ProVADA: Generating Subcellular Protein Variants via Ensemble-Guided Test-Time Steering

Wenhui Sophia Lu [*1] **Xiaowei Zhang** [*2 3] **Luis S. Mille-Fragoso** [2 3 4] **Haoyu Dai** [5] **Xiaojing J. Gao** [5 4 2]
Wing Hung Wong [1 6]

Abstract

Engineering protein variants that retain functionality in non-native environments remains a significant challenge due to the intricate topology of sequence-fitness landscapes. Experimental strategies often require extensive labor and domain expertise. While recent advances in protein generative modeling offer a promising *in silico* alternative, many of these methods rely on differentiable fitness predictors, which limits their applicability. To this end, we introduce **P**rotein **V**ariant **ADA**ptation (ProVADA), an ensemble-guided, test-time steering framework that integrates implicit generative priors with fitness oracles via a unified composite objective. ProVADA leverages Mixture-Adaptation Directed Annealing (MADA), a novel sampler integrating population-annealing, adaptive mixture proposals, and directed local mutations. Furthermore, ProVADA requires no gradients or explicit likelihoods, yet efficiently concentrates sampling on high-fitness, low-divergence variants. We demonstrate its effectiveness by redesigning human renin for cytosolic functionality. Our results achieve significant gains in predicted localization fitness while preserving structural integrity.

[*]Equal contribution [1]Department of Statistics, Stanford University, Stanford, CA, USA [2]Department of Bioengineering, Stanford University, Stanford, CA, USA [3]Sarafan ChEM-H, Stanford University, Stanford, CA, USA [4]Stanford Bio-X, Stanford University, Stanford, CA, USA [5]Department of Chemical Engineering, Stanford University, Stanford, CA, USA [6]Department of Biomedical Data Science, Stanford University, Stanford, CA, USA. Correspondence to: Wenhui Sophia Lu <sophialu@stanford.edu>, Xiaowei Zhang <zhangxw@stanford.edu>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. Introduction

Protein engineering—the search for sequence variants that exhibit desired functional properties—is a foundational technology in biological engineering. However, purely experimental approaches remain challenging mostly due to the combinatorial 20^L (where L: sequence length) sequence space to be explored. The challenge is further compounded by the complexity of the underlying fitness landscape, a subset of the sequence space where the protein exerts desired functionality. Consequently, experimental approaches often require extensive domain expertise and large-scale, iterative rounds of mutagenesis and screening, both of which are labor-intensive and costly.

One particularly challenging instance is engineering protein variants to function in radically different environments, e.g. from extracellular to cytoplasm. This problem is pressing for two reasons. First, protein activity is highly context-dependent: subcellular compartments differ markedly in pH, redox potential, ionic strength, and other physicochemical parameters, all of which can compromise fold stability and catalytic function in a non-native environment (4). Second, many biotechnological and therapeutic applications demand proteins to operate reliably outside their endogenous environment, yet such repurposing frequently leads to impaired functionalities (5).

Recent advances in machine learning demonstrate great potential in overcoming those bottlenecks. Protein language models trained on millions of sequences capture evolutionary constraints and can propose viable mutations (24; 25). On the other hand, diffusion-based generative models provide an alternative paradigm for efficiently sampling plausible variants. Finally, state-of-the-art, accurate structure prediction methods and downstream inverse-folding networks enable conditional sequence design that preserves a target fold (23; 1; 10). By proposing promising candidates, these *in silico* and hybrid approaches dramatically reduce the search space and, in turn, accelerate experimental protein engineering campaigns (18; 20; 28).

While pretrained priors capture certain fitness attributes like stability, they offer limited guidance for more com-

plex, higher-order fitness objectives—such as environment- or localization-specific functionality—resulting in inefficient optimization. To this end, we present **Pro**tein **V**ariant **ADA**ptation (ProVADA), an ensemble-guided, test-time steering framework for engineering functional proteins targeted to specific subcellular locations. ProVADA employs a novel Mixture-adaptation-directed Annealing sampling algorithm to efficiently explore the vast sequence space under multiple expert guidance.

1.1. Contributions

We summarize our key contributions and the advantages of ProVADA from four perspectives:

- **Likelihood-free generative priors.** ProVADA is compatible with any base generative model, including autoregressive (e.g., ProteinMPNN), transformer-based (e.g., ESM2), or diffusion models, *without* requiring explicit density or likelihood evaluation.
- **Test-time, ensemble-guided sampling.** By optimizing a composite functional objective, ProVADA effectively leverages an "ensemble of expert models" to direct sequence generation in a fully gradient-free manner.
- **Mixture-Adaptation Directed Annealing (MADA).** We develop a novel sampling framework that integrates population annealing and directed local mutations to efficiently explore complex fitness landscapes.
- **Application to the *in silico* redesign of cytosolic renin.** We apply ProVADA to the difficult task of engineering the catalytic domain of human renin for cytosolic functionality. Our results demonstrate significant improvements in predicted localization fitness while preserving structural integrity compared to rejection-sampling baselines.

1.2. Related literature

Classifier-guided and plug-and-play methods steer sequence generation by backpropagating through a differentiable surrogate objective (13; 9; 2). By applying gradients of this proxy reward during generation, these methods bias a pretrained generative model towards desired properties. However, these approaches rely on differentiable surrogate models and thus cannot accommodate non-differentiable or black-box scoring functions. On the other hand, fine-tuning and preference-learning methods, such as classifier-free guidance (19) and reinforcement-learning-based fine-tuning, adapt model parameters to optimize downstream reward objectives. While effective, these methods typically require substantial computational resources and large amounts of labeled datasets. Additionally, they can suffer from catastrophic forgetting, where the model loses previously acquired knowledge from pertaining.

Sampling-based test-time steering methods instead generate candidate sequences from a base model and evaluate them with external scoring functions. Simple strategies merely filter top-scoring samples, whereas more informed approaches use ProteinMPNN to fill masked positions before applying a selection step (28). Though easy to implement, these methods often struggle when exploring high-dimensional sequence spaces with complex fitness landscapes, especially when the base model's distribution poorly aligns with regions of high fitness.

2. ProVADA: Test-Time Steering Ensemble-Guided Protein Variant Adaptation

Problem setup and notation Let the protein sequence length be L, and define the discrete sequence space $\mathcal{X} = \{1, \ldots, 20\}^L$ whose elements encode all possible amino-acid strings of length L. We start from a given wild-type reference sequence $x_0 \in \mathcal{X}$. To generate candidate variants, we assume access to a generative model (e.g. ProteinMPNN) that induces an *implicit* prior $p_\phi(x)$ over $\mathcal{X}$ from which we can efficiently draw samples, even if it may lack an explicit, tractable density form. Additionally, let $F : \mathcal{X} \to \mathbb{R}$ be a potentially black-box, gradient-free fitness oracle, where larger values correspond to superior fitness. Our objective is to discover variants x of x_0 that both conform to this prior and yield high scores under the fitness function.

At a high level, ProVADA consists of two key components. First, we train a classifier that predicts the target localization on a dataset with labeled sequence-fitness pairs. Second, given a reliable fitness function $F(x)$ and an initial wild-type sequence x_0, we wish to efficiently generate protein variants of x_0 from our generative prior that also achieve high fitness. Specifically, we construct a target sampling distribution proportional to the generative prior exponentially tilted by the tempered fitness score:

$$H_\lambda(x) = \underbrace{[F(x) - \lambda\, D(x_0, x)]}_{\text{adjusted fitness}},$$

$$\pi_{\phi,\tau,\lambda}(x) = \frac{p_\phi(x) \exp\big(H_\lambda(x)/\tau\big)}{\int p_\phi(x') \exp\big(H_\lambda(x')/\tau\big)\, dx'},$$

where $D(\cdot, \cdot)$ measures sequence-level divergence (e.g. Hamming distance), $\lambda > 0$ is a tunable penalty coefficient, and τ is a temperature parameter that governs how sharply sampling concentrates on high-fitness subspace. As τ decreases, the sampler becomes more concentrated on the top-scoring sequences, whereas higher τ yields broader exploration.

By optimizing a composite functional objective, ProVADA effectively leverages the strength of an *ensemble of expert models*. The implicit generative prior p_ϕ captures

broad, low-level constraints—structural integrity, foldability, solubility—while each supervised fitness predictor $F(x)$ specializes in a particular design objective (e.g., localization, enzymatic activity, binding affinity). By annealing our sampling distribution over the product of the prior and a tempered, aggregated fitness term, ProVADA concentrates on variants that satisfy the foundational constraints and simultaneously score highly under each expert's guidance. This ensemble strategy yields high-confidence, multifunctional candidates that are robust to the idiosyncrasies of any single model, affording practitioners the flexibility to tailor and combine arbitrarily many design objectives.

In the sections that follow, we first describe how to construct the fitness function $F(x)$ for subcellular localization via a classifier that outputs the probability of a sequence residing in the target compartment, and then demonstrate how to adaptively sample from $\pi_{\phi,\tau,\lambda}(x)$ to generate high-fitness variants under our novel sampling procedure.

