Computational design of novel Cas9 PAM-interacting domains using evolution-based modelling and structural quality assessment

Cyril Malbranke [1,2*], William Rostain [2], Florence Depardieu [2], Simona Cocco [1], Rémi Monasson[1], David Bikard [2]

1 Laboratory of Physics of the Ecole Normale Superieure, PSL Research, CNRS UMR 8023, Sorbonne Université, Université de Paris, Paris, France
2 Institut Pasteur, Université Paris Cité, CNRS UMR 6047, Synthetic Biology, 75015 Paris, France

* cyril.malbranke@phys.ens.fr

Abstract

We present here an approach to protein design that enables to leverage (i) scarce functional information such as experimental data (ii) evolutionary information learned from a natural sequence variants and (iii) physics-grounded modeling. Using a Restricted Boltzmann Machine (RBM), we learned a sequence model of a protein family. We use semi-supervision to leverage available functional information during the RBM training. We then propose a strategy to explore the protein representation space that can be informed by external models such as an empirical force field method (FoldX). This method was applied to a domain of the Cas9 protein responsible for recognition of a short DNA motif. We experimentally assessed the functionality of 71 variants that were generated to explore a range of RBM and FoldX energies. We show how a combination of functional, structural and evolutionary information can identify functional variants with high accuracy. Sequences with as many as 50 differences (20% of the protein domain) to the wild-type retained functionality. Overall, 21/71 sequences designed with our method were functional. Interestingly, 6/71 sequences showed an improved activity in comparison with the original wild-type protein sequence. These results demonstrate the interest in further exploring the synergies between machine-learning of protein sequence representations and physics grounded modeling strategies informed by structural information.

1 Introduction

The growing availability of large databases of protein sequence and protein structure [1,2] is opening the way to data-driven modeling and design of proteins. When a sufficient number of sequences are available for a protein family, *i.e.*, proteins that share similar sequences, structure and functionality, it becomes possible to learn meaningful statistical features of the family of sequences. These include the position and nature of conserved residues, but also the co-variation of residues that typically occurs on evolutionary timescales when amino acids are in contact in the protein structure. Statistical models trained on multiple sequence alignments of protein families have therefore proven useful in the prediction of protein structure [3], the prediction of the effect of mutations [4], the clustering of proteins, and the design of new protein variants [5,6].

One simple yet effective approach for modeling these statistical features relies on Direct Coupling Analysis [7–9], a set of methods that learn the couplings between the different positions of a protein, revealing contacts or interactions between amino acids. Among the

different approaches to learn these couplings, the Restricted Boltzmann Machine (RBM) has the interest of providing a low dimensional continuous representation space which has been shown to be property-aware [10], and is easier to explore than the protein sequence space. However, evolution-based models often require many sequences and can have difficulties modeling the complex relationships between amino-acids, as shown by McGee et al. [11]. These sequence models also struggle to access mutations and combinations of mutations beyond those explored by evolution.

In recent years, significant efforts have been dedicated to modeling protein structures. Unlike evolutionary-based methods that rely solely on sequence information [12, 13], structural modeling of protein families incorporates physical and chemical considerations. Structural modeling approaches have been employed for various important tasks in protein design, including inferring structure from a sequence, inferring a sequence from a structure (for example backbone design with [14]), assessing the quality of a structure [15–17] and studying protein interactions with ligands such as other proteins, RNA, DNA, and metals. Over the past two decades, successful approaches have predominantly relied on empirical force fields like Rosetta [18] or FoldX [19]. These frameworks incorporate various physicochemical properties to evaluate and optimize the tertiary structure of proteins. The recent breakthrough in protein structure prediction based on deep learning algorithms [20, 21] is now generating great improvements in computational protein design, as shown in recent works [22–24].

However, some challenges remain. Many proteins react to perturbation in their environment and contact with ligands through conformational changes to carry out their function. Design efforts relying on static structure models are unlikely to work in this context.

Beyond what can be learned from natural sequence variation and structural modeling, experimental data on protein activity and specificity is sometimes available that could inform the design of novel protein variants with properties of interest. Experimental data is however usually sparse and expensive to generate. Integrating these diverse sources of information and modeling approaches in design strategies remains a challenge. In this work, we explore a method of semi-supervised training of an evolutionary-based model, an RBM, that enables to leverage sparse functional data to improve the statistical representation of the protein family. We built upon previous works [25, 26] that explored semi-supervision using a Restricted Boltzmann Machine. We further propose a sampling method based on Langevin Dynamics to explore the latent representation of the RBM while constraining single or multiple features. These features can be computed using physics-grounded models of protein properties or structure prediction, allowing the incorporation of a different source of information in the design process. For the needs of this study, we chose to work with FoldX [19] a framework that uses an empirical force field built upon the chemical terms that have been deemed important for protein stability. FoldX was recently successful in various tasks, including the prediction of the effect of mutations [27–29] or protein variant design [30]. We also used AlphaFold2 to assess the quality of the proteins generated with our method.

We evaluate our strategy on the design of Cas9 variants. The Cas9 protein is an RNA guided nuclease from the prokaryotic immune system known as CRISPR-Cas9 [31]. It forms a complex with a guide RNA, which directs Cas9 to bind complementary DNA sequences and introduce a double strand break. The first step in target recognition involves binding to a small DNA sequence motif found immediately downstream of the target sequence and known as the Protospacer Adjacent Motif (PAM). This interaction is mediated by a specific domain of the Cas9 protein, the PAM interacting domain (PID). In the case of the well described Cas9 protein from Streptococcus pyogenes (SpyCas9), the PAM motif is composed of any nucleotide followed by two guanines (NGG). The PAM requirement limits the range of DNA sequences that can be targeted by SpyCas9. Numerous efforts have therefore been dedicated over the last decade to both characterize natural variants of Cas9 with other recognition motifs (see Collias et Beisel 2021 [32] for an overview), and engineer Cas9 variants with expanded recognition

capabilities [33–35]. It was notably shown to be possible to swap the PID from SpCas9 with the PID of other natural variants [36, 37], suggesting that the PAM-Interacting domain is relatively independent of the rest of the protein. In this study, we thus chose to model the PID only, without integrating the rest of the enzyme into our design efforts.

Our approach enabled the generation of functional variants of the Cas9 PID that differ by more than 50 amino-acids from natural sequences. We demonstrate how it is possible to incorporate information from natural sequence variation, physical constrains and functional information all at once in the design of variants of a large complex protein for which a limited number of sequences are available. We provide evidence of how each category of information contributes to the overall success of the method.

2 Results

2.1 Chimera between natural PIDs and SpyCas9 show relative independence between the PID and the rest of the protein

Previous research demonstrated the function of Cas9 chimeras with recombinant PID, but only a handful of such chimeras exist in the literature [38, 39]. We therefore assessed the function of additional chimeras to better probe the modularity of the Cas9 protein and confirm that it is indeed reasonable to model the PID independently of the rest of the protein.

We built a plasmid carrying dCas9 under the control of an inducible pPhlF promoter and a guide RNA targeting the promoter of an operon expressing the reporter gene *mCherry* followed by the counter-selection marker *sacB* (see Fig 1**a**). With this system, cells expressing a non-functional variant of dCas9 will form red fluorescent colonies and die in the presence of sucrose, while repression of the operon by a functional dCas9 variant will yield low fluorescence and permit survival on sucrose (Fig 1**b**). The plasmid was designed so that the PID could be easily exchanged. We used this system to construct and test 10 PIDs with increasing phylogenetic distance to S. pyogenes Cas9 (Fig 1**c**).

The activity of the chimeras are reported as the logarithm of the repression fold f compared to an inactive dCas9 with no PID (f_0) and normalized to the WT dCas9 (f_1):

$$\text{normalized repression}(f) = \text{nr}(f) = \frac{\log(f/f_0)}{\log(f_1/f_0)} \tag{1}$$

An inactive sample will have a score of 0 and an active sample will have a score close to 1 if it is as active as the WT Cas9 PID.

SpyCas9 and all 7 chimeras within the group closest to SpyCas9 were functional, including those with differing PAM requirements, while in the out group consisting of FmaCas9, TbaCas9 and LrhCas9, the level of activity dropped significantly. These results show that modelling the PID on its own is a reasonable approach, but that the design of SpyCas9 chimera with a novel PID should be restrained to PID variants modelled to be similar to the ones seen to be functional here.

2.2 Modeling Cas9 PAM-Interacting domain with Restricted Boltzmann Machine

This section intends to present the modeling of the protein family through Restricted Boltzmann Machines (section 2.2.1) and introduce a way to better consider available functional information (section 2.2.2). In the case of Cas9 PAM-interacting domain, a major aspect of the domain's specificity is the set of PAMs that it recognizes. We show evidence that the incorporation of this functional information in the RBM improves both the ability to classify the recognized PAM

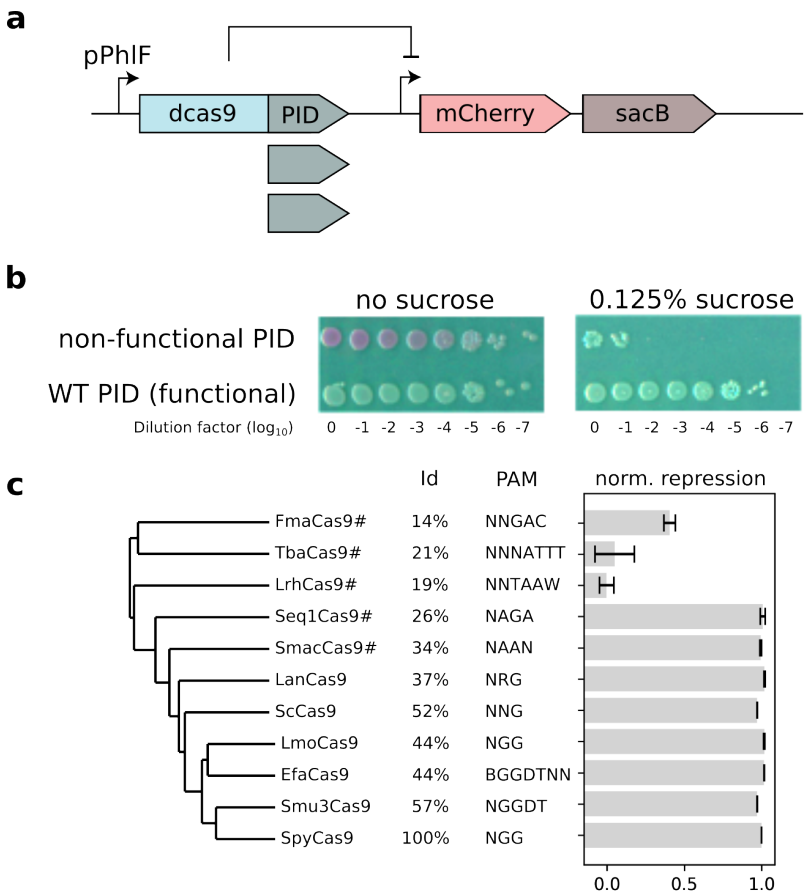

Fig 1. Experimental validation of chimeras between SpyCas9 and the PAM-interacting domain of other natural variants
a: Genetic circuit used to test Cas9 PID variants. The dCas9 gene is guided to silence an operon that consists of a mCherry reporter and the sacB counter-selection marker. The gene contruct was designed to enable the easy exchance of PID domains (see methods). **b**: Serial dilution and spotting of E. coli MG1655 carrying the wild-type SpydCas9 or dCas9 without a functional PID in the absence (left) or presence (right) of sucrose. **c**: The activity of dCas9 chimeras is reported as the normalized repression of the mCherry fluorescence. Chimeras were tested against targets with a PAM recognized by the PID they carry [32]. This score is normalized such that 1 corresponds to the activity of the WT and 0 to the negative control. The percentage of identity to the SpyCas9 PID is reported. We flagged with a # the PIDs which did not recognize TGG PAM and were therefore tested with another PAM.

sequences (section 2.2.2) and the ability of the RBM to generate high-quality sequences (section 2.4).

