayanan, R.; Rao, N. M. Thermostable Bacillus Subtilis Lipases: In Vitro Evolution and Structural Insight. *J. Mol. Biol.* **2008**, *381* (2), 324–340. https://doi.org/10.1016/j.jmb.2008.05.063.
(130) García-Marquina, G.; Núñez-Franco, R.; Peccati, F.; Tang, Y.; Jiménez-Osés, G.; López-Gallego, F. Deconvoluting the Directed Evolution Pathway of Engineered Acyltransferase LovD. *ChemCatChem* **2022**, *14* (4), e202101349. https://doi.org/10.1002/cctc.202101349.
(131) Notin, P.; Kollasch, A.; Ritter, D.; van Niekerk, L.; Paul, S.; Spinner, H.; Rollins, N.; Shaw, A.; Orenbuch, R.; Weitzman, R.; Frazer, J.; Dias, M.; Franceschi, D.; Gal, Y.; Marks, D. ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 64331–64379.
(132) Nutschel, C.; Fulton, A.; Zimmermann, O.; Schwaneberg, U.; Jaeger, K.-E.; Gohlke, H. Systematically Scrutinizing the Impact of Substitution Sites on Thermostability and Detergent Tolerance for Bacillus Subtilis Lipase A. *J. Chem. Inf. Model.* **2020**, *60* (3), 1568–1584. https://doi.org/10.1021/acs.jcim.9b00954.
(133) Ardern, Z.; Chakraborty, S.; Lenk, F.; Kaster, A.-K. Elucidating the Functional Roles of Prokaryotic Proteins Using Big Data and Artificial Intelligence. *FEMS Microbiol. Rev.* **2023**, *47* (1), fuad003. https://doi.org/10.1093/femsre/fuad003.
(134) Atsavapranee, B.; Stark, C. D.; Sunden, F.; Thompson, S.; Fordyce, P. M. Fundamentals to Function: Quantitative and Scalable Approaches for Measuring Protein Stability. *Cell Syst.* **2021**, *12* (6), 547–560. https://doi.org/10.1016/j.cels.2021.05.009.
(135) Pearson, W. R. An Introduction to Sequence Similarity ("Homology") Searching. *Curr. Protoc. Bioinforma.* **2013**, *42* (1), 3.1.1-3.1.8. https://doi.org/10.1002/0471250953.bi0301s42.
(136) Ochoa, A.; Storey, J. D.; Llinás, M.; Singh, M. Beyond the E-Value: Stratified Statistics for Protein Domain Prediction. *PLOS Comput. Biol.* **2015**, *11* (11), e1004509. https://doi.org/10.1371/journal.pcbi.1004509.
(137) Shen, L.; Liu, Y.; Chen, L.; Lei, T.; Ren, P.; Ji, M.; Song, W.; Lin, H.; Su, W.; Wang, S.; Rooman, M.; Pucci, F. Genomic Basis of Environmental Adaptation in the Widespread Poly-Extremophilic Exiguobacterium Group. *ISME J.* **2024**, *18* (1), wrad020. https://doi.org/10.1093/ismejo/wrad020.
(138) Amangeldina, A.; Tan, Z. W.; Berezovsky, I. N. Living in Trinity of Extremes: Genomic and Proteomic Signatures of Halophilic, Thermophilic, and pH Adaptation. *Curr. Res. Struct. Biol.* **2024**, *7*, 100129. https://doi.org/10.1016/j.crstbi.2024.100129.
(139) Altenhoff, A. M.; Warwick Vesztrocy, A.; Bernard, C.; Train, C.-M.; Nicheperovich, A.; Prieto Baños, S.; Julca, I.; Moi, D.; Nevers, Y.; Majidian, S.; Dessimoz, C.; Glover, N. M. OMA Orthology in 2024: Improved Prokaryote Coverage, Ancestral and Extant GO Enrichment, a Revamped Synteny Viewer and More in the OMA Ecosystem. *Nucleic Acids Res.* **2024**, *52* (D1), D513–D521. https://doi.org/10.1093/nar/gkad1020.
(140) Altenhoff, A. M.; Levy, J.; Zarowiecki, M.; Tomiczek, B.; Vesztrocy, A. W.; Dalquen, D. A.; Müller, S.; Telford, M. J.; Glover, N. M.; Dylus, D.; Dessimoz, C. OMA Standalone: Orthology Inference among Public and Custom Genomes and Transcriptomes. *Genome Res.* **2019**, *29* (7), 1152–1163. https://doi.org/10.1101/gr.243212.118.
(141) Zahn-Zabal, M.; Dessimoz, C.; Glover, N. M. Identifying Orthologs with OMA: A Primer. F1000Research January 17, 2020. https://doi.org/10.12688/f1000research.21508.1.
(142) Velankar, S.; Dana, J. M.; Jacobsen, J.; van Ginkel, G.; Gane, P. J.; Luo, J.; Oldfield, T. J.; O'Donovan, C.; Martin, M.-J.; Kleywegt, G. J. SIFTS: Structure Integration with

Function, Taxonomy and Sequences Resource. *Nucleic Acids Res.* **2013**, *41* (D1), D483–D489. https://doi.org/10.1093/nar/gks1258.

(143) Dana, J. M.; Gutmanas, A.; Tyagi, N.; Qi, G.; O'Donovan, C.; Martin, M.; Velankar, S. SIFTS: Updated Structure Integration with Function, Taxonomy and Sequences Resource Allows 40-Fold Increase in Coverage of Structure-Based Annotations for Proteins. *Nucleic Acids Res.* **2019**, *47* (D1), D482–D489. https://doi.org/10.1093/nar/gky1114.

(144) Chen, Z.; Liu, X.; Zhao, P.; Li, C.; Wang, Y.; Li, F.; Akutsu, T.; Bain, C.; Gasser, R. B.; Li, J.; Yang, Z.; Gao, X.; Kurgan, L.; Song, J. iFeatureOmega: An Integrative Platform for Engineering, Visualization and Analysis of Features from Molecular Sequences, Structural and Ligand Data Sets. *Nucleic Acids Res.* **2022**, *50* (W1), W434–W447. https://doi.org/10.1093/nar/gkac351.

(145) Lambert, T. *DsRed at FPbase*. FPbase. https://www.fpbase.org/protein/dsred/ (accessed 2024-06-06).

APPENDIX A: LEARN2THERMDB

A.1 DATA OVERVIEW

The pipeline, from downloading raw data to validating the learn2therm protein pairs, can be executed with a single command: dvc repro. For executing individual stages of the pipeline, the command dvc repro -s <name_of_stage> can be used. Many stages in the pipeline have parameters that influence runtime behavior and stage outputs, making the pipeline customizable through modifications in the params.yaml file. Detailed descriptions of each pipeline stage and the configurable parameters are outlined below and presented in

Table 4.10.