2.1. Constructing the subcellular fitness score

We formulate subcellular localization as a supervised classification task. Let the training set be $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$, where $x^{(i)} \in \mathcal{X}$ is a protein sequence and $y^{(i)} \in \{0, 1\}$ indicates its corresponding presence in the target compartment. We learn a classifier model $F_\theta(x) \in [0, 1]$ that outputs the estimated probability of localization. The model parameters θ are optimized by minimizing the empirical binary cross-entropy loss

$$\theta^* = \arg\min_\theta -\frac{1}{n}\sum_{i=1}^{n}\Big[y^{(i)}\log F_\theta(x^{(i)}) + (1-y^{(i)})\log(1-F_\theta(x^{(i)}))\Big] + \underbrace{\gamma\|\theta\|_2^2}_{\text{regularization}},$$

where $\gamma > 0$ controls the strength of the ℓ_2 regularization. Once training converges, $F_{\theta^*}(x)$ yields the predicted probability of correct localization, which we treat as a black-box score to guide our sampling. For notational brevity, we henceforth omit the explicit dependence on θ and denote our trained predictor simply as $F(x)$.

2.2. Specifying the notion divergence

To discourage excessive deviation from the wild-type scaffold, we introduce a mismatch penalty term based on the Hamming distance. For any candidate sequence $x \in \mathcal{X}$ and reference x_0, the Hamming distance, defined as

$$d_{\text{Ham}}(x_0, x) = \sum_{\ell=1}^{L} \mathbb{1}\{x_\ell \neq x_{0,\ell}\},$$

counts the number of positions at which x and x_0 differ. Thus, our target distribution becomes

$$H_\lambda(x) = F(x) - \lambda d_{\text{Ham}}(x_0, x),$$
$$\pi_{\phi,\tau,\lambda}(x) \propto p_\phi(x)\exp\big(H_\lambda(x)/\tau\big).$$

By increasing λ, we amplify the Hamming-distance penalty within the annealed score, so that each additional residue mismatch is penalized more heavily. Consequently, sequences that diverge further from the reference incur exponentially larger penalties in their weights, thereby guiding the sampler toward high-fitness variants that remain close to the reference wild-type.

2.3. Mixture-Adaptation Directed Annealing

We now introduce ***M**ixture-**A**daptation **D**irected **A**nnealing* (MADA), a novel sampling framework that efficiently explores high-dimensional, complex composite functional landscapes by integrating mixture-based adaptive proposals, directed local mutation kernels, population-annealed sequential importance sampling, and controlled resampling.

MADA comprises three main components: *selection*, *mutation*, and *stabilization*. At each iteration, MADA maintains a small mixture of promising particle prototypes that generate N offspring through importance resampling with partial rejection control. This population-based approach simultaneously preserves diversity through parallel exploration while effectively concentrating computational effort on the highest-potential regions. Each offspring is then refined by a single Metropolis-Hastings step via fitness-guided local mutation kernels and a gradually decaying temperature schedule to transition smoothly from exploration to exploitation.

These components are integrated into a unified algorithmic procedure to enable sequential and iterative refinement of the entire population. The following subsections provide a detailed description of each sampling step; the complete algorithmic procedure for MADA is provided in Algorithm 1.

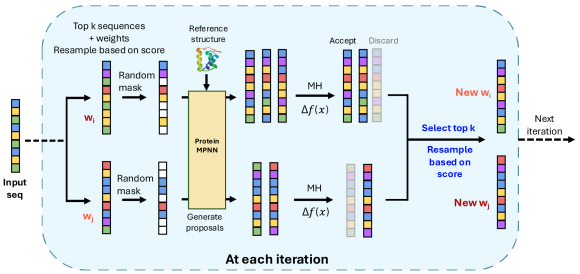

Figure 1. **Mixture-Adaptation Directed Annealing (MADA) algorithm overview.**

2.3.1. INITIALIZATION

Let p_S be the masking probability. We first sample the number of mutated sites

$$M \sim \text{Binomial}(L, p_S), \tag{1}$$

and then choose M distinct positions $S = \{i_1, \ldots, i_M\} \subseteq \{1, \ldots, L\}$ uniformly at random, i.e.,

$$\mathbb{P}(S = \{i_1, \ldots, i_M\}) = \binom{L}{M}^{-1}. \tag{2}$$

By construction, this guarantees that every proposed variant lies within an M-Hamming neighborhood of the reference wild-type x_0. Next, we generate an initial population of N particles by masking x_0 at positions given by S and repeatedly sampling candidates from the implicit generative prior p_ϕ. Concretely, for each $i = 1, \ldots, N$, we draw $x_i^{(0)} \sim p_\phi(\cdot \mid x_{0,S^c})$, so that its marginal law, conditioned on the masking locations, factorizes as $p_\phi(x_i^{(0)}) = p_\phi(x_i^{(0)} \mid x_{0,S^c}) \cdot p_\phi(x_{0,S^c})$. Finally, we initialize the temperature to $\tau_0 = \infty$, so that $1/\tau_0 = 0$ and hence $w_i^{(0)} \propto \exp\big(H_\lambda(x_i^{(0)})/\tau_0\big) = \exp(0) = 1$ reduces to uniform weights at $t = 0$, and thus provides a natural warm start with $\pi_{\phi,\tau_0,\lambda} = p_\phi$.

2.3.2. SELECTION

Having drawn samples from $\pi_{\phi,\tau_t,\lambda}$, we transition to the next tempered distribution $\pi_{\phi,\tau_{t+1},\lambda}$ via importance resampling. To do so, we first compute the annealed importance weights for each particle $x_i^{(t)}$:

$$w_i^{(t+1)} \propto \exp\left(H_\lambda(x_i^{(t)})\left(\frac{1}{\tau_{t+1}} - \frac{1}{\tau_t}\right)\right).$$

where $H_\lambda(x) = F(x) - \lambda\, D(x_0, x)$ is the adjusted fitness score. We then perform a two-stage resampling to eliminate particles with low importance weights, inspired by the bootstrap filter and partial rejection control (11):

- **Prototype selection:** Draw a small set of K prototype particles $\{\zeta_j\}_{j=1}^K$ by sampling with replacement from $\{x_i^{(t)}\}_{i=1}^N$ according to the normalized annealed weights $\{w_i^{(t+1)}\}$.
- **Population reconstruction:** Regenerate a full population of N particles $\{\bar{x}_i^{(t+1)}\}$ by sampling *uniformly* with replacement from the retained K prototypes.

This completes one round of selection and yields $\{\bar{x}_i^{(t+1)}\}_{i=1}^N$, which contains at most K distinct proposals for the subsequent mutation stage. As to be shown in Theorem B.1, this two-stage resampling procedure preserves the statistical unbiasedness of ordinary importance sampling. The down-sample–up-sample procedure above offers two advantages: it concentrates computational effort on the most promising regions and, by reusing a limited set of prototypes, amortizes costly invocations of the generative prior. In the ProteinMPNN setting, reusing the same mask and sequence context across multiple generations substantially reduces expensive model calls and reduces runtime by approximately 75% (see **Figure 7**).

Greedy selection As an alternative to stochastic resampling, one can use a greedy selection strategy: After computing the annealed weights $w_i^{(t+1)}$, deterministically retain only the top K particles with the largest weights. Then rebuild the full population of size N by sampling with replacement from the K elites according to their normalized weights. Although this top-K selection procedure introduces a bit of bias through the permanent elimination of lower-weight particles, we observe that it often results in accelerated convergence to high-fitness regions in practice.

2.3.3. MUTATION

In the mutation stage, each resampled particle $\bar{x}_i^{(t+1)}$ undergoes local perturbation under the tempered target. Specifically, we draw a mask size M and select a subset $S \subseteq \{1, \ldots, L\}$ exactly as in (1)-(2). We then fill the masked positions by sampling $x_i' \sim p_\phi(\cdot \mid \bar{x}_{i,S^c}^{(t+1)})$, conditioned on the unmasked residues of $\bar{x}_i^{(t+1)}$. We denote this local-move proposal kernel by $x' \sim q_S(\cdot \mid x)$. This mask-then-fill procedure implements a systematic-scan Gibbs mutation kernel. Because some local moves may reduce fitness, each proposed x_i' is forwarded to the stabilization stage, where it is accepted or rejected according to the Metropolis-Hastings criterion.

2.3.4. STABILIZATION

After generating each mutated proposal $\{x_i'\}_{i=1}^N$, we apply a Metropolis-Hastings (MH) accept-reject step to stabilize and direct the local exploration according to fitness. Let $\bar{x}_i^{(t+1)}$ denote the pre-mutation particle. We compute the change in adjusted fitness

$$\Delta H_\lambda = [F(x_i') - \lambda\, D(x_0, x_i')] - [F(\bar{x}_i^{(t+1)}) - \lambda\, D(x_0, \bar{x}_i^{(t+1)})],$$

which biases acceptance toward moves that increase the fitness score or incur a lower divergence penalty. We then draw $U \sim \text{Unif}[0, 1]$ and accept the proposal x_i' with probability

$$\min\left\{1, \exp\left(\frac{\Delta H_\lambda}{\tau_{t+1}}\right)\right\}.$$

If accepted, set $x_i^{(t+1)} \leftarrow x_i'$; otherwise retain $x_i^{(t+1)} \leftarrow \bar{x}_i^{(t+1)}$.