2.2.1 Restricted Boltzmann Machine for protein sequence-based modeling

RBMs are two-layer probabilistic graphical models (Fig 2) able to learn a distribution of probability over a set of inputs. Their modeling and generative capacities have driven a lot of attention these last decades [40]. Training of RBM is done through Stochastic Gradient Descent to minimize the negative log likelihood of the training set (see methods in section 4.2). This requires methods such as Persistent Contrastive Divergence [41] to sample approximately but more efficiently from the model distribution.

The RBM architecture is composed of (i) a visible layer encoding on a vector x an aligned protein sequence (as a sequence of amino acids and gap). (ii) A hidden layer composed of several continuous variables encoded in a vector h. These two layers are linked by the coupling matrix W^{xh}; the potentials g and U act on the units on the, respectively, visible and hidden layers. Training of the parameters Θ_{RBM} can be done through the minimization of a loss function, $\mathcal{L}_{\text{RBM}}(\Theta_{\text{RBM}}) = (g, U, W^{xh})$ whose computation is detailed in methods.

Under the right choices of potentials, it is possible to ensure that conditional sampling of each visible unit from the hidden layer and back can be done following simple distributions $p(x|h)$ and $p(h|x)$ (see methods in section 4.2 for more details).

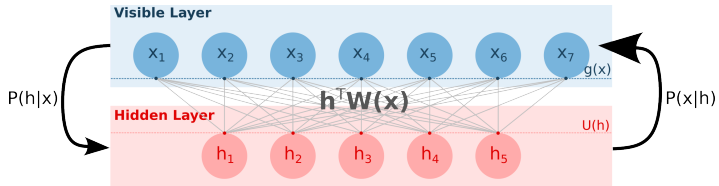

Fig 2. Schematic representation of a standard Restricted Boltzmann Machine
The RBM is a probabilistic graphical model with two layers: the visible layer encoding protein sequences x and the hidden layer encoding latent vector h. The graph is not oriented, allowing one (i) to sample from the visible layer to the representation layer using a conditional distribution $p(h|x)$ depending on x, the weight matrix W and potentials U applying on the hidden units (ii) to sample from the hidden layer to the visible layer using a conditional distribution $p(h|x)$ depending on h, the weight matrix W and potentials g representing the prevalence of the amino acids at each position.

2.2.2 Semi-Supervised Learning Restricted Boltzmann Machines (SSL-RBM)

The standard RBM is an unsupervised model that does not require any other information than amino-acid sequences. When available, experimental data on protein properties can greatly enhance the ability of models to capture meaningful information (see [26, 42]). Such experimental data is, however, usually only available for a few sequences in a protein family. We present a technique for modeling such scarce data together with the sequence using semi-supervision (see Bravi et al [26] for a semi-supervised classifier trained on top of the RBM). We incorporate a Neural Network C_{Θ_C} to the RBM model, which receives as input the hidden representation of the RBM and outputs a prediction of the supervision data (see Fig 3**a**).

In SSL-RBM, the loss function used to train the parameters of the RBM, $\mathcal{L}_{\text{RBM}}(\Theta_{\text{RBM}})$, incorporates an additive contribution, $\mathcal{L}_C(\Theta_C, \Theta_{\text{RBM}})$, to fit the parameters of the classifier to the available labelled data. This approach is similar to the one proposed in [25]. The resulting loss function is

$$\mathcal{L}_{\text{SSL-RBM}}(\Theta_{\text{RBM}}, \Theta_C) = \mathcal{L}_{\text{RBM}}(\Theta_{\text{RBM}}) + \gamma\, \mathcal{L}_C(\Theta_C, \Theta_{\text{RBM}}) \tag{2}$$

We weight the classifier $\mathcal{L}_C$ and the RBM loss $\mathcal{L}_{\text{RBM}}$ to find an equilibrium between modeling of the data (minimization of the negative log likelihood of the RBM) and performance of the classifier. If γ is small, the model will converge to a standard RBM (since only $\mathcal{L}_{\text{RBM}}$ will be important for the minimization of the loss); if γ is big, the model will look like a simple classifier unable to benefit from the joint modeling of unlabeled data. From the classification perspective, the RBM loss will act as a smart regularization loss, with unlabeled sequences structuring the RBM hidden layer.

We evaluate the SSL-RBM method by modeling the Cas9 PID (PF16595 PFAM family, 1071 sequences). As supervision data, we used a dataset of PAM recognition motifs predicted for 154

PIDs using genomics data [43]. We then evaluate the predictive capabilities of our model on a held-out validation set. We used MMseq2 to cluster the sequences with a threshold of 90% similarity, and randomly extracted a set of clusters corresponding to $\sim$15% of the sequences as our validation set. We trained 250 models with γ ranging from 10^{-2} to 10^{2}. For labelled data, each of the first 5 positions in the PAM motif is modelled independently by using as a target a binary vector encoding the recognized motifs (one unit for each nucleotide on each position, see methods in section 4.2 for more details). This enables to reduce the dimension of the target vector to a size that is more appropriate given the amount of available data. The RBM contains 200 hidden units and the classifier is a one layer dense neural network. Batch Normalization was also applied to prevent over-fitting.

For each position and nucleotide of the PAM (20 units), we computed the area under the ROC curve (AUC) for the predictions of whether a nucleotide is recognized at each position on the validation set. We then reduced it to a single mean AUC by averaging over all positions and nucleotides. We show in Fig 3**b** how this mean AUC evolves over γ. As we see, we reach a maximum for intermediate value of γ that outperforms standard RBM ($\gamma = 0$), demonstrating that supervision helps the RBM separate PIDs with different recognition motifs in the latent space. Our model also outperformed a classifier trained uniquely on the labeled sequence data (high values of γ), highlighting the contribution of unlabeled data to the determination of a useful latent representation.

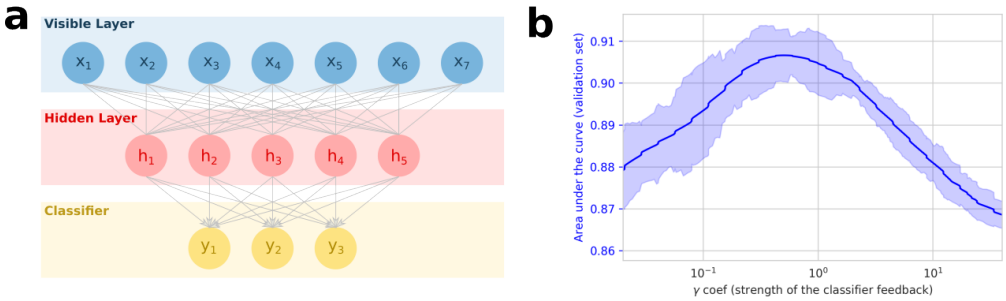

Fig 3. Semi-supervision helps the RBM learn useful representations of the Cas9 PID
a: Our Semi-Supervised Learning RBM with a one-layer classifier. This classifier is oriented and takes as input the representation of the sequence in the hidden layer. **b**: Area under the ROC curve for the prediction of the PAM on the validation set. The curve was smoothed, and we display the standard deviation over 20 consecutive values of γ through the shaded area.

2.3 Sampling variants under constraint: Constrained Langevin Dynamics in the RBM representation space

RBMs are powerful models that create an interesting representation space in which proteins with similar properties, structure, sequence, or functionality cluster together [10]. They are also relatively light models, which makes them well suited to model small datasets. Sequence models of higher complexity are easily limited by the number of available sequences [11]. Modeling frameworks relying on physical principles can also provide useful information (protein stability, binding affinity, etc.) without the need for large datasets. We explore here a novel strategy based on a Langevin dynamics process that enables direct sampling according to external inputs provided by a physics-grounded model (FoldX), while at the same time providing control over the region of the latent space being explored.