Data Ingestion

1. `s0.0_get_raw_data_taxa.py`

Pull most recent NCBI 16s r RNA sequences, and OGT records from Enqvist

- Params: `min_16s_len`, `max_16s_len` number of nucleotides required to keep and organism
- Outputs: `data/taxa.parquet`, columns include OGT, 16s sequence, taxid, other taxonomy
- Metrics: `n_taxa` total number of labelled organisms, `taxa_pulled_date` when the data was retrieved

2. `s0.1_get_raw_data_proteins.py`

Retrieve single cell uniprot. Uses FTP to download very large uniprot files

- Inputs: `data/taxa.parquet`
- Outputs: `data/taxa/uniprot/uniprot_pulled_timestamp` indicates when files were pulled
- Metrics: `taxa_pulled_date` when data was retrieved

- Untracked Outputs: The script produces `*.xml.gz` files that are untracked because they take so long to download. DVC ignores them, subsequent calls to the script skip downloading files already present.

3. `s0.2_get_proteome_mdata.py`

Get metadata for UniProt proteomes. Selects one "best" proteome per organism.

- Outputs: `data/uniprot/proteome_metadata.csv`, columns include taxa pair file, number of hits, emissions, total searchable space

4. `s0.3_parse_proteins.py`

Extract minimal protein data and store in an efficient file format. Skip proteins that we don't have OGT for or are from redundant proteomes

- Params: `max_prot_per_file` size of parquet files
- Inputs: `data/taxa/uniprot/uniprot_pulled_timestamp`, `data/taxa.parquet`
- Outputs: `data/proteins`, contains proteins in chunked files of the form `*.parquet`. Columns include protein sequence, database identifiers, and associated taxa IDs, `./data/metrics/s0.3_protein_per_data_distr.csv` table of number of proteins per taxa
- Metrics: `n_proteins` total protein count, `percent_prot_w_struc` fraction of proteins with PDB or AlphaFold id

Data Pairing

5. `s1.0_label_taxa.py`

Assign booleans for taxa as thermophile

- Params: `ogt_threshold` binary thermophile threshold
- Inputs: `data/taxa.parquet`
- Outputs: `data/taxa_thermophile_labels.parquet`
- Metrics: `n_meso`, `n_thermo`

6. `s1.1_get_16s_blast_scores.py`

Compute pairwise BLAST pairings of meso vs thermo 16s rRNA sequences.

- Params: `16s_blast_parameters` (there are a number), `blast_metrics`
- Inputs: `data/taxa*.parquet`, `./data/metrics/s0.3_protein_per_data_distr.csv` (used to skip alignment if taxa has no proteins)

- Outputs: `data/taxa_pairs/alignment/*.parquet` table of taxids and BLAST scores.

7. `s1.2_label_all_pairs.py`

Create a list of taxa pairs that meet a minimum 16s rRNA BLAST score.

- Params: `blast_metric_thresholds` defines thresholds on 16s blast metrics to consider a pair
- Inputs: `data/taxa_pairs/alignment/*.parquet`
- Outputs: `data/taxa_pairs/pair_labels/*.parquet` 1:1 index mapping to data/taxa_pairs/alignment/*.parquet of boolean labels of whether that taxa are a pair
- Metrics: `num_taxa_pairs_conservative` number of pairs that passed thresholds. Only this number will we blastp, `taxa_pair_found_ratio` fraction of pairs with metrics egn_taxa_pairs/(n_therm*n_meso) that will be searched for protein pairs

8. `s1.3_protein_alignment.py`

Runs a massive parallel cluster to align protein pairs among taxa pairs using DIAMOND.

- Params: `dask_cluster_class` class for cluster in dask_jobqueue, `max_protein_length`, `method` local aligner type, `n_jobs` parallel workers, each doing a taxa pair at a time, `method_X_params` where X is eg. "blast" params given to aligner, `blast_metrics` alignment metrics to record.
- Inputs:`data/taxa_pairs/alignment/*.parquet`,`data/taxa_pairs/pair_labels/*.parquet`, `./data/proteins`
- Outputs: `data/protein_pairs/*.parquet`, each file is an aggregation of alignments for a number of taxa pairs with protein pids, source organism taxids, and alignment metrics
- Metrics: `protein_align_X`, where X is "emissions", "hits", "time", "return". The resources and used and return on investment of protein alignment.

9. `s1.4_make_database.py`

Collect processed data files into a relational duckdb database.

- Inputs:`data/taxa.parquet`,`data/taxa_thermophile_labels.parquet`, `data/protein_pairs/*.parquet`,`data/proteins/*.parquet`, `data/uniport/proteome_metadata.csv`
- Outputs: `data/database.ddb` relational duckdb database of taxa, proteins, taxa pairs, and protein pairs

Data Validation

10. ` s2.1_get_hait_pairs.py`

Parse the protein pairs from Hait et al's excel files, query the PDB ids to get sequences.

- Outputs: `data/validation/hait_pairs.csv`, columns include meso PDB id, thermo PDB id, and sequences

11. `s2.2_compare_to_Tm.py`

Compare melting temperatures from FireProtDB and Meltome Atlas to OGTs in the dataset.

- Inputs: `data/database.ddb`
- Metrics: `Tm_OGT_spearman_p`, `Tm_OGT_spearman_r` Spearman R and null probability of relationship between OGT from our data and melting temperature from 3rd party dataset

12. `s2.3_run_hait_alignment.py`

Compute the alignment metrics for Hait et al. pairs using identical parameters as the alignment metrics for the full dataset.

- Params: `method` local aligner type, `method_X_params` where X is eg. "blast" params given to aligner, `blast_metrics` alignment metrics to record.
- Inputs: `data/validation/hait_pairs.csv`
- Outputs: `data/validation/hait_aligned_scores.csv`, columns include protein ids and alignment metrics

13. `s2.3_run_hait_alignment.py`

Compare metrics for BLAST alignments of Hait et al. pairs to our dataset and make some plots.

- Inputs: `data/validation/hait_aligned_scores.csv`, `data/database.ddb`
- Outputs: `data/validation/hait_alignment/*.png` Distribution comparison of alignment metrics, and size of your dataset if we were to use Hait scores as a filter.

14. `s2.3_run_hait_alignment.py`

Download Pfam HMMs.

- Outputs: `./data/validation/hmmer/Pfam-A.hmm`
- Metrics: `HMM_pulled_date` date of Pfam aquisition

15. `s2.3_run_hait_alignment.py`

Run Pfam against proteins in Hait et al. pairs, compute Jaccard of annotations.