Algorithm 1 Mixture-Adaptation Directed Annealing

Require: Reference sequence x_0, fitness $F(x)$, divergence penalty λ,
temperature schedule $\{\tau_1, \dots, \tau_T\}$, population size N, subsample size K

return Proposal sequences $\{x_i^{(T)}\}_{i=1}^N \sim \pi_{\phi,\tau_T,\lambda}(x)$
Initialization:
Sample mask S according to Eq. (2)
Draw N replicas $\{x_i^{(0)}\}_{i=1}^N \sim p_\phi(\cdot \mid x_{0,S^c})$
Set $\tau_0 \leftarrow \infty$
for $t = 0$ **to** $T-1$ **do**
  `/* Annealed importance Weighting */`
  for $i = 1$ **to** N **do**
    $H(x_i^{(t)}) \leftarrow F(x_i^{(t)}) - \lambda\, D(x_0, x_i^{(t)})$
    $\log \tilde{w}_i^{(t+1)} \leftarrow H(x_i^{(t)}) \left(\frac{1}{\tau_{t+1}} - \frac{1}{\tau_t}\right)$
  end for
  Compute normalized weights:

$$w_i^{(t+1)} \leftarrow \frac{\exp\left(\log \tilde{w}_i^{(t+1)}\right)}{\sum_{j=1}^N \exp\left(\log \tilde{w}_j^{(t+1)}\right)}.$$

  `/* Selection */`
  Draw K particles with replacement:
  $\{\zeta_j\}_{j=1}^K \sim \text{Categorical}(w^{(t+1)})$
  Form $\{\bar{x}_i^{(t+1)}\}_{i=1}^N$ by uniform resample from $\{\zeta_j\}$
  `/* Rejuvenation-Mutation */`
  for $i = 1$ **to** N **do**
    Sample mask S according to Eq. (2)
    Propose $x'_S \sim p_\phi(\cdot \mid \bar{x}_{i,S^c}^{(t+1)})$
    Set $x'_{S^c} \leftarrow \bar{x}_{i,S^c}^{(t+1)}$ {Keep unmasked positions unchanged}
    $H(x') \leftarrow F(x') - \lambda\, D(x_0, x')$
    $H(\bar{x}_i^{(t+1)}) \leftarrow F(\bar{x}_i^{(t+1)}) - \lambda\, D(x_0, \bar{x}_i^{(t+1)})$
    `/* Stabilization */`
    Draw $U \sim \text{Unif}[0,1]$
    Compute acceptance ratio:

$$a \leftarrow \min\left\{1, \exp\left(\frac{H(x') - H(\bar{x}_i^{(t+1)})}{\tau_{t+1}}\right)\right\}$$

    if $U \le a$ **then**
      $x_i^{(t+1)} \leftarrow x'$
    else
      $x_i^{(t+1)} \leftarrow \bar{x}_i^{(t+1)}$
    end if
  end for
end for
return $\{x_i^{(T)}\}_{i=1}^N$

This MH correction step stabilizes the sampler on the annealed target $\pi_{\phi,\tau_{t+1},\lambda}$. Theorem B.2 shows that the proposed procedure satisfies detailed balance.

2.3.5. Annealing Schedule

We employ a power-law cooling (8) schedule to gradually reduce both the masking fraction and the temperature τ_t. At step t of T total iterations, define the normalized time $s = \frac{t}{T-1}$. We then update

$$\tau_t = \tau_{\max}(1-s)^\alpha + \tau_{\min} \tag{3}$$

$$p_{S,t} = p_{S,\min} + (p_{S,\max} - p_{S,\min})(1-s)^\alpha, \tag{4}$$

where $\alpha > 0$ controls the annealing rate, $\tau_{\max}$ and $p_{S,\max}$ denote the initial temperature and masking fraction, respectively, and $\tau_{\min}$ and $p_{S,\min}$ their terminal values. This power-law decay smoothly transitions the sampler from broad exploration (high-temperature, heavy masking) to focused exploitation (low-temperature, light masking). Empirically, we observe that this approach accelerates convergence while improving the final solution quality (see Appendix 8 for decay curves under different α).

3. Application of ProVADA to *In Silico* Engineering of Cytosolic Renin Variants

3.1. Motivations for a practical example

In this section, we conduct an *in silico* experiment with our pipeline to address a practical design challenge: engineering a cytosolic variant of human peptidase renin. Renin is a secreted protease with high specificity to cleave a defined peptide sequence on its natural substrate angiotensinogen (26). Owing to this high specificity, there is considerable interest in repurposing renin as a general protease tool for precise control of protein activity via targeted cleavage (15). However, because renin normally functions in the extracellular environment, our internal data suggest that cytosolic expression leads to misfolded and non-functional renin, which impedes its development as a generalized tool. The described renin engineering campaign is an appropriate test case for our pipeline because i) there are no close homologs of renin that functions in the cytoplasm, i.e., we have little or no evolutionary information to leverage; ii) we start from a zero-activity scaffold, making hybrid directed evolution approaches inefficient; iii) assays for cytosolic renin activity must be performed in live mammalian cells, which limit the throughput and are unaffordable to carry out at large scale, thus favoring test-time steering methodologies.

We begin by constructing a reliable fitness oracle for cytosolic functionality. Motivated by the well-established links between subcellular localization and certain sequence characteristics, such as N-linked glycosylation sites and disulfide bonds (17; 4), we train a binary classifier on pro-

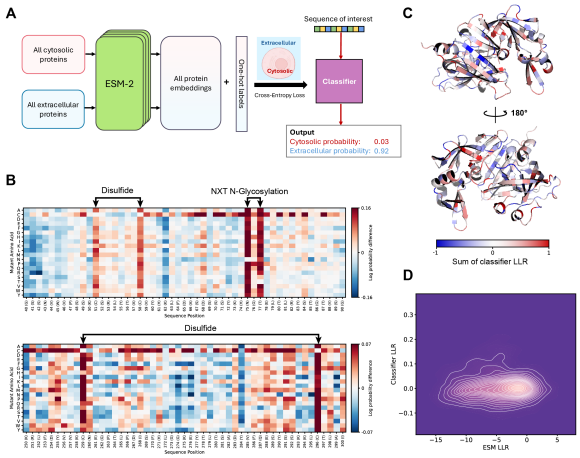

Figure 2. **Subcellular classifier: construction and performance.** (**A**) Workflow for training and inference. (**B**) Heatmap of cytosolic log-likelihood ratio (LLR) for all single-point mutations in wild-type renin, shown for residues 40–90 and 250–300. Identified hotspots—positions where non-WT substitutions markedly increase cytosolic probability—correspond to known secretory features: disulfide bonds C51–C58, C259–C296, and the N75–T77 N-glycosylation motif. (**C**) Mapping of per-position cytosolic LLR onto the ESMFold-predicted renin catalytic domain. (**D**) Comparison of classifier and ESM LLR landscapes.

tein language model embeddings to predict the probability of cytosolic localization. To ensure our model focuses on those intrinsic protein properties that determine localization-dependent viability, we curate our dataset by removing low-complexity signals, including signal peptides and transmembrane domains. We apply MADA with our trained classifier as the fitness oracle to the human renin catalytic domain; this approach respects structural and evolutionary constraints via our generative prior while driving the search toward variants with high predicted cytosolic compatibility and preserved enzymatic fold.

3.2. Classifier construction, training and evaluation

Classifier construction We assemble a curated set of vertebrate cytosolic and extracellular proteins from UniProt Swiss-Prot, remove low-complexity sequences (signal peptides and transmembrane domains), and reduce redundancy by clustering at 30 % sequence identity with MMSeqs2. We then extracted mean-pooled ESM2-650M embeddings and trained a logistic regression classifier. See **Appendix A** for full details.

Benchmark against existing subcellular location predictors Several existing subcellular localization predictors leverage protein language model representations. To highlight the difference, we compare our classifier to three established subcellular localization predictors: DeepLoc2.0-fast (ESM-1b), LocPro (ESM2 ensemble), and MULocDeep (bidirectional LSTM) (29; 33; 21). We evaluate performance on two benchmark datasets: (1) curated Swiss-Prot sequences and (2) extracellular proteins from the Human Protein Atlas Secretome (30). **Table 1** summarizes the results. For curated Swiss-Prot, we report the weighted average AUROC for cytosolic versus extracellular predictions. A detailed breakdown by label and metric appears in **Table 2**. Since the HPA-Secretome dataset includes only extracellular proteins, we measure performance by extracellular classification accuracy. As shown in **Table 1**, our classifier outperforms existing predictors on the curated Swiss-Prot dataset, as expected given that those methods focus primarily on signal sequences (29). **Table 2** further demonstrates that the greatest improvement of our classifier performance comes from the high cytosolic label precision and extracellular label recall. Finally, our classifier also surpasses existing models on the orthogonal HPA-Secretome dataset of intact secreted proteins, indicating that it captures intrinsic sequence determinants of native localization rather than depending on low-complexity signal peptides.