Sampling of Restricted Boltzmann Machine is usually done through Gibbs Sampling by alternating sampling of the visible layer from the hidden layer and the hidden layer from the visible layer. However, it was recently shown that Gibbs sampling does not guarantee that the

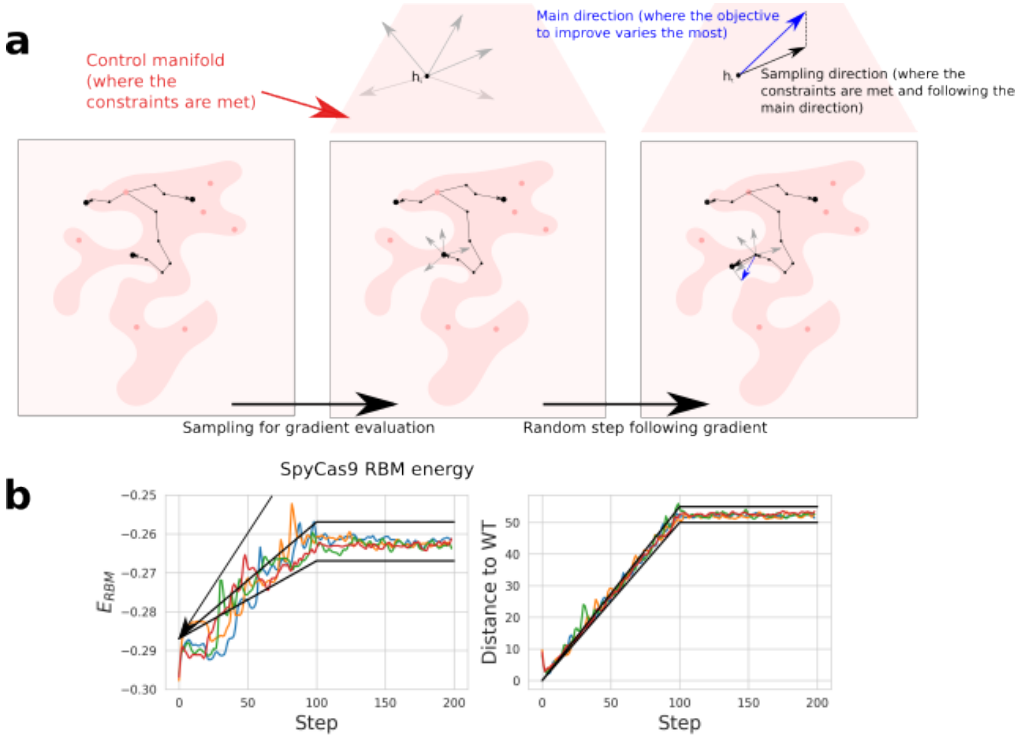

Fig 4. Constrained Langevin Dynamics as a sampling method
a: The Constrained Langevin Dynamics in the representation space consists of two steps. The first step is a round of sampling to evaluate the gradient of the main criterion and the gradient of the control criterion through the expectation formula (see methods), the second is a random step following the direction drawn by these two gradients (orthogonal projection of the main direction vector regarding the control direction). A Brownian noise is added to create randomness and diversity in the samples. **b**: Typical Random Walks obtained starting from the WT SpyCas9 PID and progressively targeting an RBM energy between -0.267 and -0.257 and a Hamming distance to the WT from 50 to 55 amino acids. The black lines are representing the time-dependent target intervals along the random walk.

full distribution of sequences will be sampled in a reasonable time [44], and this approach does not easily unlock the possibility of conditional sampling of sequences.

We propose another sampling strategy (see Fig 4**a**) that enables to improve one particular objective, such as the dissimilarity to the WT (to find functional sequences that are close or far from a natural sequence), while controlling a set of constraints (RBM score, structural stability...). This approach relies on two pillars.

First, we introduce an embedding of any function f over the protein sequence space x into the representation space h. For instance $f(x)$ may be a predictor of the biochemical properties of the protein associated to x or of its structural similarity with a known structure. The embedding, denoted by $\phi[f]$, estimates the average value of $f(x)$ for sequences x drawn from the conditional distribution $p(x|h)$, where h is a fixed representation:

$$\phi[f](h) = \mathbb{E}_{x|h}\left(f(x)\right) \tag{3}$$

The function $\phi[f]$ and its gradient are easy to estimate through sampling (see methods in section 4.3)

Introduction of the embedding ϕ above allows us to carry out a generalized Langevin dynamics in the representation space [45], to both optimize the objective f under constraints and

create diversity. Step by step, we perform a walk in the representation space by following the gradient of $\phi[f]$ and a Brownian perturbation. We also exploit the same embedding framework to enforce a set of constraints, as the embedded functions form a support to project back the representations into the space meeting the constraints:

$$h_{t+1} = \text{Proj}\left[h_t + \alpha\, \partial_h \phi[f](h_t) + \epsilon\, \mathcal{N}(0, I)\right] \tag{4}$$

Details about this projection can be found in section 4.3.

To illustrate the control of Constrained Langevin Dynamics through RBM, we carried out random walks targeting different RBM energies and Hamming distance to the reference SpyCas9 PID. The constraints drive the sequences towards target values of similarity and energy. In Fig 4**b** and Fig 4**c**, we show two examples of random walks with 5 different chains. We also display in supplementary (S5 file) some additional results on MNIST. The sequences sampled with this approach were tested experimentally, see section 2.5.

2.4 Semi-supervision helps the RBM generate higher quality sequences

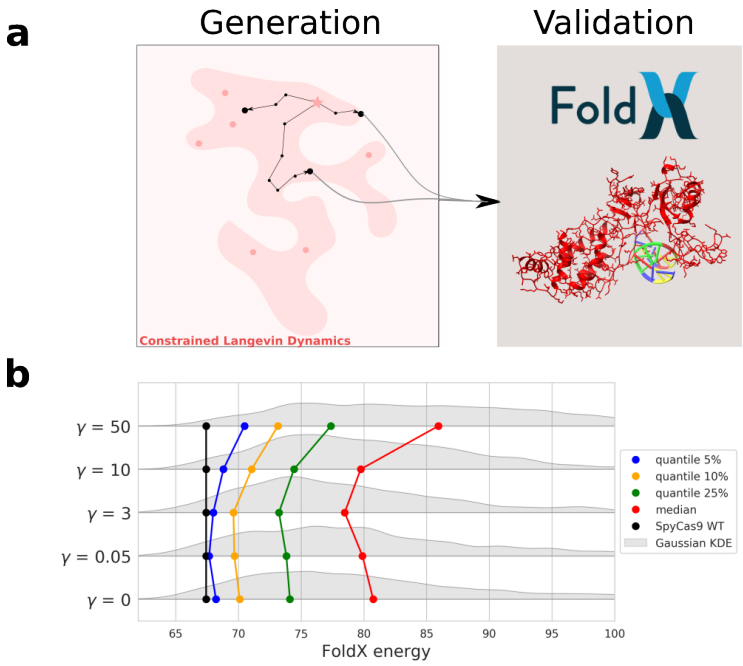

Fig 5. Generative capacities of the SSL-RBM
a: We tested the generative capacities of our models by generating 120 sequences with Constrained Langevin Dynamics using the trained SSL-RBM. We then use FoldX to compute the energy of the protein-DNA complex for the generated sequences and evaluate their quality. **b**: Distributions of FoldX energies of sequences generated with increasing values of γ. Distributions are drawn in gray using Gaussian Kernel Density Estimation, quantiles are also displayed for the different distributions. These quantiles and distributions show that overall, SSL-RBM trained with intermediate values of γ tend to generate sequences with better (lower) FoldX energies.

Before testing variants of Cas9 experimentally, we first tested the generative capacities of the SSL-RBM *in silico*. To do so, we used Constrained Langevin Dynamics to generate 120 protein sequences per RBM, with an RBM energy comparable to the one of the reference sequence (SpyCas9) and numbers of mutations ranging from 10 to 50. Sequences were generated in this manner using values of γ ranging from 10^{-2} to 10^{2}.

We then used FoldX to obtain a fast evaluation of the quality of the generated sequences (see Fig 5**a**). The evaluation was performed by taking the structure of the Cas9-PID in complex with DNA (PDB-ID : 5y36). We then replaced the amino acids by their values in the generated sequences, and repaired the structure by fixing the side chains. Functional sequences are expected to have a resulting FoldX energy close to that of the WT sequence.

We show the distribution of FoldX DNA energies obtained for different values of γ in Fig 5**b**. We observe a balanced regime where intermediate values of γ are better than in the regimes with weak (small values of γ) and with strong (high values of γ) feedbacks from the classifier. This results indicate that semi-supervised training on functional data, here the PAM recognition sequence, can improve the overall ability of the model to generate proteins that are likely to fold properly.

2.5 Design of SpyCas9 Pam-Interacting domains

2.5.1 Sampling of the RBM-SSL using Langevin Dynamics leads to functional PAM interacting domains

To evaluate the efficacy of our method, we employed Langevin Dynamics to create a set of sequences with varying degrees of similarity to SpyCas9, RBM scores, and predicted levels of protein-DNA complex stability determined by FoldX energy calculations. These sequences were generated utilizing the representation space of a self-supervised learning Restricted Boltzmann Machine (SSL-RBM) trained with a parameter value of $\gamma = 5$. Subsequently, the designed sequences underwent experimental assessment. Detailed information regarding all the sequences, including their corresponding AlphaFold-predicted structures, can be found in the supplementary materials (S1 file). Furthermore, computational scores and experimental validation results for these sequences are also provided (S2 file).

We adopted two different methods for sampling:

- The first method (and batch) aimed at increasing the Hamming distance to SpyCas9 WT, while constraining both the RBM log likelihood and the FoldX score within a certain interval (sequences referenced dgfx1 to dgfx29).
- The second method (and batch) constrained the RBM score and the Hamming distance to SpyCas9 within a certain interval using Langevin Dynamics, and then selected sequences with good FoldX score *a posteriori* (sequences referenced dgfx30 to dgfx75). We generated variants aiming for intervals of 5 mutations spanning from 10 to 70 mutations ([10, 15], [15, 20], ..., [65, 70]) and intervals of RBM energy spanning from -0.21 to -0.07 ([-0.21,-0.19], ..., [-0.09, -0.07]).