- Params: `e_value` Maximum e-value for hmmer to report an annotation, `njobs` cores for parallel
- Inputs: `data/validation/hait_pairs.csv`, `./data/validation/hmmer/Pfam-A.hmm`

- Outputs: `data/validation/hait_scores.csv`, columns include jaccard score of Pfam annotations
- Metrics: `mean_jaccard` Mean jaccard score of annotations over Hait pairs, `fraction_found` Fraction of proteins with at least one Pfam annotation

16. `s2.3_run_hait_alignment.py`

Scan Pfam against all proteins on our database that are in protein pairs.

- Params: `e_value` Maximum e-value for hmmer to report an annotation, `njobs` cores for parallel, `scan` boolean to hmmscan or hmmsearch, `prefetch` whether to prefetch HMMs into memory or leave as file iterator, `chunk_size` size of protein chunks to run and save to file at one time.
- Inputs: `data/database.ddb`, `./data/validation/hmmer/Pfam-A.hmm`
- Outputs: `data/validation/hmmer_outputs/*.parquet` Pfam annotations for each protein, columns include the PID and a 'accessions', a semicolon separated string of Pfam accessions annotated for the PID
- Metrics: `n_proteins_in_pairs` number of proteins in pairs, `n_proteins_labeled` number of proteins with at least one Pfam annotation.

17. `s2.3_run_hait_alignment.py`

Parse hmmer results into a table of protein pairs and their Jaccard scores of annotations

- Inputs: `data/validation/hmmer_outputs/*.parquet`
- Outputs: `data/validation/hmmer_labels/*.parquet`, columns include meso PID, thermo PID, and Jaccard overlap of Pfam annotations
- Metrics: `mean_jaccard` Mean jaccard score of annotations over protein pairs, `fraction_found` Fraction of proteins with at least one Pfam annotation

18. `s2.3_run_hait_alignment.py`

Compare Pfam annotations of Hait et al. pairs to our dataset and make some plots.

- Inputs: `data/validation/hmmer_labels/*.parquet`, `data/database.ddb`
- Outputs: `data/validation/hmmer/compare_jaccard_hist.png` Distribution comparison of alignment Jaccard scores for Hait et al. pairs and our pairs.
- Metrics: `t_pvalue_base` p-value of t-test of Jaccard scores between Hait et al. and our pairs, `t_pvalue_95` p-value of t-test of Jaccard scores between Hait et al. and our pairs with > 95% blast coverage

19. `s2.3_run_hait_alignment.py`

Sample some protein pairs to conduct structural alignment on, uniform over BLAST coverage.

- Params: `sample_size` number of protein pairs to sample, `metrics` list of queries to make from pairs table to sample pairs uniformly over
- Inputs: `data/database.ddb`
- Outputs: `data/validation/structure/sample_l2t_data.csv`, a sample of pairs from the pairs table, columns include meso and thermo PID, and metrics specified in params

20. `s2.11_structure_hait.py`

Run FATCAT structural alignment for Hait et al. pairs by getting PDB structures.

- Inputs: `data/validation/hait_pairs.csv`
- Outputs: `data/validation/structure/hait_fatcat.csv`, column include the P-value from FATCAT for the alignment

21. `s2.11_structure_hait.py`

Run FATCAT structural alignment for L2T pairs by getting PDB or AlphaFold structures.

- Inputs: `data/validation/structure/sample_l2t_data.csv`
- Outputs: `data/validation/structure/l2t_sample_fatcat.csv`, column includes meso and thermo PIDs, metrics originally specified to sample uniformly from in s2.10

Parameter

Table 4.10. All tunable parameters for Learn2thermDB

Stage	Parameter name	Description	Published Value
get_raw_data_taxa	min_16s_len	Minimum number of bases in 16S rRNA sequence to keep the sequence	1300
	max_16s_len	Maximum number of bases in 16S rRNA sequence to keep the sequence	1600
parse_proteins	max_prot_per_file	Number of proteins per chunk for saving UniProtKB as minimal table	100,000
label_taxa	ogt_threshold	Binary threshold to consider a taxa thermophilic, not inclusive	40.0
get_16s_blast_scores	num_threads	BLAST cpus for computation	20
	word_size	Size of initial exact nucleotide matching in BLAST	28
	gapopen_penalty	Penalty for bit score incurred by starting a gap in the alignment	2
	gapextend_penalty	Penalty for bit score incurred by extending a gap in the alignment	1
	reward	Score increase for a match in alignment	1

	penalty	Penalty for mismatch in alignment	-2
	ungapped	Whether to do an alignment without allowing gaps	False
	blast_metrics	List of metrics defined in learn2therm.blast. BlastMetrics to compute for 16S rRNA alignment	* See note 1 below
label_all_pairs	blast_metric_thresholds	Nested dictionary of name of alignment metric and two fields “thresh” which is the threshold for considering a pair, non inclusive, and “greater”, a boolean of whether we want bigger or smaller than the threshold	* See note 2 below
get_protein_blast_scores	dask_cluster_class	Class from Dask Jobqueue1 to use for parallel workers	SLURMCluster
	max_protein_length	Maximum number of amino acids, inclusive, to keep for protein pair search	-
	method	Which aligner to use, one of ‘blast’ or ‘diamond	diamond

	n_jobs	Number of independent parallel workers to keep running	80
	save_frequency	Number of taxa pairs between DVC checkpointing	20,000
	method_blast_params	Parameters for BLASTp algorithm	See Table 4.11 below.
	method_diamond_params	Parameters for DIAMOND algorithm	See Table 4.11 below
	blast_metrics	List of metrics defined in learn2therm.blast. BlastMetrics to compute for protein alignment	* See note 1 below
run_hmmer	e_value	Maximum E value for labelling Pfam annotations with HMMER	1.e-10
	chunk_size	Number of vector chunks to run at a time	2000
	prefetch	Boolean of whether to load HMMs into memory, or leave as disk iterator	True
	njobs	Number of CPUs to use for parallel pyhmmer	32
	scan	Whether to use HMMScan or HMMSearch	False

sample_data_for_structure	sample_size	Number of protein pairs to keep for structural alignment	10,000
	metrics	Queries of protein pairs table to compute and use for sampling, the columns is binned into 5 bins and samples uniformly from those bins	[“(query_align_cov+subject _align_cov)/2.0"]