Classifier probabilities identify sequence features that align with domain knowledge We assess whether the probabilities output by the classifier reflect a sequence's propensity to localize to the target compartment. To this end, we perform an *in silico* deep mutation scan (DMS) on the 340-residue catalytic domain of human renin. We then compare each mutant's predicted probability to that of the wild-type sequence. Analysis of these probability shifts reveals "hot-spot" positions where substitutions of the wild-type sequence are strongly favored. Many of these positions coincide with known extracellular signatures, including the documented two NXT N-glycosylation sites and three disulfide bonds. **Figure 2B** exemplifies three such examples by the heatmap of the cytosolic probability shifts to WT renin. The full DMS heatmap can be found in **Figure 6**. By further mapping these "hot-spots" to renin structure via summing those probability shifts by position, **Figure 2C** shows that the distribution of these "hot-spots" is dispersed throughout the structure and is not enriched in any specific regions or surface areas. Moreover, the KDE plot in **Figure 2D** reveals a very weak correlation between classifier probability shifts and ESM2 logit likelihoods across all single-point mutations. This suggests that, despite being trained on ESM2 embeddings, the classifier's localization landscape is largely orthogonal to ESM2's intrinsic fitness landscape. These observations underscore the difficulty of classifier-guided renin design and motivate our MADA algorithm, which incorporates both structural information and supports broad, global sequence exploration.

Table 1. Performance comparison of our localization classifier against three baselines on two datasets. Boldface indicates the best result.

DATA SET	OURS	DEEPLOC2.0	LOCPRO	MULOCDEEP
CURATED SWISS-PROT (AUROC ↑)	**97.18**	85.70	90.83	51.01
HPA-SECRETOME (ACC ↑)	**82.31**	74.28	51.20	47.55

3.3. Empirical Evaluations of MADA for Cytosolic Renin Engineering

In this subsection, we present *in silico* results obtained from using MADA to design cytosolic variants of the human renin catalytic domain. The wild-type sequence exhibits a low predicted cytosolic localization probability of 0.035. We compare ProVADA against two rejection-sampling baselines, and a generative baseline from the built-in guided generation approach in *ESM3* (16). The naïve rejection sampler randomly masks up to 50% of positions and replaces each with a uniformly sampled amino acid. The ProteinMPNN-based rejection sampler uses the same masking scheme (up to 100% of sites) but refills masked positions using ProteinMPNN's generative prior. All ProteinMPNN-based masked generations in this manuscript are generated with the temperature set at 0.5 with cysteine- and non-canonical residue-free designs. For ESM3-guided generation, we use the ESM3-open model (14), fix its predicted structural and functional tokens for the renin catalytic domain, and steer decoding with cytosolic probabilities from our classifier.

3.3.1. RUNTIME COST VS. NUMBER OF PROTEINMPNN FILL CALLS

We measure wall-clock time under two regimes to illustrate the efficiency gains of our down-sample–up-sample strategy: i) invoking ProteinMPNN separately for each of 10 masked sequences (10 calls), and ii) a single invocation that returns 10 filled sequences in one batch. We repeat each experiment 10 times and report the average runtimes in **Figure 7**, where the results show that our down-sample–up-sample strategy improved runtime efficiency by approximately 4-fold.

3.3.2. COMPARISON OF FITNESS SCORES FOR GENERATED VARIANTS

We benchmark MADA against the three baselines. We collect 1500 sequences from the final iteration of 50 independent MADA runs (30 iterations each, greedily retaining the top 20% per round). The initial temperature is set to $\tau_1 = 2.0$, and we employ a power-law cooling schedule with $\alpha = 3.0$ (see **Figure 8**). Both the naïve and ProteinMPNN rejection samplers yield 1,500 sequences each, while the ESM3 generative baseline is limited to 200 sequences for computational traceability.

In **Figure 3A**, we compare cytosolic probability distributions across methods. The fitness distribution of MADA variants exhibits a clear mode around 0.7, with most variants exceeding the 0.5 label-flipping threshold, while both rejection baselines' outputs remain near the wild-type probability. MADA achieves a 9.5-fold increase in predicted fitness over these baselines. On the other hand, ESM3-generated variants center near 0.4, with only a few surpassing the threshold. **Figure 3B** shows that MADA variants carry on average 47% mutations, whereas ESM3 variants exceed 50% yet retain higher cosine similarity to wild-type embeddings. The MADA mutation profile thus occupies a viable yet diverged range from the wild-type renin; such distinction is appropriate and necessary given the absence of cytosolic renin homologs.

To assess convergence within a single MADA run, we execute 100 iterations with a population of 30 sequences per round, greedily retaining the top 20% at each step. Starting from a masking fraction of 0.2, temperature $\tau_1 = 2.0$, and Hamming-penalty $\lambda = 0.1$, we track the best fitness score and mutation fraction over sampling (**Figure 3C**). Cytosolic probability plateaus by iteration 40, while mutation fraction peaks at 0.5 by iteration 20 before declining under the divergence penalty. With our annealing schedule, 30 iterations strike a practical balance, reaching near convergence with manageable computation.

We further characterize the top variants from 50 independent MADA runs (30 iterations each). As shown in **Figure 3D**, counting extracellular signatures in the highest-fitness sequence at each iteration reveals a rapid drop to near zero within 5 iterations. We also track the sequence logo at the N75–T77 N-glycosylation site, which demonstrates progressive loss of the motif pattern over successive iterations. In **Figure 3E**, sequence logos for four extracellular motifs in the final MADA variants illustrate their complete elimination—most notably, both NXT glycosylation motifs are efficiently removed. Finally, **Figure 3F** shows convergence across 50 runs for **i)** cytosolic probability, **ii)** adjusted fitness, **iii)** MPNN score, and **iv)** mutation fraction.

3.3.3. SEQUENCE ANALYSIS OF MADA SAMPLED VARIANTS

We analyze the 50 highest-fitness MADA variants' sequence features, including keyword annotations, homology relationships, and per-position mutation frequencies.

First, we run InterProScan on the 50 MADA variants alongside 50 ESM3-generated sequences (22). **Figure 4A** shows that both sets preserve the superfamily and domain-level keywords found in WT renin; all ESM3 sequences and 48

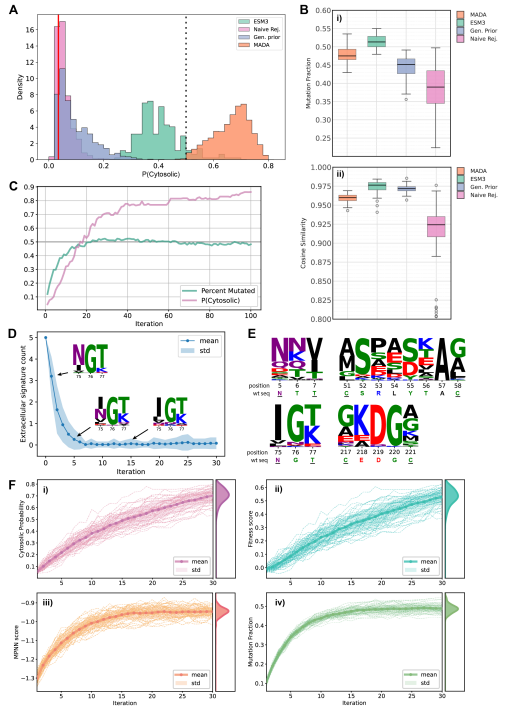

Figure 3. **MADA sampling performance.** **(A)** Cytosolic probability distributions from MADA (1500 samples), ESM3 (200), naïve rejection (1500), and ProteinMPNN prior (1500); red line = WT, dotted = 0.5 threshold. **(B)** Mutation fraction and ESM2-embedding cosine similarity to WT renin. **(C)** Example MADA trajectory (100 iterations, 30 chains): cytosolic probability and mutation fraction. **(D)** Decline of extracellular signatures over iterations; inset Logos of N75–T77 motif at iterations 1, 5, 15. **(E)** LogoPlots of four extracellular signatures in final top variants. **(F)** Trajectories of cytosolic probability, adjusted fitness, MPNN score, and mutation fraction across 50 runs; right: KDE of final values (bold = mean).

of the 50 MADA variants retain the conserved aspartyl protease active site. Neither group preserves the renin-family annotation, which is unsurprising given the approximately 50% mutation rate. Both also introduce novel keywords absent from the wild-type sequence. These findings indicate that MADA produces variants that retain domain-level functionality related to aspartic proteases.

Next, we assess MADA output diversity via pairwise Hamming distances and cosine similarities in the ESM2 embedding space. **Figure 4B** shows that the generated variants exhibit an average pairwise Hamming distance of approximately 120 (30% mutations), which is significantly lower than their Hamming distances to WT renin (Hamming distance around 170, 45% mutations). In **Figure 4C**, the distributions of pairwise cosine similarity in ESM2 embedding space from MADA variants and natural renin homologs (from BlastP) overlap closely, indicating comparable sequence-space diversity. **Figure 9** in the Appendix confirms that MADA variants and homologs form separate

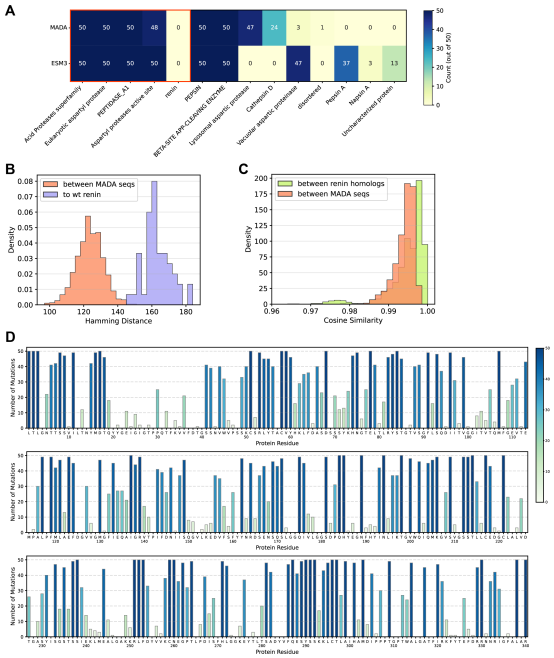

Figure 4. **Sequence analysis of MADA-sampled renin variants.** **(A)** InterProScan keyword heatmap for 50 MADA vs. ESM3 variants (WT keywords boxed). **(B)** Pairwise Hamming distance distributions: MADA–MADA (orange) and MADA–WT (purple). **(C)** ESM2 embedding cosine similarity: MADA–MADA (orange) vs. natural renin homologs (green). **(D)** Per-position mutation frequency across 50 MADA variants.

clusters. These results indicate that MADA produces a set of distant yet plausible variants, introducing sufficient diversity to enable cytosolic localization without compromising core structural and functional integrity.