The first method is more computationally expensive than the second one, as FoldX energies need to be computed at each step of the Langevin dynamics to evaluate the gradient of this function in the representation space. In the second method, FoldX scores are only computed after sampling as a filtering step once the model is already trained. This method could be easily implemented since a large share of sampled sequences have a good FoldX score (see section 2.4, Fig 2.4)

We selected 71 sequences spanning different predicted scores to be assessed experimentally (29 with method 1 and 42 with method 2). DNA sequences encoding the desired proteins were synthesized and introduced in E. coli. The ability of the generated proteins to bind a target position with a NGG PAM motif was evaluated using a mCherry reporter gene as described in section 4.4 (see methods). The activity of dCas9 is reported as the normalized repression nr defined in Eq. 1 (Fig 6). We noticed that a fair share of sequences showed a good activity score in both batches, with 17/29 sequences presenting some activity in the first batch (including 10

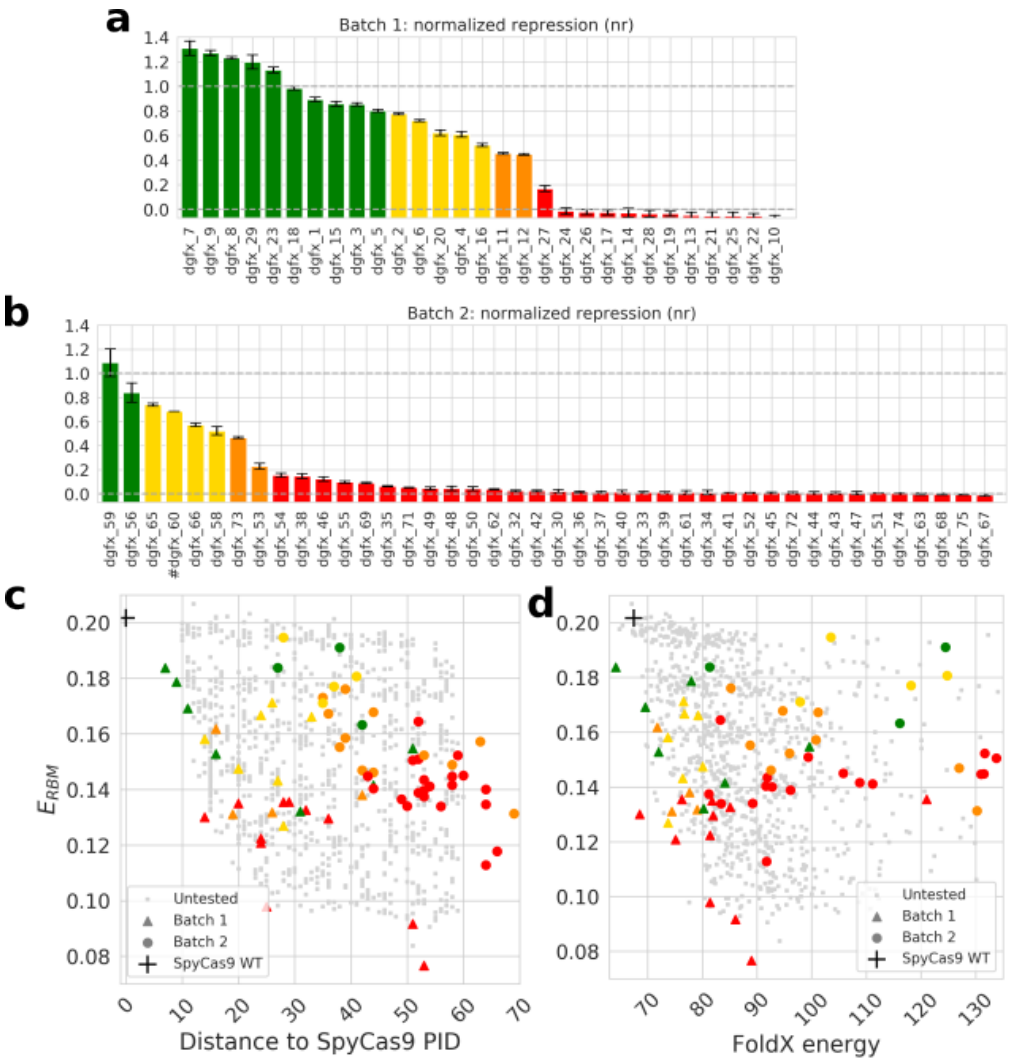

Fig 6. Experimental evaluation of generated sequences and relationship with Hamming distance, RBM and FoldX scores
a and **b**: The activity of the dCas9 chimera carrying generated PIDs was measured using the mCherry fluorescence assay. Arbitrary thresholds were used to classify proteins as very functional (green, nr > 0.8), functional (yellow, $0.5 <$ nr < 0.8), marginally functional (orange, $0.2 <$ nr < 0.5), not functional (red, nr < 0.2). Bar plot **a** represents the sequence in the first batch of tested sequences, and bar plot **b** represents the sequences in the second batch of tested sequences. Error bars are plotted using 2 σ, where σ in the standard deviation between independent measurements (see methods) (# dgfx_60 large standard deviation obtained from experimental measurement was removed from the figure for clarity) **c**: Scatter plot of generated protein sequences as a function of the RBM energy and Hamming distance of the sequence to that of the WT. Gray dots are untested sequences generated through Constrained Langevin Dynamics. We found very active sequences with up to 50 amino-acids difference to the WT and any known natural sequences **d**: Scatter plot of generated protein sequences as a function of the RBM energy and the FoldX energy.

with an activity that matches or supersedes the WT). The second batch also displayed 8/42 variants with activity, including 2 with an activity similar to the WT. Notice that the second batch includes sequences designed with relaxed objectives of distance to SpyCas9 PID, RBM

and FoldX energy compared to the first batch (see below). On average, the sequences in the second batch had (by design) worse RBM and FoldX energy and a larger Hamming distance to the wild-type. The lower amount of functional variants is likely a reflection of these relaxed constraints.

In Fig 6**c** and Fig 6**d**, we display the activity of the generated sequences compared to the RBM score, Hamming distance, and the FoldX energy. We see that some sequences are still active while being quite distant from the SpyCas9 sequence (50 amino acids), as long as they keep a low RBM Energy. The bottom-right part of Fig 6**d** (good FoldX score, good RBM score) displayed a high proportion of active or very active sequences. We also show sequences generated by Constrained Langevin Dynamics as part of batch 2, but that were not experimentally tested. We notice that a substantial part of the generated sequences display a good FoldX score, which justifies the use of this structural score as a post-processing step rather than a constraint during Langevin Dynamics sampling in the second method.

2.5.2 Semi-supervision of the RBM leads to better prediction of Cas9 activity

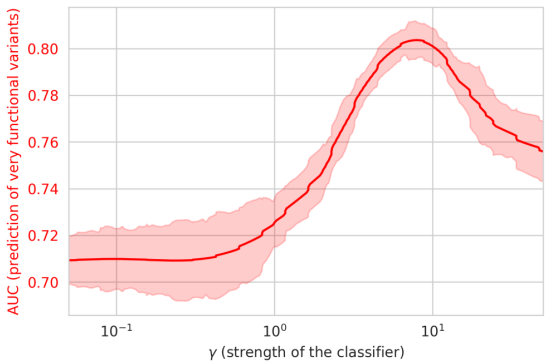

Fig 7. Correlation between SSL-RBM energy and experimental activity
The correlation between the experimental activity and the SSL-RBM score is plotted as a function of the classifier strength (γ) (red curve). The curve was smoothed, and we display the standard deviation over 20 consecutive values of γ.

We then investigated whether our experimental data would corroborate the *in silico* results obtained in section 2.4 that semi-supervision improved the ability of the RBM to generate good sequences. To that end, we computed the correlation between the RBM score and the experimentally determined Cas9 activity, using models trained with different values of γ (the strength of the classifier during training). We can see (Fig 7) that the correlation between the RBM score and Cas9 activity improves when the unsupervised and supervised losses are balanced. Interestingly, the optimal value of γ in this situation is slightly higher than the optimal value for the classification task, even though a higher γ gives more strength to the classifier.

Overall, these findings offer additional evidence that semi-supervision enhances both the RBM's capacity to separate Cas9 PIDs based on their PAM recognition sequences and its ability to identify functional sequences. This further confirms that the inclusion of extra information through this approach generally enhances the model's performance, even when the supervision did not relate to predicting activity, but rather specificity.

2.5.3 A combination of sequence and structural information best predicts sequence activity

While our RBM score is on its own a good predictor of Cas9 activity, we see that the FoldX score provides complementary information and that the combination of both features can be

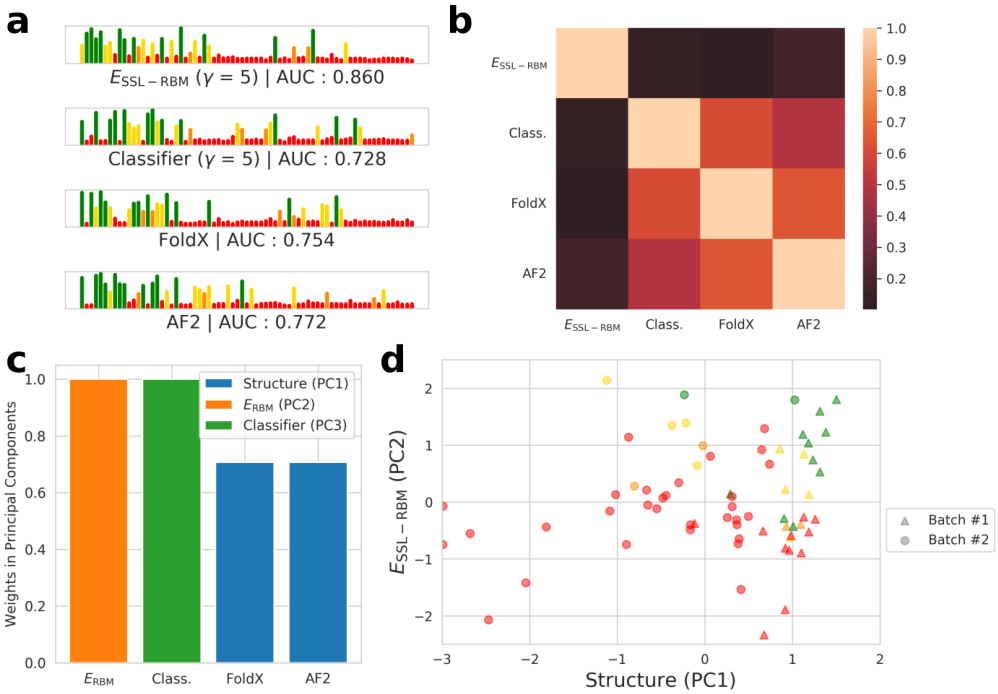

Fig 8. RBM energy and structure scores predict sequence activity
a: Sequences ordered by their score according to each of the metrics, the height of the bars is the activity measured experimentally. The RBM score displayed the best Spearman correlation, with AlphaFold showing also very good performance in detecting functional samples (AUROC curve for detecting sample with activity over 0.5). **b**: Correlation between the different scores. **c**: Three components emerge from a sparse PCA: a structure component (combination of AlphaFold and FoldX scores), the RBM energy and the classifier score. **d**: Experimental results displayed along the two principal components inferred by the sparse PCA. The Cas9 PID variants are colored according to their activity, as in Fig 6.

advantageously used to identify active Cas9 variants. We further attempted to compare the information obtained by FoldX with the more recent protein structure prediction tool AlphaFold. AlphaFold2 is a framework that uses modern neural networks to predict a distance map from a sequence and an alignment of the sequence homologs. This distance map is then used to build a 3-dimensional model of the protein structure through physically motivated methods. For each experimentally tested variants, we predicted a structure through an AlphaFold model (using ColabFold [46]) and compared this structure to the one predicted by the same AlphaFold model for the wild-type protein. Last, we also compared the experimental data with the classifier score for predicting how likely a PID sequence is to recognize a NGG motif.