Table 4.11. Parameters for BLASTp if used

Parameter Name	Description	Published Value
num_threads	CPUs for each worker to run BLASTp	6
word_size	Size of initial Amino Acid matching in BLASTp	3
gapopen	Penalty to bit score for opening a gap in alignment	11
gapextend	Penalty to bit score for extending a gap	1
matrix	Substitution matrix for comparing amino acids in alignment	BLOSUM62
threshold	Minimum score for word to be added to lookup table	11
ungapped	Whether to run alignment without gaps	False
evalue	Maximum E value to keep scoring alignment	0.00001
qcov_hsp_perc	Minimum percent coverage of query strand to keep alignment	75

Table 4.12. Parameters for DIAMOND if used

Parameter Name	Description	Published Value
num_threads	CPUs for each worker to run DIAMOND	6
sensitivity	How sensitive the search should be for initial matching	ultra-sensitive
gapopen	Penalty to bit score for opening a gap in alignment	11
gapextend	Penalty to bit score for extending a gap	1
matrix	Substitution matrix for comparing amino acids in alignment	BLOSUM62
iterate	Whether to start with lower sensitivity alignment	False
global_ranking	Limit on the number of Smith Waterman extensions per query, with the target sequences ranked by their ungapped extension scores.	Null
evalue	Maximum E value to keep scoring alignment	0.00001
qcov_hsp_perc	Minimum percent coverage of query strand to keep alignment	75

* Note 1: Any metric that is a method name of the class learn2therm.blast.BlastMetrics can be specified here. See Section C.1.2 for definitions of these metrics. The values used in this work are: [local_gap_compressed_percent_id, scaled_local_query_percent_id,

scaled_local_symmetric_percent_id, local_E_value, query_align_start, query_align_end, subject_align_end, subject_align_start, query_align_len, query_align_cov, subject_align_len, subject_align_cov, bit_score]

* Note 2: To threshold by E value less than 1e-10, you have to edit the 'params.yaml'

A.2 ALIGNMENT METRICS

A variety of metrics have been calculated for pairwise alignments of 16S sequences and proteins, which are included in the database. Detailed definitions, descriptions, and corresponding database tags for each metric are provided below.

Given an alignment of length L including gaps upon query sequence and subject sequence with lengths A and B, a column at position i ∈ [0,L] has:

- $I_i = 1$ if the column is a match between the two strands, 0 otherwise
- $G_i = 1$ if the column has a gap in the alignment, 0 otherwise
- $G_i^X = 1$ if the column has a gap in the X strand, 0 otherwise
- $GF_i = 1$ if the column has a gap in the alignment and has no gap column to its left, 0 otherwise
- S_i, the bit score of the alignment according to a substitution matrix (BLOSUM62 was used, however this is a free parameter)

$N(X) = \sum_{i=0}^{L} X_i$

M is the number of sequences scanned for alignments.

Table 4.13. Table summarizing definition of alignment equations

Tag	Description	Definition
local_gap_compressed_percent_id	Percent identity of the alignment, with contiguous gaps counted as only one	$\frac{N(I)}{L - N(G) + N(GF)}$
scaled_local_query_percent_id	Percent identity normalized to the query strand	$\frac{N(I)}{A}$
scaled_local_symmetric_percent_id	Percent identity normalized to the average strand length	$\frac{2\,N(I)}{A + B}$
bit_score	BLAST bit score	$\mathrm{N(S)}$
local_E_value	BLAST E value	$\frac{M\,A}{2^{N(S)}}$
query/subject_align_cov	Coverage of alignment on strand	$\frac{L - N(G^{query})}{A}$, $\frac{L - N(G^{subject})}{B}$

Note: Start and end positions of the alignment on a strand is also tracked as query/subject_align_start/end. The difference between start and end, e.g., the length of the alignment per strand is tracked as query/subject_align_len.

16S rRNA alignment

Taxa Pairs were determined via alignment of 16s rRNA strands from Refseq using BLASTn 2.14.0+. The parameters for the search are shown in Table 4.14 below, default if not listed.

Table 4.14. Table of BLASTn parameters

Parameter	Value
word_size	28
gapopen	2
gapextend	1
reward	1
penalty	-2
ungapped	False
evalue	1.0
perc_identity	0.5

The best high-scoring pair (HSP) was identified as the one with the highest average coverage across both strands and retained for each putative pair. Taxa were classified as pairs if the coverage for both strands was at least 98.5% and the gap-compressed percent identity was 81% or higher.

Protein alignments

Protein Pairs were determined via alignment of UniProtKB sequence using DIAMOND version 2.1.6, a BLAST-like software. The parameters for the search are shown in Table 4.15 below, default if not listed.

Table 4.15. Table of DIAMOND parameters

Parameter	Value
word_size	3
sensitivity	ultra-sensetive
iterate	False
matrix	BLOSSUM62
global_ranking	Null
gapopen	11
gapextend	1
evalue	0.00001
Max_hsps	100
id	0
query-cover	75
subject-cover	75

Cost of alignment

Before proceeding with protein alignment across numerous taxa pairs and risking computational resources, we compared the performance of BLASTp to DIAMOND using identical parameters wherever possible, such as word size, scoring matrix, and penalties. This comparison involved aligning 10,000 random proteins against another set of 10,000 random proteins. The results, which include compute time, carbon footprint, and the return on investment in terms of the number of hits, are presented in Figure 3.4.1. DIAMOND was selected for alignment due to its marginally lower sensitivity but significantly reduced computational cost, nearly an order of magnitude less.