Lastly, **Figure 4D** shows per-position mutation frequencies for the MADA renin variants. Both catalytic Aspartic residues (D38 and D226) remain fully conserved, as do their flanking motifs. In contrast, the active-site flap (T80–G90) is highly variable, with key substrate-binding residues such as Y83 and S84 nearly completely mutated (32). The results above suggest that MADA variants preserve Aspartic protease activity but will likely lose renin-specific substrate selectivity. To maintain realistic renin specificity, it will be necessary to hard fix substrate-contacting residues.

3.3.4. Structural Analysis of MADA Sampled Variants

We evaluate the structural viability of the sampled variants by predicting folded structures with ESMFold and comparing their alignment to the native renin catalytic domain. We analyze the same 50 MADA and 50 ESM3 sequences shown in **Figure 4**. For each rejection-sampling baseline, we select the top 50 variants by fitness from 1500 generated sequences.

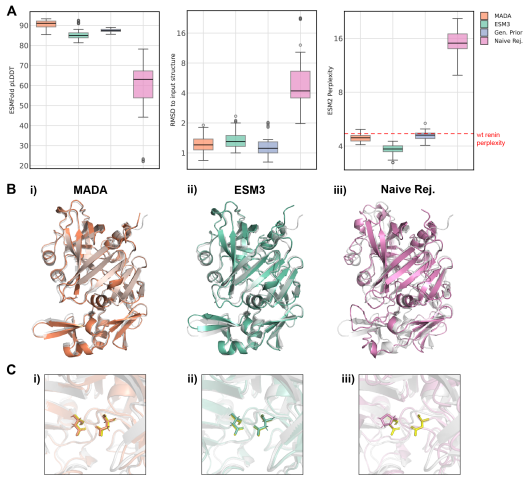

Figure 5. **Structural analysis of MADA-sampled renin variants. (A)** ESMFold-predicted pLDDT, RMSD to WT, and ESM2 pseudo-perplexity for the top 50 variants (red dashed = WT perplexity). **(B)** Representative alignments of MADA, ESM3, and naïve variants to the WT structure. **(C)** Close-up of catalytic Asp side-chain conformations (WT in yellow).

Figure 5A (left and middle) presents the overall structural confidence (pLDDT) and structural deviation from the WT renin (RMSD), while the right panel reports ESM2 pseudo-perplexity. ProteinMPNN-conditioned variants preserve both structural integrity and evolutionary plausibility, in contrast to random mutations that disrupt both. ESM3-generated variants also maintain fold quality but exhibit lower ESM2 perplexity, reflecting a preference for evolutionary likelihood that can be unnecessary and even undesirable when targeting fitness objectives unrelated to evolutionary constraints.

To highlight the structural differences, **Figure 5B** displays representative alignments to the WT renin fold of variants from (i) MADA, (ii) ESM3, and (iii) naïve rejection. Both MADA- and ESM3-generated variants align closely, whereas the naïve rejection sampler produced a fully misfolded subdomain at the bottom of the structure. **Figure 5C** zooms in on the catalytic dyad side chains: MADA and ESM3 variants reproduce the WT conformation, while the rejection variant loses the expected geometry. These results demonstrate that MADA efficiently enhances fitness without compromising structural integrity or evolutionary plausibility.

4. Conclusion

In this work, we present ProVADA, a test-time steering, ensemble-guided framework powered by our novel Mixture-Adaptation Directed Annealing (MADA) sampler. ProVADA efficiently generates protein variants tailored to desired fitness objectives while maintaining structural and evolutionary integrity. Leveraging our high-accuracy subcellular localization classifier, we demonstrate ProVADA's effectiveness through *in silico* engineering of cytosolic renin variants. Notably, MADA achieves a remarkable 9.5-fold increase in sampling efficiency compared to conventional masked-fill rejection sampling. Our experiments demonstrate that ProVADA delivers diverse renin variants with substantially improved predicted cytosolic localization while preserving both structural stability and evolutionary plausibility.

Beyond engineering subcellular variants, ProVADA demonstrates broad applicability across diverse protein design challenges. By training predictors for specific compartments such as endosomes or mitochondria, ProVADA can optimize protein stability for intracellular therapeutics requiring endosomal escape (7) or enhance the efficiency of mitochondrial base editors (12). Furthermore, with recent advances in immunogenicity prediction (6), ProVADA could enable the guided generation of de-immunized variants for therapeutic protein humanization (31). Collectively, ProVADA provides a versatile, structure-aware framework for directing protein design across varied functional landscapes, offering significant potential for protein variant engineering applications.

Several directions warrant consideration for future work. While our approach omits explicit substrate-interaction guidance, potentially compromising substrate specificity, this can be readily addressed by fixing critical contact residues or imposing additional structural constraints. Furthermore, despite ProVADA's strong *in silico* performance, experimental validation remains essential to confirm the real-world effectiveness of engineered variants. These directions represent important avenues for future work to fully realize ProVADA's therapeutic potential.

Acknowledgments

The authors would like to thank Brian Trippe, Ben Viggiano, and anonymous reviewers for helpful discussions and insightful remarks. W.S.L. gratefully acknowledges support from the Stanford Data Science Fellowship and NIH grant GM 134483. X.Z. is supported by the Stanford Interdisciplinary Graduate Fellowship affiliated with ChEM-H. X.J.G is supported by the Stanford Bio-X Interdisciplinary Initiatives seed grant program (R12-8, to X.J.G.).

References

[1] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C.-C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Con-

greve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, and J. M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, May 2024.

[2] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023.

[3] A. Bateman, M.-J. Martin, S. Orchard, M. Magrane, A. Adesina, S. Ahmad, E. H. Bowler-Barnett, H. Bye-A-Jee, D. Carpentier, P. Denny, J. Fan, P. Garmiri, L. J. d. C. Gonzales, A. Hussein, A. Ignatchenko, G. Insana, R. Ishtiaq, V. Joshi, D. Jyothi, S. Kandasaamy, A. Lock, A. Luciani, J. Luo, Y. Lussi, J. S. M. Marin, P. Raposo, D. L. Rice, R. Santos, E. Speretta, J. Stephenson, P. Totoo, N. Tyagi, N. Urakova, P. Vasudev, K. Warner, S. Wijerathne, C. W.-H. Yu, R. Zaru, A. J. Bridge, L. Aimo, G. Argoud-Puy, A. H. Auchincloss, K. B. Axelsen, P. Bansal, D. Baratin, T. M. Batista Neto, M.-C. Blatter, J. T. Bolleman, E. Boutet, L. Breuza, B. C. Gil, C. Casals-Casas, K. C. Echioukh, E. Coudert, B. Cuche, E. de Castro, A. Estreicher, M. L. Famiglietti, M. Feuermann, E. Gasteiger, P. Gaudet, S. Gehant, V. Gerritsen, A. Gos, N. Gruaz, C. Hulo, N. Hyka-Nouspikel, F. Jungo, A. Kerhornou, P. L. Mercier, D. Lieberherr, P. Masson, A. Morgat, S. Paesano, I. Pedruzzi, S. Pilbout, L. Pourcel, S. Poux, M. Pozzato, M. Pruess, N. Redaschi, C. Rivoire, C. J. A. Sigrist, K. Sonesson, S. Sundaram, A. Sveshnikova, C. H. Wu, C. N. Arighi, C. Chen, Y. Chen, H. Huang, K. Laiho, M. Lehvaslaiho, P. McGarvey, D. A. Natale, K. Ross, C. R. Vinayaka, Y. Wang, and J. Zhang. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, Nov. 2024.

[4] T. J. Bechtel and E. Weerapana. From structure to redox: The diverse functional roles of disulfides and implications in disease. *PROTEOMICS*, 17(6), Mar. 2017.

[5] F. Cesaratto, A. López-Requena, O. R. Burrone, and G. Petris. Engineered tobacco etch virus (tev) protease active in the secretory pathway of mammalian cells. *Journal of Biotechnology*, 212:159–166, Oct. 2015.

[6] B. Chen, M. S. Khodadoust, N. Olsson, L. E. Wagar, E. Fast, C. L. Liu, Y. Muftuoglu, B. J. Sworder, M. Diehn, R. Levy, M. M. Davis, J. E. Elias, R. B. Altman, and A. A. Alizadeh. Predicting hla class ii antigen presentation through integrated deep learning. *Nature Biotechnology*, 37(11):1332–1343, Oct. 2019.