On Fig 8**a**, we display the correlation with the structural metrics, the RBM score and the classifier score. We also computed the Area Under the ROC Curve (AUROC) for the detection of active samples. Each bar plot displays the sequences by their scores according to the RBM, FoldX and AlphaFold, while their color and height corresponds to their experimental activity.

All scores were able to identify functional samples (activity over 0.5) with relatively high accuracy, with the RBM displaying the best results, followed by AlphaFold. Interestingly, models seemed to capture complementary information, as can be seen in the failure of AlphaFold to spot inactive samples which are correctly predicted as active by the RBM and vice-versa.

We noticed that AlphaFold and FoldX scores were highly correlated (Fig 8**b**). A Sparse PCA

performed on the vectors of the different scores associated to each sequence revealed three main components: the first one encompassing the structural scores from FoldX and AlphaFold, the second one the RBM score alone, and the third one the classifier score alone (Fig 8**c,d**). This confirms our observation that the structural scores are strongly correlated, while the SSL-RBM brings another view of the data. We also trained Logistic Regression with cross validation using the different scores and computed the Area Under the Curve for their ability to predict functional variants. While the RBM energy alone had an AUC of 0.86 (see on Figure 8**a**), combining the RBM energy with the classifier score (AUC = 0.920), the structural scores (AUC = 0.918), or both (AUC = 0.929) improved discriminative performance, as displayed in supplementary materials (S3 file).

The classifier score (pseudo-likelihood score for the recognition of the NGG motif) correlates well with the activity and is complementary to the RBM energy, see Fig 8**b**. Let us however stress that we focused in this work on the NGG PAM only. The reason to do so is two-fold. First, the NGG motif is well represented in our initial labelled data. Second, this motif is recognized by our reference enzyme and by the large majority of close natural variants present in our dataset, as indicated by the comprehensive phylogenetic tree provided in supplementary (S6 file). Due to these limitations, our evolution-based method is not ideal for generating close variants with modified PAM specificity.

2.5.4 Strengths and weaknesses of the RBM and structural scores

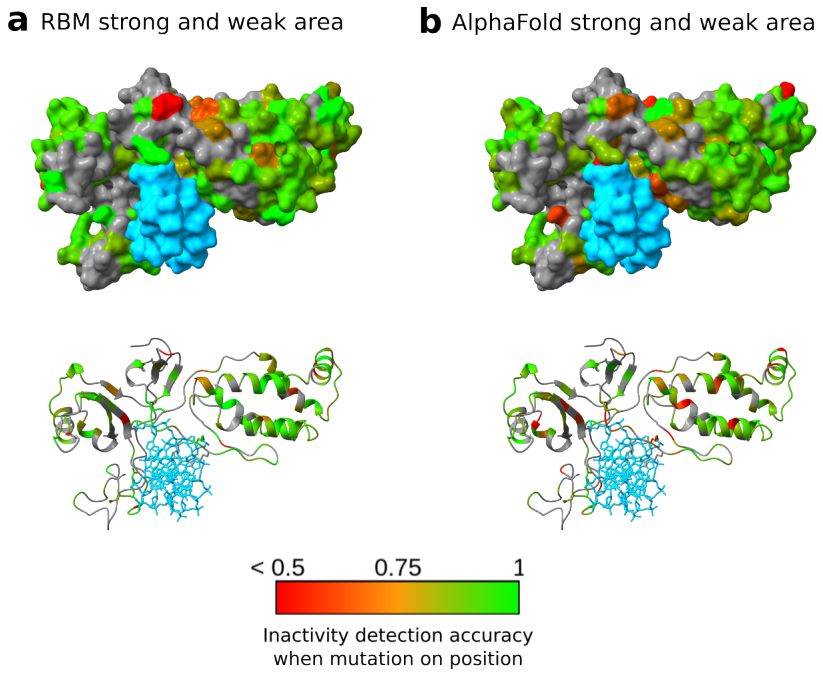

Fig 9. Strengths and weaknesses of the RBM and AlphaFold models
For each position, we display the accuracy for detective inactive variants using the RBM energy (**a**) or the AlphaFold RMSD (**b**) as a function of the localization of the mutations on the protein sequence and structure. DNA is shown in blue. In green are the areas where the deleterious effect of mutations are correctly predicted, while in red are the areas where the deleterious effect of mutations in incorrectly predicted. Positions not mutated in our experimental dataset are shown in gray. As we can see, AlphaFold RMSD predicts the effects of mutation on the binding interface with the DNA more poorly than the RBM.

To better understand the respective strengths and weaknesses of the RBM and structural

scores, we used the experimentally tested sequences to investigate whether specific positions in the structure were better modeled by the RBM or by AlphaFold. For each residue on the protein sequences, we gathered all the inactive sequences (normalized repression below 0.2) with a mutation at this position. We then checked whether the RBM energy or the AlphaFold RMSD could be used to correctly predict that the variants were inactive. We computed the ratio between the number of sequences correctly predicted as inactive by each of the scores and the total number of inactive sequences. On Fig 9 we display this accuracy for the RBM energy and AlphaFold (Fig 9**b**). The accuracy on positions were displayed only if at least five variants included mutation at the position.

One of the key observations is that AlphaFold performed more poorly than the RBM on residues that interact with DNA (represented in sky blue on Fig 9**b**). This can likely be explained by the fact that AlphaFold was not trained to consider interactions with other molecular partners, such as DNA.

3 Discussions

Structural biology has been a field of major breakthroughs these last years with the increasing success of Deep Learning models for secondary structure prediction [47, 48], ligand or protein binding site prediction [49, 50] and of course tertiary structure prediction with frameworks such as AlphaFold2 [20], RoseTTAFold [21] or ESM Fold [51]. Older frameworks like Rosetta [18], Amber [52] or FoldX [19] have been developed around an empirical force field to encompass molecular and atomistic interactions. These frameworks rely on strong physicochemical priors designed to tackle the general lack of structural data. Even today, while hundreds of millions of protein sequences are available in public databases, only tens of thousands of structures are available since their determination is complex and expensive. Force fields can evaluate the compatibility between an amino acid sequence and a given structure, and are generally more robust to over-fitting than data-based models.

Machine learning methods have also been applied to the modeling of proteins based on their primary sequence information alone. Such efforts range from simple models, like the RBM employed here, trained on single protein families, to large language models trained on hundreds of million of diverse protein sequences with impressive successes in various tasks [12, 13, 53–55]. Interestingly, these models typically provide a property-aware latent representation of the protein that can be leveraged together with the generative capabilities of the models for various design tasks [56–58]. We investigate here a strategy to combine these evolutionary models together with physically grounded models for protein design, in line with previous works to leverage the strength of both approaches [59].

We propose the use of Langevin Dynamics as a method enabling the exploration of a latent representation informed by external optimization goals, such as physics-grounded models of the protein properties. Here, we demonstrated this by constraining the latent space exploration using FoldX energies. Alternatively, physics grounded optimization goals can be used as a post-filtering step on proteins generated by the evolutionary model. Our method yielded functional Cas9-PAM Interacting domain variants differing by more than 50 amino-acids from any known natural sequence. The data generated highlighted how evolution-based modeling and structural modeling capture different information, and how the combination of both can advantageously be used to identify functional variants with high confidence. AlphaFold and FoldX can predict the protein structure but miss some information about the functionality of the sequence that can be inferred directly from the first and second order statistics of the Multiple Sequence Alignment. This observation could drive further theoretical works to better understand and circumvent the blind spots of structural modeling. One possibility is that RBMs capture information related to the protein biological function, such as its ability to undergo

conformational changes or to interact with other molecular substrates, that are reflected in the sequence statistics of the protein family. Such information might, however, be missed by AlphaFold or FoldX if it is not directly relevant to the stability of the protein fold.

In this work, we also demonstrated how scarce functional data about the protein family can be used to improve the model using semi-supervision. We used data on the PAM binding sequence of the Cas9 PID available for 14% of our dataset. The use of this data during RBM training improved both the ability of our model to predict PAM sequences based on the PID sequence, and more interestingly contributed to improving the accuracy of our model in predicting the activity of Cas9 variants. Surprisingly, our model generated Cas9 variants that were more active than SpyCas9 for the task of gene silencing, even if this was not an explicit goal of our design. A possible explanation is that the constraints we placed on FoldX energies led to the generation of proteins that bind the target DNA more stably and are therefore more active in the gene silencing task. A future direction of this work will be to employ our models to generate Cas9 variants with modified PAM specificity while maintaining wild-type level efficiency or even improving on it. One promising strategy could involve employing Deep Learning frameworks like ProteinMPNN [60], or other platforms specifically designed to model binding interactions [61]. These frameworks could serve as viable alternatives to FoldX, offering enhanced flexibility and potentially more robust modeling capabilities.

Altogether, our results highlight the possibility to further improve our protein modeling capabilities by incorporating, structural, evolutionary but also functional information into our models and protein design process.

4 Material & Methods

4.1 Dataset construction

4.1.1 Data collection

The dataset used in this study is the PFAM family PF16595 which contained 1102 sequences at the time of data collection, including the PID of SpyCas9.

Supervision data for the SSL-RBM consisted in the PAM recognition sequence of 154 PID variants obtained from Vink et al. [43]. This dataset contains PAM sequences inferred from the alignment of regions adjacent to target sequences of natural CRISPR-Cas systems.

4.1.2 Data engineering

To build a validation set, sequences of the PF16595 PFAM family were clustered using MMSEQS2 with a 90% similarity threshold [62]. Clusters were randomly extracted one by one to build a validation set of around 10% of the total set, ensuring that sequences in both sets labelled or unlabeled were not too similar. Sequences were also weighted based on the size of the cluster they belong to, in order to avoid assigning excessive importance to data that is too similar. Concretely, let x be a sequence and $\mathcal{C}(x)$ its cluster built through MMSEQS2. Over a mini-batch, each sequence is weighted according to $\frac{1}{\text{card } \mathcal{C}(x)}$ for the computation of the loss, following the protocol proposed in Tubiana et al. [10]. This ensures that all clusters have the same importance over a full iteration of the data, which reduces the biases that could emerge from the composition of the training set.