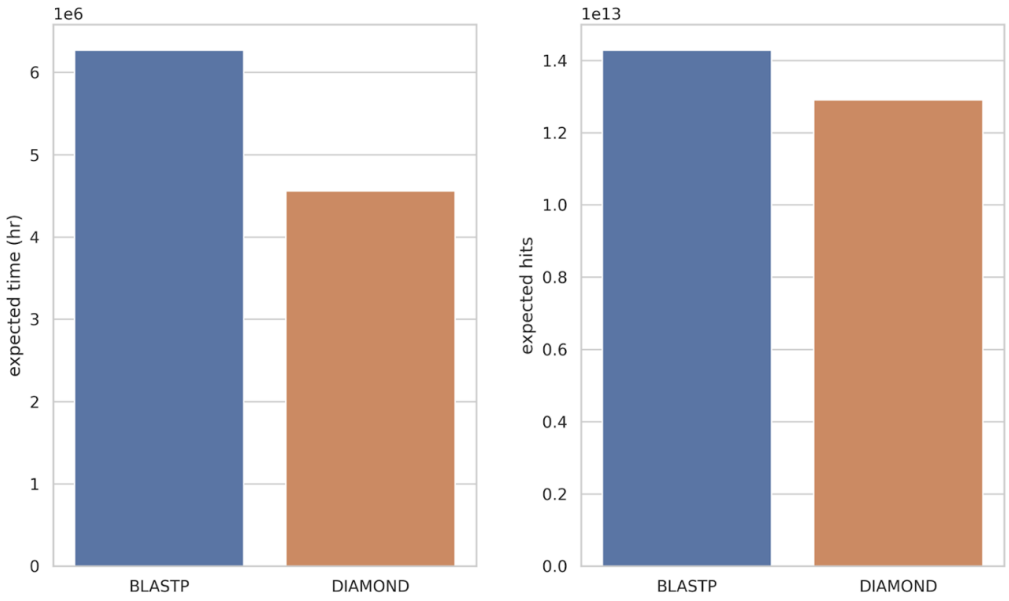

Figure 3.4.1. Bar plots for BLASTp vs DIAMOND resource testing

A.3 DATABASE SCHEMA

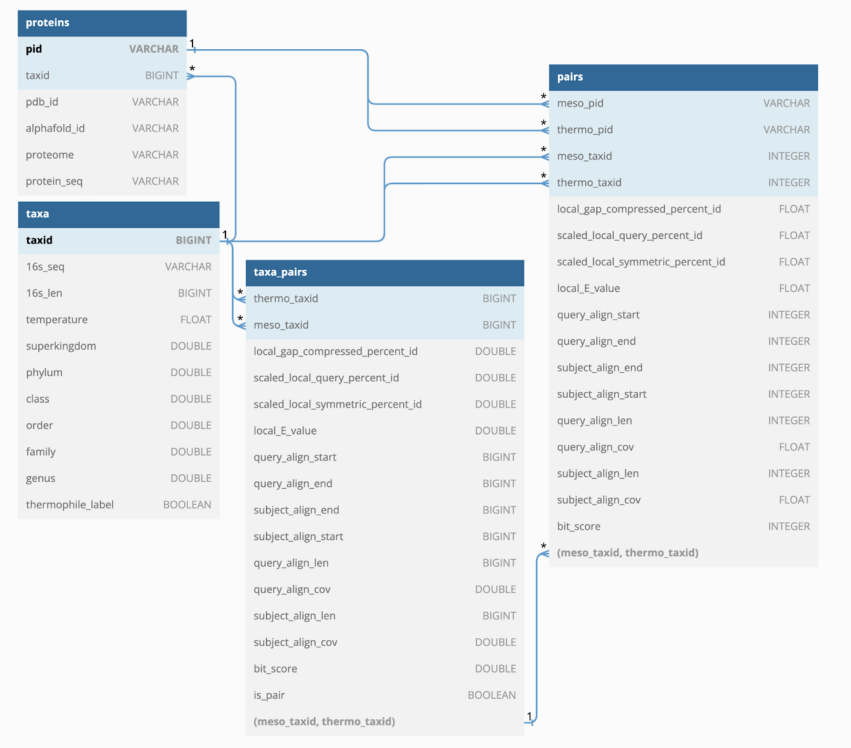

Figure 3.4.2. Schematic of the presented database. The 'taxa' and 'proteins' tables compile raw data from sources including UniProtKB, RefSeq, and Engqvist. The 'taxa_pairs' and 'pairs' tables document the alignment results for 16S rRNA and protein sequences, respectively. For detailed descriptions of each field in the database, refer to Table 4.16 below.

Table 4.16. Entities within learn2thermDB

Table	Field	Description
Taxa	taxid	Primary key, NCBI taxid
	16s_seq	Nucleotides of 16S rRNA sequence
	16s_len	Length of 16s rRNA strand
	temperature	Optimal Growth Temperature
	superkingdom	NCBI superkingdom integer identifier
	phylum, ..., genus	NCBI integer identifier for taxonomy
	thermophile_label	Indicator if organism is a thermophile as determined by pipeline parameters
Proteins	pid	Primary key, UniProtKB identifier
	taxid	NCBI taxid of host organism, foreign key to taxa
	pdb_id	PDB cross reference if present
	alphafold_id	AlphaFold database cross reference
	proteome	Identifier of UniProtKB proteome that protein is a part of (if present)
	protein_seq	Amino acids of the sequence
Taxa_pairs	thermo_taxid	NCBI taxid of thermophile, foreign key to taxa. Serves as compound primary key with meso_taxid
	meso_taxid	NCBI taxid of mesophile, foreign key to taxa. Serves as compound primary key with thermo_taxid
	is_pair	Indicator if pair of taxa is considered a pair by pipeline parameters for downstream protein alignments.
	*	A number of alignment metrics were computed. See Appendix A.2
Pairs	meso_pid	UniProtKB identifier of mesophilic protein, foreign key to proteins. Serves as compound primary key with thermo_pid
	thermo_pid	UniProtKB identifier of thermophilic protein, foreign key to proteins. Serves as compound primary key with meso_pid
	meso_taxid	NCBI taxid of mesophile, foreign key to taxa. Serves as compound foreign key with thermo_taxid to taxa_pairs
	thermo_taxid	NCBI taxid of thermophile, foreign key to taxa. Serves as compound foreign key with meso_taxid to taxa_pairs
	*	A number of alignment metrics were computed. See Appendix A.2

A.4 PFAM ANNOTATIONS

Pfam was utilized to annotate proteins within pairs using 19,632 HMMs from its version 35.0, processed via pyhmmer. Only annotations with a sequence hit E-value less than 1e-10 were retained. Subsequently, the set Jaccard score was calculated for the annotations between two proteins as defined below:

$$\mathrm{J} = \frac{\cap(A,B)}{\cup(A,B)} \tag{4.2}$$

In this context, sets A and B represent the Pfam annotations for proteins A and B, respectively. For example, if protein A is labeled with the families “pfam_0” and “pfam_1,” and protein B is labeled with only “pfam_0,” the Jaccard score would be 0.5. If Pfam fails to find annotations for one of the proteins in a pair that meet the minimum HMMER criteria, the Jaccard score is assigned a value of 0.0. Protein pairs for which neither protein received an annotation were excluded from the analysis, as these typically represent unknown protein families.

A.5 OGT PREDICTOR TRAINING

ProtBERT was fine-tuned to classify proteins from the learn2therm dataset as either thermophilic or mesophilic based solely on their sequences. The preprocessing steps for this fine-tuning included selecting 10 million proteins from the database with sequence lengths of 250 amino acids or fewer. Proteins with Optimal Growth Temperatures (OGT) in the range of [30°C, 60°C) were excluded from the dataset. Those with higher temperatures were labeled as thermophilic, while the rest were labeled as mesophilic. To balance the dataset, it was randomly downsampled to include 175,000 proteins from each category.