[7] P. Chen and H. Cabral. Enhancing targeted drug delivery through cell-specific endosomal escape. *ChemMedChem*, July 2024.

[8] M. C. Choi. An improved variant of simulated annealing that converges under fast cooling. *arXiv preprint arXiv:1901.10269*, 2019.

[9] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[10] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, Oct. 2022.

[11] A. Doucet, N. De Freitas, N. J. Gordon, et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001.

[12] Editorial. Powering new therapeutics with precision mitochondrial editing. *Nature Biotechnology*, 43(6):831–832, June 2025.

[13] P. Emami, A. Perreault, J. Law, D. Biagioni, and P. St. John. Plug and play directed evolution of proteins with gradient-based discrete mcmc. *Machine Learning: Science and Technology*, 4(2):025014, Apr. 2023.

[14] EvolutionaryScale Team. evolutionaryscale/esm, 2024.

[15] X. J. Gao, L. S. Chong, M. S. Kim, and M. B. Elowitz. Programmable protein circuits in living cells. *Science*, 361(6408):1252–1258, Sept. 2018.

[16] T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, R. Badkundri, I. Shafkat, J. Gong, A. Derry, R. S. Molina, N. Thomas, Y. A. Khan, C. Mishra, C. Kim, L. J. Bartie, M. Nemeth, P. D. Hsu, T. Sercu, S. Candido, and A. Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025.

[17] A. Helenius and M. Aebi. Roles of n-linked glycans in the endoplasmic reticulum. *Annual Review of Biochemistry*, 73(1):1019–1049, June 2004.

[18] B. L. Hie, V. R. Shanker, D. Xu, T. U. J. Bruun, P. A. Weidenbacher, S. Tang, W. Wu, J. E. Pak, and P. S. Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, Feb 2024.

[19] J. Ho and T. Salimans. Classifier-free diffusion guidance, 2022.

[20] K. Jiang, Z. Yan, M. D. Bernardo, S. R. Sgrizzi, L. Villiger, A. Kayabolen, B. J. Kim, J. K. Carscadden, M. Hiraizumi, H. Nishimasu, J. S. Gootenberg, and O. O. Abudayyeh. Rapid in silico directed evolution by a protein language model with evolvepro. *Science*, 387(6732):eadr6006, 2025.

[21] Y. Jiang, D. Wang, Y. Yao, H. Eubel, P. Künzler, I. M. Møller, and D. Xu. Mulocdeep: A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation. *Computational and Structural Biotechnology Journal*, 19:4825–4839, 2021.

[22] P. Jones, D. Binns, H.-Y. Chang, M. Fraser, W. Li, C. McAnulla, H. McWilliam, J. Maslen, A. Mitchell, G. Nuka, S. Pesseat, A. F. Quinn, A. Sangrador-Vegas, M. Scheremetjew, S.-Y. Yong, R. Lopez, and S. Hunter. Interproscan 5: genome-scale protein function classification. *Bioinformatics (Oxford, England)*, 30(9):1236—1240, May 2014.

[23] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, July 2021.

[24] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[25] A. Madani, B. Krause, E. R. Greene, S. Subramanian, B. P. Mohr, J. M. Holton, J. L. Olmos, C. Xiong, Z. Z. Sun, R. Socher, J. S. Fraser, and N. Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, Jan. 2023.

[26] P. B. Persson. Renin: origin, secretion and synthesis. *The Journal of Physiology*, 552(3):667–671, Nov. 2003.

[27] M. Steinegger and J. Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, Oct. 2017.

[28] K. H. Sumida, R. Núñez-Franco, I. Kalvet, S. J. Pellock, B. I. M. Wicky, L. F. Milles, J. Dauparas, J. Wang, Y. Kipnis, N. Jameson, A. Kang, J. De La Cruz, B. Sankaran, A. K. Bera, G. Jiménez-Osés, and D. Baker. Improving protein expression, stability, and function with proteinmpnn. *Journal of the American Chemical Society*, 146(3):2054–2061, Jan. 2024.

[29] V. Thumuluri, J. J. Almagro Armenteros, A. R. Johansen, H. Nielsen, and O. Winther. Deeploc 2.0: multi-label subcellular localization prediction using protein language models. *Nucleic Acids Research*, 50(W1):W228–W234, Apr. 2022.

[30] M. Uhlén, M. J. Karlsson, A. Hober, A.-S. Svensson, J. Scheffel, D. Kotol, W. Zhong, A. Tebani, L. Strandberg, F. Edfors, E. Sjöstedt, J. Mulder, A. Mardinoglu, A. Berling, S. Ekblad, M. Dannemeyer, S. Kanje, J. Rockberg, M. Lundqvist, M. Malm, A.-L. Volk, P. Nilsson, A. Månberg, T. Dodig-Crnkovic, E. Pin, M. Zwahlen, P. Oksvold, K. von Feilitzen, R. S. Häussler, M.-G. Hong, C. Lindskog, F. Ponten, B. Katona, J. Vuu, E. Lindström, J. Nielsen, J. Robinson, B. Ayoglu, D. Mahdessian, D. Sullivan, P. Thul, F. Danielsson, C. Stadler, E. Lundberg, G. Bergström, A. Gummesson, B. G. Voldborg, H. Tegel, S. Hober, B. Forsström, J. M. Schwenk, L. Fagerberg, and A. Sivertsson. The human secretome. *Science Signaling*, 12(609), Nov. 2019.

[31] E. Wolfsberg, J.-S. Paul, J. Tycko, B. Chen, M. C. Bassik, L. Bintu, A. A. Alizadeh, and X. J. Gao. Machine-guided dual-objective protein engineering for deimmunization and therapeutic functions. *Cell Systems*, page 101299, June 2025.

[32] Y. Yan, A. Zhou, R. W. Carrell, and R. J. Read. Structural basis for the specificity of renin-mediated angiotensinogen cleavage. *Journal of Biological Chemistry*, 294(7):2353–2364, Feb. 2019.

[33] Y. ZHANG, L. ZHENG, N. YOU, W. HU, W. JIANG, M. LU, H. XU, H. DAI, T. FU, and Y. ZHOU. Locpro: a deep learning-based prediction of protein subcellular localization for promoting multi-directional pharmaceutical research. *Journal of Pharmaceutical Analysis*, page 101255, Mar. 2025.

A. Data preprocessing and classifier construction

Dataset acquisition We query the UniProt Swiss-Prot database to curate a dataset of cytosolic and extracellular proteins from all vertebrate species (3). To ensure data reliability, we include only entries with experimental evidence and exclude proteins localized to organelles including lysosome, endosome, peroxisome, and mitochondria.

Sequence truncation and filtering In order to isolate intrinsic sequence features that govern fitness in different subcellular compartments, we remove low-complexity localization signals (signal peptides and transmembrane domains) that are strongly correlated with localization but unrelated to fitness (29). We retain only mature peptide regions and discard any annotated signal sequences and propeptide regions. In cases of multiple annotated domains, only shorter, individual domains (50–1000 residues) are retained. For transmembrane proteins, we extract only annotated extracellular domains within the same size range. To ensure balanced representation of protein families, we cluster all resulting sequences using MMSeqs2 with a 30% sequence similarity threshold (27). We retain one representative per cluster, resulting in a reduced dataset (40% of the original size). Finally, the dataset is stratified and split into training, validation, and test sets in a 70:20:10 ratio.

Embedding generation and training We select the ESM2-650M-UR50D model for protein sequence embedding due to its strong performance in various downstream tasks and its balanced trade-off between accuracy and computational efficiency (24). If not specifically stated, all ESM2 models in this article refer to this specific model mentioned above. We embed each protein sequence with the model, extracting per-residue representations from the final (33^{rd}) transformer layer. We train a logistic regression classifier on the 1280-dimensional mean-pooled sequence embeddings across the sequence length. A graphical summary of classifier construction can be found in **Figure 2A**.

B. Theoretical Analysis

Theorem B.1. *Let $\{x_i\}_{i=1}^N$ be a given collection of particles with corresponding normalized importance weights $\{w_i\}_{i=1}^N$, where $\sum_{i=1}^N w_i = 1$ and $w_i \geq 0$. Consider the two-stage resampling procedure described in Section 2.3.2, which produces samples $\{\bar{x}_\ell\}_{\ell=1}^N$. For each $\ell \in \{1, \dots, N\}$, it holds that $\mathbb{P}(\bar{x}_\ell = x_i) = w_i$, and consequently, for any bounded measurable function h, the estimator $\frac{1}{N}\sum_{\ell=1}^N h(\bar{x}_\ell)$ is unbiased for the weighted expectation $\sum_{i=1}^N w_i\, h(x_i)$.*

Theorem B.2. *Consider the Markov kernel K defined as:*

$$K(x \to x') = \begin{cases} \sum_S P(S) q_S(x' \mid x) a(x, x') & \text{if } x \neq x' \\ 1 - \sum_{x'' \neq x} K(x \to x'') & \text{if } x = x' \end{cases}$$

where $q_S(x' \mid x) = \mathbb{1}_{\{x'_{S^c} = x_{S^c}\}} p_\phi(x'_S \mid x_{S^c})$ and $a(x, x') = \min\left\{1, \frac{\pi_{\phi,\tau_t,\lambda}(x') q_S(x|x')}{\pi_{\phi,\tau_t,\lambda}(x) q_S(x'|x)}\right\}$. This kernel satisfies detailed balance with respect to $\pi_{\phi,\tau_t,\lambda}$, i.e., $\pi_{\phi,\tau_t,\lambda}(x) K(x \to x') = \pi_{\phi,\tau_t,\lambda}(x') K(x' \to x)$ for all x, x'. Moreover, when $\pi_{\phi,\tau_t,\lambda}(x) \propto p_\phi(x) \exp(H_\lambda(x)/\tau_t)$, the acceptance ratio simplifies to $a(x, x') = \min\left\{1, \frac{\exp(H_\lambda(x')/\tau_t)}{\exp(H_\lambda(x)/\tau_t)}\right\}$.