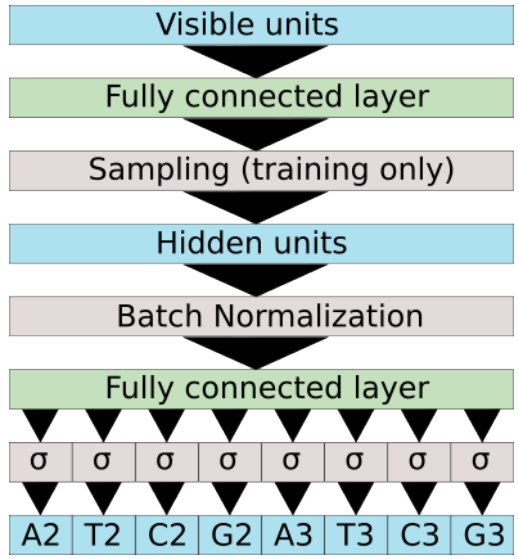

Fig 10. Architecture of the SSL-RBM classifier. During training the classifier takes as input the hidden layer after sampling (equivalent to a noisy representation of the input sequence) while during prediction it uses the hidden representation. The hidden representation is then batch normalized and fed into a fully connected layer. The network attempts to predict if the PID recognizes each nucleotide at each position and in the PAM.

4.2 Restricted Boltzmann Machine

4.2.1 Architecture

The RBM has a 2-layers architecture, the visible layer has 736 visible units (length of the Multiple Sequence Alignment), encoded by Multinoulli variables with 21 states (20 amino acids + gap). The hidden layer includes 200 Gaussian units. Each of the hidden units is connected to each of the visible units, ensuring conditional sampling of each visible unit from the hidden layer and back. The weights are initialized with Xavier initialization [63], following the distribution $\mathcal{U}(-\frac{1}{\sqrt{qn_x}}, \frac{1}{\sqrt{qn_x}})$ where $n_x = 736$ is the length of the sequences and $q = 21$ the number of states per unit.

We define $x = (x_i)_i$ an amino-acid sequence of length n_x. This first layer of the RBM is defined by $g(x)$ where g is a linear potential that indirectly represents the prevalence of different amino acids at each position. The hidden units $h_\mu, \mu = 1, ..., 200$ are subject to the potentials $U_\mu(h_\mu)$. These two layers are linked by the coupling matrix W_{xh} of size $n_x \times n_h$ such that :

$$P(x, h) = \frac{1}{Z} \exp\left[\sum_i g_i(x_i) + \sum_\mu U_\mu(h_\mu) + \sum_{i,\mu} h_\mu W^{xh}_{i\mu}(x_i)\right] \tag{5}$$

where g_i and W are trainable and $U_\mu(h_\mu) = \frac{1}{2}h_\mu^2$is fixed.

The conditional probability of hidden unit μ reads

$$P(h_\mu|x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}(h_\mu - \sum_i W^{xh}_{i\mu}(x_i))^2\right] ,$$

which can be simulated with a Gaussian Law $\mathcal{N}(\left(\sum_i W^{xh}_{i\mu}(x_i)\right)_\mu, I)$. Similarly, the

conditional probability of visible unit i reads

$$P(x_i|h) = \text{softmax}\left[g_i(x_i) + \sum_\mu h_\mu W_{i\mu}^{xh}(x_i)\right] ,$$

Which can be simulated with a multinomial law.

The SSL-RBM (see Fig 10) has the same architecture, except that we add a classifier to the previous architecture. The classifier is a one-layer neural network taking as inputs the hidden configuration (after sampling according to the potentials) and after a batch normalization layer. This ensures that more diverse input will feed the classifier (data augmentation), which enhances the quality of training.

The classifier (Fig 10) was trained to predict PAM recognized by the PAM-Interacting domain in input. We encoded the first n (here $n = 5$) positions of the motifs and the four nucleotides in a $\{0,1\}^{n\times 4} = (A_1, T_1, C_1, G_1, ..., T_n, C_n, G_n)$ vector with $X_k = 1$ if nucleotide X is accepted in position k . For example, a NGGNN motif will be encoded by (1,1,1,1,0,0,0,1,0,0,0,1,1,1,1,1,1,1,1,1).

4.2.2 Training the RBM

The training of RBM is done through the maximization of the likelihood of the training set T. Let Θ_{RBM} be the set trainable parameters we want to optimize. In the general case we have $\Theta_{\text{RBM}} = (g, U, W)$. We denote by:

$$p(x|\Theta_{\text{RBM}}) = \log[\frac{1}{Z(\Theta_{\text{RBM}})}\exp\left[-E_{\text{RBM}}(x|\Theta_{\text{RBM}})\right]] = -E_{\text{RBM}}(x|\Theta_{\text{RBM}}) - \log Z(\Theta_{\text{RBM}}) \tag{6}$$

the log-likelihood of a sequence x given the set of parameters Θ_{RBM}. We perform maximum likelihood estimation on the training set to optimize the parameters.

$$\Theta^*_{\text{RBM}} = \underset{\Theta_{\text{RBM}}}{\arg\max} \sum_{x\in T} p(x|\Theta_{\text{RBM}}) \tag{7}$$

We then denote the loss function for training the RBM throughy maximum likelihood estimation to be:

$$\mathcal{L}_{\text{RBM}}(\Theta_{\text{RBM}}) = \mathbb{E}_{x\sim T}\left[-\log P(x|\Theta_{\text{RBM}})\right] \tag{8}$$

One way to do this is to perform Gradient Descent. We have to estimate:

$$\mathbb{E}_{x\sim T}\left[\frac{\partial p(x|\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}}\right] = -\mathbb{E}_{x\sim T}\left[\frac{\partial E_{\text{RBM}}(x|\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}}\right] - \frac{\partial \log Z(\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}} \tag{9}$$

It is easy to approximate the first term with mini batches of samples from the training set T. The other term, however, is trickier :

$$\frac{\partial \log Z(\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}} = \frac{1}{Z(\Theta_{\text{RBM}})}\frac{\partial}{\partial \Theta_{\text{RBM}}}\sum_x e^{-E_{\text{RBM}}(x|\Theta_{\text{RBM}})} \tag{10}$$

$$= -\sum_x \frac{\partial E_{\text{RBM}}(x|\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}} P(x|\Theta_{\text{RBM}}) \tag{11}$$

where $P_{\Theta_{\text{RBM}}}$ the distribution of the RBM model, denoted by M. We then have

$$\mathbb{E}_{x\sim T}\left[\frac{\partial p(x|\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}}\right] = \mathbb{E}_{x\sim M}\left[\frac{\partial E_{\text{RBM}}(x|\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}}\right] - \mathbb{E}_{x\sim T}\left[\frac{\partial E_{\text{RBM}}(x|\Theta_{\text{RBM}})}{\partial \Theta_{\text{RBM}}}\right] \tag{12}$$

While sampling from the training set is easy, sampling from the model distribution is harder. For this, we mainly rely on **Persistent Contrastive Divergence** [41]. We implemented our own RBM in PyTorch (available on https://github.com/CyrilMa/torchpgm-standalone) running on GPU, and we used the AdamW [64] optimizer as it yielded a more robust training.

4.2.3 Training the SSL-RBM

Training the Semi-supervised Learning RBM requires the use of an additional training set T_{sup} of data containing a set of sequences x labelled by y. We add a term to the loss function defined previously so that we can train both the parameters of the RBM Θ_{RBM} and the parameters of the classifier Θ_C:

$$\mathcal{L}_{\text{SSL-RBM}}(\Theta_{\text{RBM}}, \Theta_C) = \mathcal{L}_{\text{RBM}}(\Theta_{\text{RBM}}) + \gamma \mathcal{L}_C(\Theta_C, \Theta_{\text{RBM}}) \tag{13}$$

where γ is a parameter that weights the relative contribution of each term of the loss function.

In the case of a classification task, we want to minimize a cross entropy term over the training set T_{sup} composed of labelled data:

$$\mathcal{L}_C(\Theta_C, \Theta_{\text{RBM}}) = \mathbb{E}_{x \sim T_{sup}, h \sim (h|x, \Theta_{\text{RBM}})} \left[\sum_i y_i \log(C_{\Theta_C}(h))_i \right] \tag{14}$$

where $y = (y_i)_i$ is a vector encoding for a label (specifically the PAM) and $C_{\Theta_C}(h)$ is the output vector of the classifier which takes in input stochastic representation of x. Estimation of this loss function is done using mini batches of labelled protein sequences x, with h sampled from the distribution of $x|h$. We point out that the loss of the classifier $\mathcal{L}_C$ depends on the parameters of the RBM.

4.3 Sampling RBM through Constrained Langevin Dynamics

4.3.1 Function embedding in RBM representation space

Let f be an evaluation function that on the visible space, that is, on a sequence x. We are interested in the optimization of this function in the hidden space. To achieve this, we consider $\phi[f](h) = \mathbb{E}_{x|h}[f(x)]$ and its gradient $\partial\phi[f](h) = \vec{\nabla}_h \phi[f](h)$. We can write

$$\vec{\nabla}_h \mathbb{E}_{x|h}(f(x)) = \vec{\nabla}_h \sum_x P(x|h) f(x) = \sum_x f(x) P(x|h) \vec{\nabla}_h \log P(x|h) \tag{15}$$

and,

$$\vec{\nabla}_h \log P(x|h) = Wx - \sum_x Wx\, \mathbb{P}(x|h) = Wx - \mathbb{E}_{x|h}(Wx) \tag{16}$$

We conclude that :

$$\begin{aligned} \vec{\nabla}_h \mathbb{E}_{x|h}(f(x)) &= \sum_x f(x) P(x|h) \vec{\nabla}_h \log P(x|h) && (17) \\ &= \sum_x f(x) \left(Wx - \mathbb{E}_{x|h}(Wx) \right) \mathbb{P}(x|h) && (18) \\ &= \mathbb{E}_{x|h}(f(x)Wx) - \mathbb{E}_{x|h}(f(x)) \mathbb{E}_{x|h}(Wx) && (19) \end{aligned}$$

We then have a simple way to get estimates of $\phi[f](h) = \mathbb{E}_{x|h}[f(x)]$ and of its gradient $\vec{\nabla}_h \phi[f](h)$ by sampling the conditional probability distribution $x|h$, which has been shown to be easy above. We denote by $\hat{\phi}[f](h)$ and $\hat{\partial\phi}[f](h)$ these estimators.

4.3.2 Improving an objective through Langevin Dynamics (LD) in representation space

Let f be a criterion in the visible space, then by considering $\phi[f]$ and $\partial\phi[f]$ defined in section 4.3.1, we can build a directed random walk as follows :

Initialization: Let x_0 be the starting sequence. We initialize our random walk by $h_0 = \arg\max_h \mathbb{P}(h|x)$, which, for quadratic potential U, is simply $h_0 = Wx_0$.