To minimize redundancy, a MinHash distance removal was performed on sequence tokens, a technique often used in traditional natural language datasets to reduce data leakage. This involved

computing the set of all 3-mers for each amino acid sequence, calculating the MinHash for these sets using 100 permutations, and then clustering the proteins based on a MinHash Jaccard score greater than 0.6. Only one protein from each cluster was retained.

The dataset was subsequently split into development and test sets, with 10% of the data reserved for testing. This split was done using random assignment based on the protein's taxonomic identifier (taxid) to ensure that no two proteins from the same organism were present in both sets.

For the model architecture, we utilized ProtBERT with pre-trained weights from Huggingface, augmented with a newly added Multilayer Perceptron (MLP) layer to serve as a binary predictor. The pre-trained embeddings of size 1024 were pooled using a linear projection of the same size and then averaged over the amino acids in the sequence, followed by a binary predictor with two neurons. Below is the PyTorch readout for the model, employing neural modules from Huggingface's ecosystem. For more details on the model classes, see https://huggingface.co/docs/transformers/v4.30.0/en/model_doc/bert.

```
BertForSequenceClassification( (bert): BertModel(

(embeddings): BertEmbeddings(
(word_embeddings): Embedding(30, 1024, padding_idx=0) (position_embeddings):
Embedding(40000, 1024) (token_type_embeddings): Embedding(2, 1024)
(LayerNorm): LayerNorm((1024,), eps=1e-12, elementwise_affine=True) (dropout):
Dropout(p=0.0, inplace=False)

)
(encoder): BertEncoder(

(layer): ModuleList( (0-29): 30 x BertLayer(

(attention): BertAttention( (self): BertSelfAttention(

(query): Linear(in_features=1024, out_features=1024, bias=True) (key):
Linear(in_features=1024, out_features=1024, bias=True)

(value): Linear(in_features=1024, out_features=1024, bias=True)

(dropout): Dropout(p=0.0, inplace=False) )

(output): BertSelfOutput(
(dense): Linear(in_features=1024, out_features=1024, bias=True) (LayerNorm):
LayerNorm((1024,), eps=1e-12, elementwise_affine=True) (dropout): Dropout(p=0.0,
inplace=False)
```

```
) )

(intermediate): BertIntermediate(
(dense): Linear(in_features=1024, out_features=4096, bias=True) (intermediate_act_fn):
GELUActivation()

)
(output): BertOutput(

(dense): Linear(in_features=4096, out_features=1024, bias=True) (LayerNorm):
LayerNorm((1024,), eps=1e-12, elementwise_affine=True) (dropout): Dropout(p=0.0,
inplace=False)

) )

) )

(pooler): BertPooler(
(dense): Linear(in_features=1024, out_features=1024, bias=True) (activation): Tanh()

) )

(dropout): Dropout(p=0.0, inplace=False)

(classifier): Linear(in_features=1024, out_features=2, bias=True) )
```

The model was then fine-tuned for two epochs on the 280k training proteins. The training parameters are given below:

```
TrainingArguments( _n_gpu=1, adafactor=False, adam_beta1=0.9,

adam_beta2=0.999, adam_epsilon=1e-08, auto_find_batch_size=False, bf16=False,
bf16_full_eval=False, data_seed=None, dataloader_drop_last=False,
dataloader_num_workers=0, dataloader_pin_memory=True, ddp_bucket_cap_mb=None,
ddp_find_unused_parameters=None, ddp_timeout=1800,

debug=[],
deepspeed=None,
disable_tqdm=False,
do_eval=True,
do_predict=False,
do_train=True, eval_accumulation_steps=25, eval_delay=0,
eval_steps=6, evaluation_strategy=steps,
fp16=True,
fp16_backend=auto, fp16_full_eval=False, fp16_opt_level=O1,
fsdp=[],
fsdp_min_num_params=0, fsdp_transformer_layer_cls_to_wrap=None,
full_determinism=False, gradient_accumulation_steps=25, gradient_checkpointing=True,
greater_is_better=False, group_by_length=False, half_precision_backend=cuda_amp,
hub_model_id=None, hub_private_repo=False, hub_strategy=every_save,
hub_token=<HUB_TOKEN>, ignore_data_skip=False,

save_on_each_node=False, save_steps=6, save_strategy=steps, save_total_limit=None,
seed=42,
sharded_ddp=[], skip_memory_metrics=True, tf32=None,
torch_compile=False, torch_compile_backend=None, torch_compile_mode=None,
```

```
torchdynamo=None, tpu_metrics_debug=False, tpu_num_cores=None, use_ipex=False,
use_legacy_prediction_loop=False, use_mps_device=False, warmup_ratio=0.0,
warmup_steps=0, weight_decay=0.0, xpu_backend=None,


)
```

The final model model was saved and evaluated against the test set. It can be found on the Huggingface Hub, https://doi.org/10.57967/hf/0815.

A.6 OGT PREDICTOR TRAINING

For the 2D mapping of proteins (see Figure 2.2.1-right), ESM2 embeddings were employed. Initially, embeddings from the ESM Atlas were retrieved for proteins with a Tm (melting temperature) greater than 0.9 and a pLDDT (predicted Local Distance Difference Test) score above 0.9, indicating high confidence in the structural predictions of the ESM language model. These embeddings, representing the average over sequence tokens from the model's output latent space, produced a single vector of size 2,560 for each protein sequence.

Due to the high computational demands, the dataset was reduced through random sampling to make dimensionality reduction feasible. From the ESM Atlas, only 500,000 proteins were selected. Additional random sampling was conducted for proteins from the learn2therm dataset and Hait et al.'s dataset, maintaining proportional representation relative to the original dataset sizes and the ESM Atlas. This process yielded 240,000 proteins from the learn2therm dataset and 82 from Hait et al.'s dataset.

The same version of the pretrained model, 'esm2_t36_3B_UR50D,' which was used for the ESM Atlas, was employed to embed proteins from the learn2therm and Hait datasets. Subsequently, dimensionality reduction was performed using multicore T-SNE to project the embedded vectors down to two dimensions, utilizing the following specific parameters, with all others left at their default settings.