B.1. Proof of Theorem B.1

Proof. Let's denote by $\mathcal{Z} = \{\zeta_1, \zeta_2, \dots, \zeta_K\}$ the set of selected prototypes. For any $\ell \in \{1, 2, \dots, N\}$, we have

$$\mathbb{P}(\bar{x}_\ell = x_i) = \sum_{\mathcal{Z}} \mathbb{P}(\bar{x}_\ell = x_i \mid \mathcal{Z}) \mathbb{P}(\mathcal{Z}).$$

Since $\bar{x}_\ell$ is sampled uniformly from $\mathcal{Z}$, we have

$$\mathbb{P}(\bar{x}_\ell = x_i \mid \mathcal{Z}) = \frac{n_i(\mathcal{Z})}{K},$$

where $n_i(\mathcal{Z}) \sim \text{Binomial}(K, w_i)$ is the number of times x_i appears in $\mathcal{Z}$.

Now, since each ζ_j is drawn independently with replacement according to weights $\{w_i\}$, the expected number of times x_i appears in $\mathcal{Z}$ is $K \cdot w_i$. Therefore,

$$\mathbb{E}[n_i(\mathcal{Z})] = K \cdot w_i.$$

Combining these results, we have

$$\begin{aligned}\mathbb{P}(\bar{x}_\ell = x_i) &= \mathbb{E}_{\mathcal{Z}}[\mathbb{P}(\bar{x}_\ell = x_i \mid \mathcal{Z})] \\ &= \mathbb{E}_{\mathcal{Z}}\left[\frac{n_i(\mathcal{Z})}{K}\right] \\ &= \frac{1}{K}\mathbb{E}_{\mathcal{Z}}[n_i(\mathcal{Z})] \\ &= \frac{1}{K} \cdot K \cdot w_i \\ &= w_i.\end{aligned}$$

Hence, $\mathbb{E}[h(\bar{x}_\ell)] = \sum_i w_i h(x_i)$, and averaging over ℓ yields the unbiasedness of the overall estimator. □

B.2. Proof of Theorem B.2

Proof. We verify that the kernel K satisfies detailed balance with respect to $\pi_{\phi,\tau_t,\lambda}$. That is, we need to show that the following expression holds

$$\pi_{\phi,\tau_t,\lambda}(x)K(x \to x') = \pi_{\phi,\tau_t,\lambda}(x')K(x' \to x).$$

If $x = x'$, this holds automatically.

For $x \neq x'$, we have

$$\begin{aligned}\pi_{\phi,\tau_t,\lambda}(x)K(x \to x') &= \pi_{\phi,\tau_t,\lambda}(x)\sum_S P(S)q_S(x' \mid x)a(x,x'), \\ &= \sum_S P(S)\pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)a(x,x').\end{aligned}$$

Now fix a subset S. By definition of $a(x,x')$:

$$\begin{aligned}\pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)a(x,x') &= \pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)\min\left\{1, \frac{\pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x')}{\pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)}\right\}, \\ &= \min\left\{\pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x), \pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x')\right\}.\end{aligned}$$

Similarly, for the reverse transition:

$$\begin{aligned}\pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x')a(x',x) &= \pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x')\min\left\{1, \frac{\pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)}{\pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x')}\right\}, \\ &= \min\left\{\pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x'), \pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)\right\}.\end{aligned}$$

Hence, these expressions are equal. As this holds for every subset S, we thus have

$$\pi_{\phi,\tau_t,\lambda}(x)K(x \to x') = \pi_{\phi,\tau_t,\lambda}(x')K(x' \to x).$$

For the simplification of the acceptance ratio, assume $\pi_{\phi,\tau_t,\lambda}(x) \propto p_\phi(x)\exp(H_\lambda(x)/\tau_t)$. We have

$$\frac{\pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x')}{\pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)} = \frac{p_\phi(x')\exp(H_\lambda(x')/\tau_t) \cdot \mathbb{1}_{\{x_{S^c} = x'_{S^c}\}}p_\phi(x_S \mid x'_{S^c})}{p_\phi(x)\exp(H_\lambda(x)/\tau_t) \cdot \mathbb{1}_{\{x'_{S^c} = x_{S^c}\}}p_\phi(x'_S \mid x_{S^c})}.$$

Since $x'_{S^c} = x_{S^c}$ (by definition of q_S), the indicators are both 1. We decompose $p_\phi(x) = p_\phi(x_S \mid x_{S^c})p_\phi(x_{S^c})$ and similarly for x'. This gives

$$\begin{aligned}\frac{\pi_{\phi,\tau_t,\lambda}(x')q_S(x \mid x')}{\pi_{\phi,\tau_t,\lambda}(x)q_S(x' \mid x)} &= \frac{p_\phi(x'_S \mid x_{S^c})p_\phi(x_{S^c})\exp(H_\lambda(x')/\tau_t) \cdot p_\phi(x_S \mid x_{S^c})}{p_\phi(x_S \mid x_{S^c})p_\phi(x_{S^c})\exp(H_\lambda(x)/\tau_t) \cdot p_\phi(x'_S \mid x_{S^c})}, \\ &= \frac{\exp(H_\lambda(x')/\tau_t)}{\exp(H_\lambda(x)/\tau_t)}.\end{aligned}$$

Therefore,

$$a(x, x') = \min \left\{ 1, \frac{\exp(H_\lambda(x')/\tau_t)}{\exp(H_\lambda(x)/\tau_t)} \right\}.$$

This completes the proof. □

C. Supplementary Figures

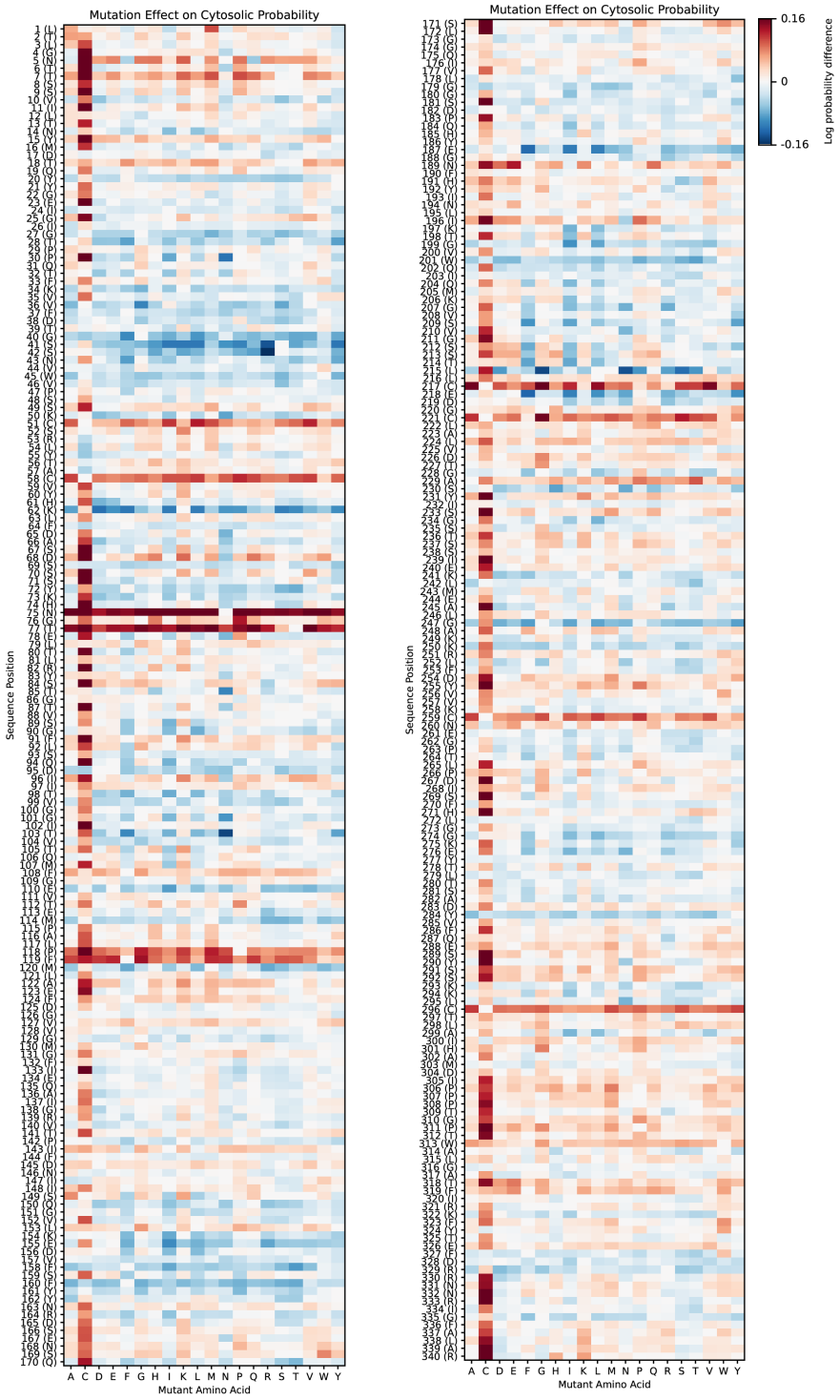

Figure 6. Full heatmap of cytosolic classifier probability LLR for every single point mutation on the renin catalytic domain to the WT sequence.