Recurrence: Let h_t be our position in the representation space at time t, then we update

$$h_{t+1} = h_t + \gamma \left(\hat{\partial\phi}[f](h_t) + \epsilon \mathcal{N}(0, I) \right) , \tag{20}$$

where γ and ϵ are two parameters to be chosen wisely and where $\mathcal{N}(0, I)$ is a Brownian noise.

We then have in fact two sources of randomness in our random walk. The first one is the Brownian noise, controlled by ϵ, and the second one is $(\hat{\partial\phi}[f](h_t) - \partial\phi[f](h_t))$, which depends on the variance of the estimator (and thus can be lowered by raising the number of samples if needed).

This procedure, however, is problematic because the norm of h tends to increase to values not encountered for representation of samples in the training set. In other words, we would explore a region of the representation space that does not correspond to our protein family data.

To address this problem, we introduced a regularization term, and now look for the maximum of $\phi'[f](h) = \phi[f](h) - \frac{1}{2}\lambda\|h\|_2^2$. This modifies the update rule in Eq. (20) as follows :

$$h_{t+1} = h_t + \gamma \left(\hat{\partial\phi}[f](h_t) + \epsilon \mathcal{N}(0, I) - \lambda h_t \right) ,$$

with λ to be appropriately chosen. We then rewrite this equation as

$$h_{t+1} = h_t + \gamma \, , \mathcal{C}_{\epsilon,\lambda}[f](h_t)$$

4.3.3 Controlling constraints through Constrained Langevin Dynamic (CLD)

Let g a function (constraint) in the visible space that we want to keep fixed, say, to zero. We consider $\phi[g]$ and $\partial\phi[g]$ as defined in section 4.3.1. We propose to update h_{t+1} according to

$$h_{t+1} = h_t + \gamma \left[\mathcal{C}_{\epsilon,\lambda}[f](h_t) + b_t \, \hat{\partial\phi}[g](h_t) \right] \tag{21}$$

where b_t is chosen to ensure $\partial\phi[g](h_{t+1}) = 0$. To do so, using Taylor formula, we can approximate $\phi(h_{t+1})$:

$$\phi[g](h_{t+1}) \approx \hat{\phi}[g](h_t) + \partial\phi[g](h_t) \cdot (h_{t+1} - h_t) \tag{22}$$

$$\approx \hat{\phi}[g](h_t) + \gamma \, \partial\phi[g](h_t) \cdot [\mathcal{C}_{\epsilon,\lambda}[f](h_t) + b_t \, \hat{\partial\phi}[g](h_t)] \tag{23}$$

Imposing that $\phi[g](h_{t+1}) = 0$ we obtain:

$$b_t = \frac{1}{\|\partial\phi[g](h_t)\|_2^2} \left[\frac{1}{\gamma} \hat{\phi}[g](h_t) - \partial\phi[g](h_t) \cdot \mathcal{C}_{\epsilon,\lambda}[f](h_t) \right] \tag{24}$$

Finally, we can also choose dynamical criterion with function changing over time.

4.3.4 Objective and constraints function

Here we detail the function that we used as our criteria for sampling

- **Difference to reference sequence** (first batch)

  For the first batch we had difficulties finding sequences distant from the reference sequence due to strong constraints. This then became our main criterion to maximize

$$f(x) = \sum_i \mathbf{1}(x_i = x_i^0) \tag{25}$$

As for the control criteria, here are the details:

- **RBM score** (first and second batch)

  Another criterion we control is the RBM score $P(x)$ defined in section 4.2 which estimates how well the interactions between the residues in the protein are reproduced. Starting from the RBM energy of SpyCas9, e_0 we also target different intervals comprised between $e_{\min}$ and $e_{\max}$

$$f(x) = \left((e_{\max} - e_{\min}) + \frac{t}{100}e_{\min} - E_{\mathrm{RBM}}(x)\right)^+ + \left(E_{\mathrm{RBM}}(x) - \frac{t}{100}e_{\min}\right)^+ \tag{26}$$

- **Function prediction** (first batch)

  We also use as criterion the output of complex function in the visible space. We then choose to maintain the FoldX scores f previously defined below a certain threshold through a control criterion of the form:

$$f(x) = (f_a(x) - c)^+ \tag{27}$$

  where c is the maximal value we allow $f_a(x)$ to have. This option was dropped for the second batch.

- **Difference to reference sequence** (second batch)

  Difference to reference sequence was controlled starting with a linear transition from the SpyCas9 sequence x^0 to the expected distance, followed by a phase where this distance is kept fixed. For $t < 100$ and a final similarity in an interval comprised between $c_{\min}$ and $c_{\max}$ we have:

$$f(x) = \left((c_{\max} - c_{\min}) + \frac{t}{100}c_{\min} - \sum_i \mathbf{1}(x_i = x_i^0)\right)^+ + \left(\sum_i \mathbf{1}(x_i = x_i^0) - \frac{t}{100}c_{\min}\right)^+ \tag{28}$$

4.4 Experimental Validation

4.4.1 Cloning and bacterial growth conditions

PID variants were cloned into plasmid pWR5 depicted in the supplementary information (S4 file). pWR5 was constructed based on plasmid pFR56 [65] (GenBank: MT412099.1), a p15a plasmid with a chloramphenicol resistance gene carrying dCas9 expressed under the control of the DAPG inducible pPhlF promoter and a guide RNA under the control of a constitutive promoter. To obtain plasmid pWR5, the PID domain of dCas9 was replaced by a ccdB gene framed by BsaI restriction sites, enabling to easily swap in new PID sequences. In addition, an operon consisting of the mCherry gene followed by the sacB gene under the control of a constitutive promoter was added to the plasmid downstream of the dCas9 gene. Finally, the plasmid contains a guide RNA targeting the promoter of the mCherry-sacB operon. To test the functionality of PID with different PAM recognition sequences, we also constructed plasmid pWR6 which carries BsmBI restriction sites enabling to modify the sequence of the mCherry-sacB promoter and swap in target sites with different PAM motifs. Plasmids were cloned in *E. coli* DH10B, then introduced and characterized in *E. coli* MG1655.

4.4.2 Sucrose resistance and fluorescence assays

Resistance to sucrose was assessed by inoculating 1 mL cultures with individual colonies in 96 deep-well plates, refreshed 1:200 with inducer for 6h. 5 µL of serial dilutions were drop-plated

onto square LB-agar plates supplemented with 25 µM chloramphenicol, 50 µM DAPG and sucrose (0.125% or 0.25%) as appropriate, then grown overnight and imaged (S3 file). Fluorescence was measured by growing individual colonies in 96 deep-well plates overnight, refreshing 1:200-fold, growing 5 h then measuring the OD_{600} (9 nm bandwidth) and red fluorescence (ex. 580/9 nm, em. 620/20 nm) of 200 µL of culture in a black 96 well plate (Corning costar 3603) using a Tecan M200 plate reader. Unless noted, all PID variants were tested with a guide targeting a sequenced flanked by a standard NGG PAM (TGGGTTT). Fusions targeting non-NGG PAMs were tested with the same gRNA, but the PAM sequence adjacent to the gRNA target was modified to: TAGACTT (Seq1Cas9 and FmaCas9 PIDs), TAAATTT (TbaCas9 and sMacCas9 PIDs) or TTTAAAT (LrhCas9 PID). Sequences other than the PID and the 7-nt PAM were identical in all PID tests.

4.4.3 Phylogenetic tree of Cas9 PID chimeras

Cas9 sequences were obtained from [32], and whole Cas9 sequences or PID domains only were aligned using MUSCLE then maximum-likelihood phylogenetic trees were built using MEGA11 [66].

5 Supporting information

Data & Code The data and code used in this study can be accessed via the following link: `https://github.com/CyrilMa/DesignCas9WithCLD`

S1 file **Experimentally tested sequences**: This file contains the experimentally tested sequences in a zip format. It includes Fasta files, AlphaFold Prediction with PDB file, pLDDT, and more.

S2 file **Sequences with experimental and computational scores**: This Excel file contains sequences along with their corresponding experimental and computational scores.

S3 file **Additional plots on assessments of computational Score**: This PDF file contains additional plots that assess the computational score in combination to predict the activity of variants.

S4 file **Experimental measurements of the activity through both mCherry and sacB**: This PDF file contains experimental measurements of the activity through both mCherry and sacB.

S5 file **Additional Constrained Langevin Dynamics results with MNIST**: This PDF file contains additional results of constrained Langevin dynamics with MNIST including the building of transition paths with the method.

S6 file **Phylogenetic tree of all variants selected from Vink et al.**: This PNG file contains the phylogenetic tree of all variants selected from the study by Vink et al. [43]

Acknowledgements & Funding

SC and RM were supported by the Agence Nationale de la Recherche grant numbers ANR-17-CE30-0021 RBMPro and ANR-19-CE30-0021 Decrypted. CM is recipient of a PhD funding from AMX program, École polytechnique and benefits from financial support from the Centre de Recherche Interdisciplinary (CRI) through ”École Doctorale Frontières de l’Innovation en Recherche et Education – Programme Bettencourt”. DB, WR and FD were supported by European Research Council [677823], European Research Council [101044479], Agence Nationale de la Recherche [ANR-10-LABX-62-IBEID].

The authors declare no conflict of interest.