Table 4.17. Summary of the parameters used for the T-SNE dimensionality reduction of ESM embeddings

Parameter	Value
n_iter_early_exag	500
n_iter	2000
perplexity	30.0
theta	0.5

APPENDIX B: NOMELT

B.1 PARAMETERS

Pairwise alignment

Case studies discussed in the main text, including En-HD, LovD, and LipA, were BLASTed against the training set of NOMELT to verify that no close homologs were present during training. The parameters used were:

```
-evalue 2.0 -outfmt 5 -num_threads 32 -word_size 3 -matrix BLOSUM62 - qcov_hsp_perc 80
```

Biopython Smith-Waterman alignment algorithm was used to generate mutations. The parameters were:

```
matrix: BLOSUM62 match_score: 1 mismatch_score: -1 gapopen: -4
gapextend: -1 penalize_end_gaps: false
```

Jackhmmer parameters

Jackhmmer was employed to generate Multiple Sequence Alignments (MSAs) for each test set example against other thermophilic sequences, facilitating the computation of positional cross entropy to assess natural variation. The parameters used matched those of the AF2 searches:

```
--noali --F1 0.0005 --F2 0.00005 --F3 0.0000005 --incE 0.0001 -E 0.0001 --cpu 32
-N 1
```

MMSeqs2 parameters

```
--min-seq-id 0.5 --cluster-reassign 1 --cluster-steps 5 -s 7 --max-seqs 1000 -c 0.95 --cov-mode 0 --similarity-type 2 -e 1e-3 --cluster-mode 1 --threads 32
```

B.2 HYPERPARAMETERS

NOMELT is a fine-tuned from ProtT5-XL, the Pytorch readout is given below:


```
{
"_name_or_path": "Rostlab/prot_t5_xl_uniref50", "architectures": [

"T5ForConditionalGeneration"

],
"classifier_dropout": 0.0,
"d_ff": 16384,
"d_kv": 128,
"d_model": 1024, "decoder_start_token_id": 0, "dense_act_fn": "relu", "dropout_rate": 0.1,
"eos_token_id": 1, "feed_forward_proj": "relu", "initializer_factor": 1.0,
"is_encoder_decoder": true, "is_gated_act": false, "layer_norm_epsilon": 1e-06,
"model_type": "t5",
"n_positions": 512, "num_decoder_layers": 24,
"num_heads": 32,
"num_layers": 24,
"output_past": true,
"pad_token_id": 0, "relative_attention_max_distance": 128,
"relative_attention_num_buckets": 32, "transformers_version": "4.34.0.dev0",
"use_cache": true,

"vocab_size": 128 }
```


The parameters and their final versions used to prepare the data and train the model are given in Table 4.18.

Table 4.18. NOMELT final model parameters

Parameter	Final Value	Description
	Data Preparation	
min_thermo_temp	60.0	Temperature to consider a thermophile
min_temp_diff	0.0	Minimum difference in temps to consider pair
max_meso_temp	40.0	Maximum temperature to consider a mesophile
min_align_cov	0.95	Minimum protein pair alignment coverage (avg)
mmseq_params	coverage: 0.95	Parameters passed to MMSeqs to cluster dataset for splitting
	min-seq-id: 0.5	
	cluster-mode: 1	
	similarity-type: 2	
	sensitivity: 7	
	max-seqs: 1000	
	cluster-steps: 5	
	cluster-reassign: 1	
	e: 1e-3	
test_size	0.1	Fraction used to split dev and test set, and val set from dev set. Approximate, based on MMSeqs' cluster sizes
additional_filters	'abs({seq_len_diff})/ {meso_seq_len}<0.1 '	Statement that can be evaluated on columns in the protein pairs learn2therm dataset to filter it
	Model Architecture	
pretrained_model	Rostlab/prot_t5_xl_uniref50	Foundational model to start with
model_hyperparams	dropout_rate: 0.1	Hyperparameters directly passed to Huggingface T5Params class
	relative_attentin_max_distance: 250	
generation_max_length	250	Max size of stochastic generations

generation_num_beams	10	Number of beams for BEAM search
	Training	
reweight	False	Whether to weigh examples in training set by inverse cluster size for loss computation
freeze_early_layers	0.2	Fraction of T5 blocks to be frozen, bottom-up
epochs	1	Number of training epochs for the evalution model. Ignored for the final model which trains the same fraction of steps as the eval model did
early_stopping	True	Whether to stop training early for the evalutation model
early_stopping_patience	4	Steps to wait for improvement before stopping
early_stopping_threshold	0.01	Evaluation loss required to count as improvement
per_device_batch_size	20	Size of training examples per GPU per step
gradient_accumulation	8	Batches of gradients to accumulate before taking a backward step
learning_rate	1.e-4	Optimizer learning rate
gradient_checkpointing	True	Whether to use gradient checkpointing
evals_per_save	3	Number of evaluation steps between model evaluation, for checkpointing and early stopping
evals_per_epoch	500	Number of evaluation steps per epoch during training
lr_scheduler_type	‘linear’	Type of learning rate schedule over training epochs
warmup_ratio	0.1	Fraction of total training steps spent warming up to max from 0.0 learning rate
label_smoothing_factor	0.001	Smoothing for one-hot encoded labels
optim	‘adamw_hf’	Optimizer type used

optim_args	Null	Additional hyperparameters for optimizer
bf16	True	Use bf16 parameter precision
fp16	False	Use fp16 parameter precision
eval_single_example_per_cluster	True	Use one random example from each validation set cluster to run evaluation, otherwise whole set
Training Parameters		
only_one_from_cluster	True	Use one random example from each test set cluster for held out testing, otherwise whole set
Mutation Optimization		
use_relaxed	False	Whether to run AMBER relaxation for AF2 predictions
fix_msas	True	Whether to skip MSA generation for AF2 inputs that have already been run
num_replicates	25	Number of stochastic AF2 predictions per protein
residue_length_norm	True	Whether to normalize average Rosetta free energy of AF2 ensemble by number of amino acids
n_trials	100	Number of trials to run, each with a sample of the set of mutations
sampler	NSGAIISampler	Method of sampling next mutations from a set of mutations
population_size	10	Size of generation for each round of evolutionary optimization
cut_tails	Null	Number of residues to keep off the end of a mutant when aligned with the wild type when determining mutations suggested by the model
gap_compressed_mutations	True	Compress gaps suggested by model into a single mutation
matrix	‘BLOSUM62’	Matrix used for alignment when determining mutations suggested by the model

match_score	1	Score for matches when aligning wild type to model translation
mistmatch_score	-1	Score for mismatches when aligning wild type to model translation
gapopen	-4	Penalty for opening a gap when aligning wild type to model translation
gapextend	-1	Penalty for extending a gap when aligning wild type to model translation
penalize_end_gaps	False	Whether to penalize end gaps when aligning wild type to model translation

B.3 MOLECULAR DYNAMICS

Molecular dynamics simulations of the En-HD variants were conducted using GROMACS, initiated from PDB files that were converted into the .gro format. The simulation procedure involved several steps, starting with positioning the protein within a simulation box. The box was then filled with solvent molecules to create a solvated environment around the protein, followed by the addition of ions to neutralize the system:

```

gmx editconf -f chd.gro -o chd_box.gro -c -d 1.0 -bt cubic gmx solvate -cp
chd_box.gro -o chd_solv.gro -p topol.top

gmx grompp -f ions.mdp -c chd_solv.gro -p topol.top -o ions.tpr -maxwarn 1 gmx genion
-s ions.tpr -o chd_ions.gro -p topol.top -neutral