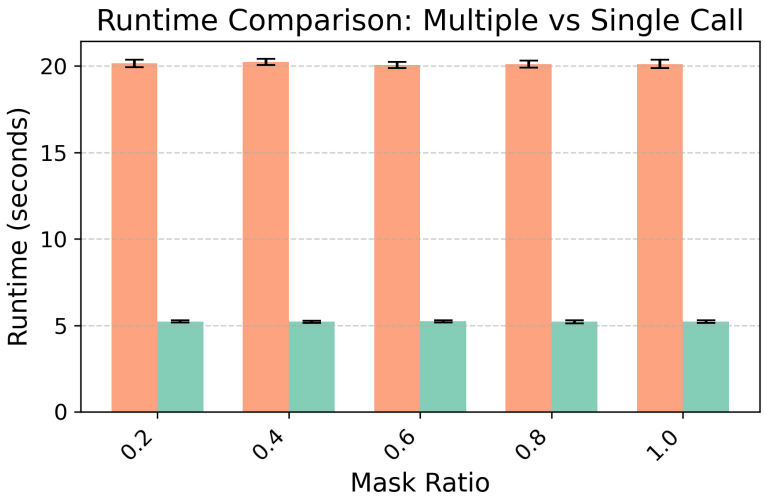

Figure 7. Average runtime for filling 10 sequences over 10 repeats via ProteinMPNN. Multiple separate calls (orange) versus one batched call (green). One batched call achieves approximately a 4x speedup.

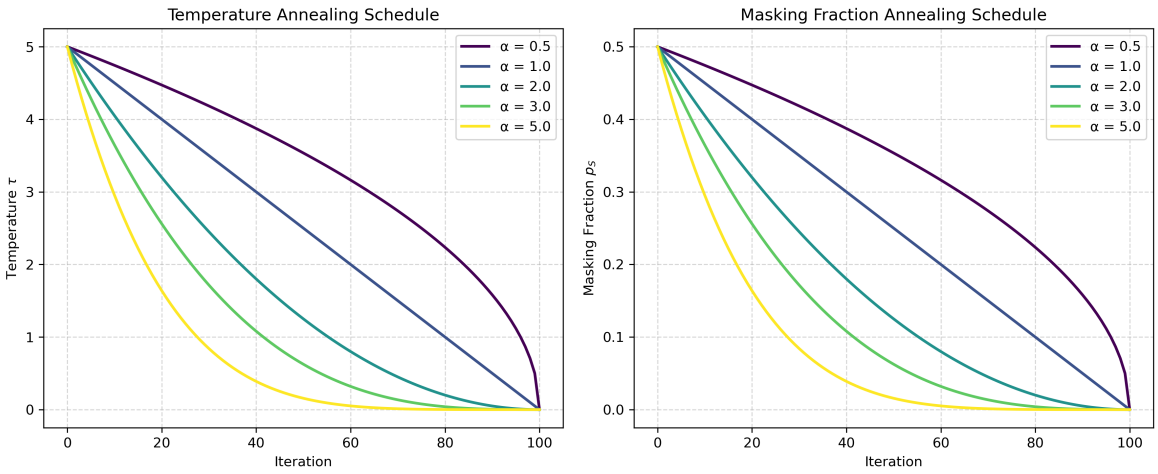

Figure 8. Comparison of power-law cooling schedules for temperature τ_t and masking fraction $p_{S,t}$ under different decay exponents α. Higher α yields a sharper initial drop and slower tail.

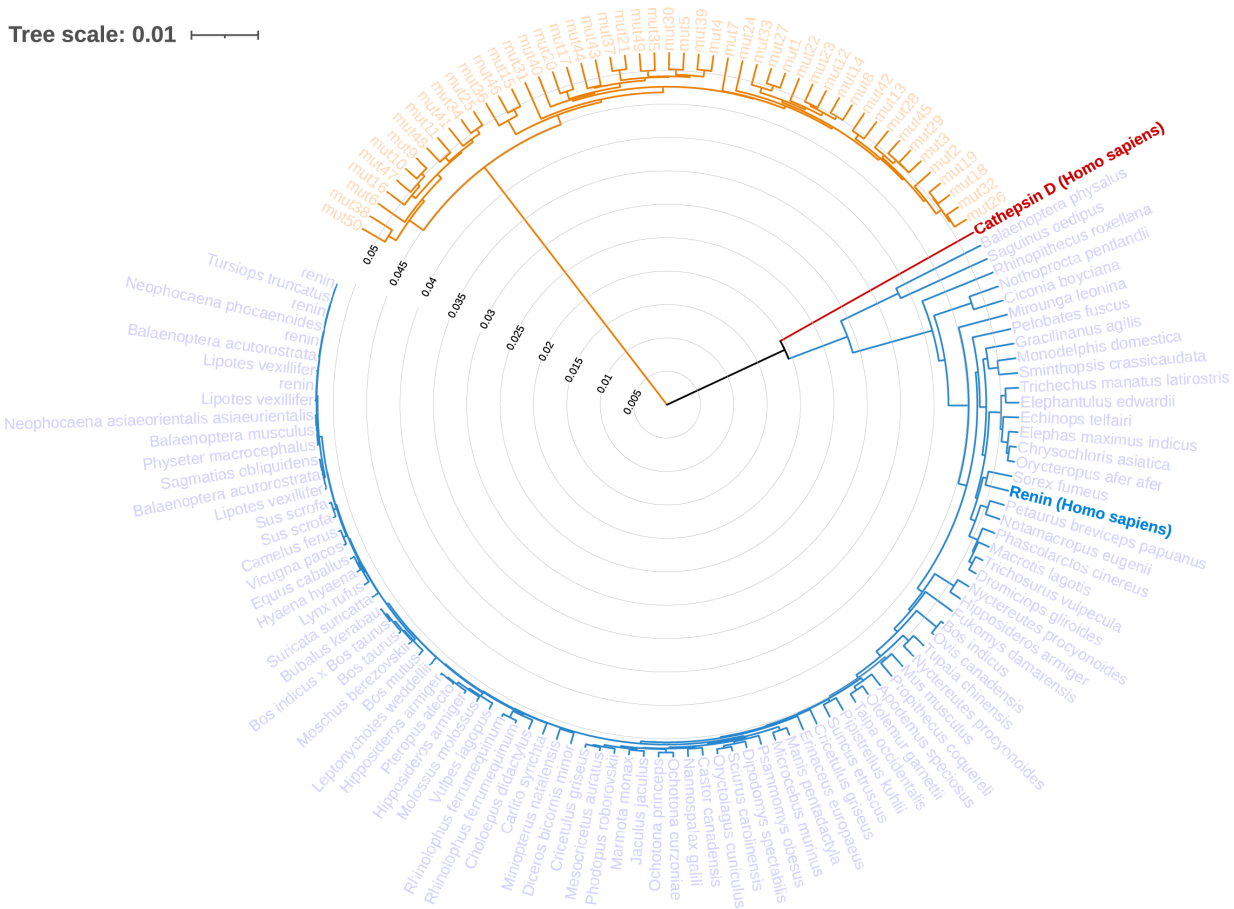

Figure 9. Circular dendrogram of MADA renin variants with renin homologs and human Cathepsin D by their cosine similarity in embedding space of ESM2. Orange: MADA cytosolic renin variants. Blue: Renin homologs retrieved by BlastP search as mentioned in **Figure 4**, highlighted node represents human renin. Red: Human Cathepsin D.

D. Supplementary Tables

Table 2. Full performance comparison of our localization classifier against three baselines on the curated Swiss-Prot dataset. Boldface indicates the best result for each metric.

Classifier	Metric	Cytosolic	Extracellular	Weighted Avg
Our Classifier	precision	**94.88**	98.92	**97.43**
	recall	97.10	**96.42**	**96.67**
	AUROC	**97.00**	**97.28**	**97.18**
	F1 score	**95.98**	**97.65**	**97.03**
DeepLoc2.0	precision	69.07	**99.72**	88.41
	recall	**97.65**	71.58	81.20
	AUROC	85.85	85.62	85.70
	F1 score	80.91	83.34	82.44
LocPro	precision	89.71	99.41	95.83
	recall	96.49	77.63	84.59
	AUROC	94.96	88.42	90.83
	F1 score	92.98	87.18	89.32
MuLocDeep	precision	37.60	97.60	75.46
	recall	97.62	2.43	37.55
	AUROC	50.75	51.16	51.01
	F1 score	54.29	4.74	23.02