```

NPT Equilibration:

```
;
; GROMACS
; Input for NPT
;
;
```

```
;define = -DPOSRES
integrator =  md
nsteps =  100000
dt= 0.002
;
; Removing CM Translation and Rotation

 comm_mode = Linear

nstcomm = 1000

;
; Output Control

Nstlog = 1000
nstenergy = 100
nstxout = 0

nstvout = 0

nstxtcout = 1000

nstfout = 0

;

; Neighbour Searching

Nstlist = 10

ns_type = grid

pbc = xyz

rlist = 1.0

;

; Electrostatic

rcoulomb = 1.0

coulombtype = pme

fourierspacing = 0.12

;periodic_molecules = yes

;

; VdW
```

```
vdw-type = shift

rvdw = 1.0

;

; Constraints
constraints = h-bonds

constraint-algorithm = lincs

lincs_iter = 4
;
; Temperature
Tcoupl = v-rescale
tc_grps = system
tau_t = 0.1

ref_t = 298.15
;
; Pressure
Pcoupl =  berendsen
Pcoupltype =  isotropic
tau_p =  1.0
compressibility = 4.5e-5

ref_p = 1.0
;
; Initial Velocities

gen_vel = yes

gen_temp = 298.15

gen_seed = -1
```

Production Run:

```
Integrator = md

dt = 0.002

nsteps = 500000000     ;1us

nstxtcout = 40000

nstvout = 0

nstfout = 0

nstcalcenergy = 100

nstenergy = 10000

nstlog = 10000
```

```
;
cutoff-scheme = Verlet
nstlist = 10
vdwtype = Cut-off
vdw-modifier = Force-switch
rvdw_switch = 1.0
rvdw = 1.2
rcoulomb = 1.2
rlist = 1.2
coulombtype = PME
;
tcoupl = V-rescale
tc_grps = System
tau_t = 1.0
ref_t = 298
;
;
constraints = h-bonds
constraint_algorithm = LINCS
continuation = yes

;
Nstcomm = 100
comm_mode = linear
;
```

APPENDIX C: PAIRPROPHET

C.1 ALIGNMENT

A variety of metrics have been calculated for pairwise alignments of proteins, which are included in the training of PairProphet. Detailed definitions, descriptions, and corresponding database tags for each metric are provided below.

Given an alignment of length L including gaps upon query sequence and subject sequence with lengths A and B, a column at position i ∈ [0,L] has:

- I_i = 1 if the column is a match between the two strands, 0 otherwise
- G_i = 1 if the column has a gap in the alignment, 0 otherwise
- G_i^X = 1 if the column has a gap in the X strand, 0 otherwise
- GF_i = 1 if the column has a gap in the alignment and has no gap column to its left, 0 otherwise
- S_i, the bit score of the alignment according to a substitution matrix (BLOSUM62 was used, however this is a free parameter)

$N(X) = \sum_{i=0}^{L} X_i$

M is the number of sequences scanned for alignments.

Table 4.19. Table summarizing definition of alignment equations

Tag	Description	Definition
global_gap_compressed_percent_id	Percent identity of the alignment, with contiguous gaps counted as only one	$\frac{N(I)}{L - N(G) + N(GF)}$
scaled_ global_query_percent_id	Percent identity normalized to the query strand	$\frac{N(I)}{A}$
scaled_ global_symmetric_percent_id	Percent identity normalized to the average strand length	$\frac{2\,N(I)}{A + B}$
bit_score	BLAST bit score	N(S)
global_E_value	BLAST E value	$\frac{M\,A}{2^{N(S)}}$
query/subject_align_cov	Coverage of alignment on strand	$\frac{L - N(G^{query})}{A}$, $\frac{L - N(G^{subject})}{B}$

Note: Start and end positions of the alignment on a strand is also tracked as query/subject_align_start/end. The difference between start and end, e.g., the length of the alignment per strand is tracked as query/subject_align_len.

C.2 IFEATUREOMEGACLI

Table 4.20. Table summarizing selected iFeatureOmegaCLI features

Feature	Name	Definition
AAC	Amino Acid Composition	The Amino Acid Composition (AAC) encoding (2) calculates the frequency of each amino acid type in a protein or peptide sequence
GAAC	Enhanced Amino Acid Composition	The Enhanced Amino Acid Composition (EAAC) feature calculates the AAC based on the sequence window of fixed length that continuously slides from the N- to C-terminus of each peptide and can be usually applied to encode the peptides with an equal length
DistancePair	PseAAC of distance-pair and reduced alphabet	The descriptor incorporates the amino acid distance pair coupling information and the amino acid reduced alphabet profile into the general pseudo amino acid composition vector.
CTDC	Composition/Transition/Distribution for hydrophobicity	The global compositions (percentage) of polar, neutral and hydrophobic residues of the protein
CTDT	Composition/Transition/Distribution for transition from polar to neutral	A transition from the polar group to the neutral group is the percentage frequency with which a polar residue is followed by a neutral residue or a neutral residue by a polar residue
CTDD	Composition/Transition/Distribution for distribution of polarity	The Distribution descriptor consists of five values for each of the three groups (polar, neutral and hydrophobic)
CTriad	Conjoint triad	the properties of one amino acid and its vicinal amino acids by regarding any three continuous amino acids as a single unit
GDPC type 1	Grouped Di-Peptide Composition type 1	A variation of the dipeptide composition composed of 20 descriptors
GDPC type 2	Grouped Di-Peptide Composition type 2	The raw count of the 25 grouped amino acid pairs.
CKSAAGP type 1	Composition of k-Spaced Amino Acid Group Pairs type 1	Calculates the frequency of amino acid group pairs separated by any k residues

CKSAAGP type 2	Composition of k-spaced Amino Acid Pairs type 2	Calculates the raw count of amino acid pairs separated by any k residues.
PseKRAAC type 2	Pseudo K-tuple reduced amino acids composition	Reduced amino acid alphabet based on physiochemical features, which has shown to be predictive for protein analysis
PseKRAAC type 3A	-	Please check iFeatureOmega manual for more info
PseKRAAC type 7	-	-
PseKRAAC type 9	-	-
Geary	Geary autocorrelation descriptor	Please check iFeatureOmega manual for more info
APAAC	Amphiphilic Pseudo-Amino Acid Composition	Composition description, which can predict cellular function
QSOrder	Quasi-sequence-order	A descriptor which have been successfully applied to protein subcellular location prediction