Author Contributions

Conceptualization Cyril Malbranke, David Bikard, Simona Cocco, Rémi Monasson

Data curation Cyril Malbranke

Investigation Cyril Malbranke, William Rostain, Florence Depardieu, David Bikard, Simona Cocco, Rémi Monasson

Methodology (computational) Cyril Malbranke, David Bikard, Simona Cocco, Rémi Monasson

Methodology (experimental) William Rostain, Florence Depardieu, David Bikard

Project administration David Bikard, Simona Cocco, Rémi Monasson

Ressources David Bikard, Simona Cocco, Rémi Monasson

Software Cyril Malbranke

Supervision David Bikard, Simona Cocco, Rémi Monasson

Visualization Cyril Malbranke, William Rostain

Writing Cyril Malbranke, William Rostain

References

1. Burley SK, Berman HM, Bhikadiya C, Bi C, Chen L, Di Costanzo L, et al. RCSB Protein Data Bank: Biological Macromolecular Structures Enabling Research and Education in Fundamental Biology, Biomedicine, Biotechnology and Energy. Nucleic Acids Research. 2019;47(D1):D464–D474. doi:10.1093/nar/gky1004. (Back to p.1)

2. Sagendorf JM, Berman HM, Rohs R. DNAproDB: An Interactive Tool for Structural Analysis of DNA–Protein Complexes. Nucleic Acids Research. 2017;45(W1):W89–W97. doi:10.1093/nar/gkx272. (Back to p.1)

3. Morcos F, Pagnani A, Lunt B, Bertolino A, Marks DS, Sander C, et al. Direct-Coupling Analysis of Residue Coevolution Captures Native Contacts across Many Protein Families. Proceedings of the National Academy of Sciences. 2011;108(49):E1293–E1301. doi:10.1073/pnas.1111471108. (Back to p.1)

4. Hopf T, Ingraham J, Poelwijk F, Schärfe C, Springer M, Sander C, et al. Mutation Effects Predicted from Sequence Co-Variation. Nature Biotechnology. 2017;35. doi:10.1038/nbt.3769. (Back to p.1)

5. Russ WP, Figliuzzi M, Stocker C, Barrat-Charlaix P, Socolich M, Kast P, et al. An Evolution-Based Model for Designing Chorismate Mutase Enzymes. Science. 2020;369(6502):440–445. doi:10.1126/science.aba3304. (Back to p.1)

6. Hawkins-Hooker A, Depardieu F, Baur S, Couairon G, Chen A, Bikard D. Generating Functional Protein Variants with Variational Autoencoders. PLOS Computational Biology. 2021;17(2):e1008736. doi:10.1371/journal.pcbi.1008736. (Back to p.1)

7. Ekeberg M, Hartonen T, Aurell E. Fast Pseudolikelihood Maximization for Direct-Coupling Analysis of Protein Structure from Many Homologous Amino-Acid Sequences. Journal of Computational Physics. 2014;276:341–356. doi:10.1016/j.jcp.2014.07.024. (Back to p.1)

8. Weigt M, White RA, Szurmant H, Hoch JA, Hwa T. Identification of Direct Residue Contacts in Protein–Protein Interaction by Message Passing. Proceedings of the National Academy of Sciences. 2009;106(1):67–72. doi:10.1073/pnas.0805923106. (Back to p.1)

9. Trinquier J, Uguzzoni G, Pagnani A, Zamponi F, Weigt M. Efficient Generative Modeling of Protein Sequences Using Simple Autoregressive Models. Nature Communications. 2021;12(1):5800. doi:10.1038/s41467-021-25756-4. (Back to p.1)

10. Tubiana J, Cocco S, Monasson R. Learning protein constitutive motifs from sequence data. Elife. 2019;8:e39397. (Back to p.2, 6, 15)

11. McGee F, Hauri S, Novinger Q, Vucetic S, Levy RM, Carnevale V, et al. The Generative Capacity of Probabilistic Protein Sequence Models. Nature Communications. 2021;12(1):6302. doi:10.1038/s41467-021-26529-9. (Back to p.2, 6)

12. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, et al. Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences. Proceedings of the National Academy of Sciences. 2021;118(15). (Back to p.2, 14)

13. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, et al. ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2021; p. 1–1. doi:10.1109/TPAMI.2021.3095381. (Back to p.2, 14)

14. Anishchenko I, Pellock SJ, Chidyausiku TM, Ramelot TA, Ovchinnikov S, Hao J, et al. De Novo Protein Design by Deep Network Hallucination. Nature. 2021;600(7889):547–552. doi:10.1038/s41586-021-04184-w. (Back to p.2)

15. Baldassarre F, Hurtado DM, Elofsson A, Azizpour H. GraphQA: Protein Model Quality Assessment Using Graph Convolutional Network. Submitted to Bioinformatics. 2020;. (Back to p.2)

16. Jing B, Eismann S, Soni PN, Dror RO. Equivariant Graph Neural Networks for 3D Macromolecular Structure; 2021. (Back to p.2)

17. Malbranke C, Bikard D, Cocco S, Monasson R. Improving Sequence-Based Modeling of Protein Families Using Secondary-Structure Quality Assessment. Bioinformatics. 2021;37(22):4083–4090. doi:10.1093/bioinformatics/btab442. (Back to p.2)

18. Leman JK, Weitzner BD, Lewis SM, Adolf-Bryfogle J, Alam N, Alford RF, et al. Macromolecular Modeling and Design in Rosetta: Recent Methods and Frameworks. Nature Methods. 2020;17(7):665–680. doi:10.1038/s41592-020-0848-2. (Back to p.2, 14)

19. Delgado J, Radusky LG, Cianferoni D, Serrano L. FoldX 5.0: Working with RNA, Small Molecules and a New Graphical Interface. Bioinformatics. 2019;35(20):4168–4169. doi:10.1093/bioinformatics/btz184. (Back to p.2, 14)

20. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly Accurate Protein Structure Prediction with AlphaFold. Nature. 2021;596(7873):583–589. doi:10.1038/s41586-021-03819-2. (Back to p.2, 14)

21. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, et al. Accurate prediction of protein structures and interactions using a three-track neural network. Science. 2021;373(6557):871–876. (Back to p.2, 14)

22. Moffat L, Greener JG, Jones DT. Using AlphaFold for Rapid and Accurate Fixed Backbone Protein Design; 2021. (Back to p.2)

23. Jendrusch M, Korbel JO, Sadiq SK. AlphaDesign: A de Novo Protein Design Framework Based on AlphaFold; 2021. (Back to p.2)

24. Gao Z, Tan C, Li SZ. AlphaDesign: A Graph Protein Design Method and Benchmark on AlphaFoldDB; 2022. (Back to p.2)

25. Larochelle H, Mandel M, Pascanu R, Bengio Y. Learning Algorithms for the Classification Restricted Boltzmann Machine. The Journal of Machine Learning Research. 2012;13:643–669. (Back to p.2, 5)

26. Bravi B, Tubiana J, Cocco S, Monasson R, Mora T, Walczak AM. RBM-MHC: a semi-supervised machine-learning method for sample-specific prediction of antigen presentation by HLA-I alleles. Cell systems. 2021;12(2):195–202. (Back to p.2, 5)

27. Shorthouse D, Hall MW, Hall BA. Computational Saturation Screen Reveals the Landscape of Mutations in Human Fumarate Hydratase. Journal of Chemical Information and Modeling. 2021;61(4):1970–1980. (Back to p.2)

28. Rahman MS, Islam MR, Alam ASMRU, Islam I, Hoque MN, Akter S, et al. Evolutionary Dynamics of SARS-CoV-2 Nucleocapsid Protein and Its Consequences. Journal of Medical Virology. 2021;93(4):2177–2195. doi:10.1002/jmv.26626. (Back to p.2)

29. Othman H, Bouslama Z, Brandenburg JT, da Rocha J, Hamdi Y, Ghedira K, et al. Interaction of the Spike Protein RBD from SARS-CoV-2 with ACE2: Similarity with SARS-CoV, Hot-Spot Analysis and Effect of the Receptor Polymorphism. Biochemical and Biophysical Research Communications. 2020;527(3):702–708. doi:10.1016/j.bbrc.2020.05.028. (Back to p.2)

30. Wang R, Wang S, Xu Y, Yu X. Enhancing the Thermostability of Rhizopus Chinensis Lipase by Rational Design and MD Simulations. International Journal of Biological Macromolecules. 2020;160:1189–1200. (Back to p.2)

31. Doudna JA, Charpentier E. The New Frontier of Genome Engineering with CRISPR-Cas9. Science. 2014;346(6213):1258096. doi:10.1126/science.1258096. (Back to p.2)

32. Collias D, Beisel CL. CRISPR Technologies and the Search for the PAM-free Nuclease. Nature Communications. 2021;12(1):555. doi:10.1038/s41467-020-20633-y. (Back to p.2, 4, 21)

33. Chatterjee P, Lee J, Nip L, Koseki SRT, Tysinger E, Sontheimer EJ, et al. A Cas9 with PAM Recognition for Adenine Dinucleotides. Nature Communications. 2020;11(1):2474. doi:10.1038/s41467-020-16117-8. (Back to p.3)

34. Walton RT, Christie KA, Whittaker MN, Kleinstiver BP. Unconstrained Genome Targeting with Near-PAMless Engineered CRISPR-Cas9 Variants. Science. 2020;368(6488):290–296. doi:10.1126/science.aba8853. (Back to p.3)

35. Nishimasu H, Shi X, Ishiguro S, Gao L, Hirano S, Okazaki S, et al. Engineered CRISPR-Cas9 Nuclease with Expanded Targeting Space. Science (New York, NY). 2018;361(6408):1259–1262. doi:10.1126/science.aas9129. (Back to p.3)

36. Nishimasu H, Ran FA, Hsu PD, Konermann S, Shehata SI, Dohmae N, et al. Crystal Structure of Cas9 in Complex with Guide RNA and Target DNA. Cell. 2014;156(5):935–949. doi:10.1016/j.cell.2014.02.001. (Back to p.3)

37. Ma D, Xu Z, Zhang Z, Chen X, Zeng X, Zhang Y, et al. Engineer Chimeric Cas9 to Expand PAM Recognition Based on Evolutionary Information. Nature Communications. 2019;10(1):560. doi:10.1038/s41467-019-08395-8. (Back to p.3)

38. Edraki A, Mir A, Ibraheim R, Gainetdinov I, Yoon Y, Song CQ, et al. A compact, high-accuracy Cas9 with a dinucleotide PAM for in vivo genome editing. Molecular cell. 2019;73(4):714–726. (Back to p.3)

39. Wei J, Hou L, Liu J, Wang Z, Gao S, Qi T, et al. Closely related type II-C Cas9 orthologs recognize diverse PAMs. eLife. 2022;11:e77825. doi:10.7554/eLife.77825. (Back to p.3)

40. Salakhutdinov R. Learning and Evaluating Boltzmann Machines. Utml Tr. 2008;2:21. (Back to p.4)

41. Tieleman T. Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient. In: Proceedings of the 25th International Conference on Machine Learning. ICML '08. Helsinki, Finland: Association for Computing Machinery; 2008. p. 1064–1071. (Back to p.4, 17)

42. Biswas S, Khimulya G, Alley EC, Esvelt KM, Church GM. Low-N protein engineering with data-efficient deep learning. Nature methods. 2021;18(4):389–396. (Back to p.5)

43. Vink JNA, Baijens JHL, Brouns SJJ. PAM-repeat Associations and Spacer Selection Preferences in Single and Co-Occurring CRISPR-Cas Systems. Genome Biology. 2021;22(1):281. doi:10.1186/s13059-021-02495-9. (Back to p.6, 15, 21)

44. Roussel C, Cocco S, Monasson R. Barriers and Dynamical Paths in Alternating Gibbs Sampling of Restricted Boltzmann Machines. Physical Review E. 2021;104(3):034109. (Back to p.7)

45. Kumar S, Paria B, Tsvetkov Y. Constrained Sampling from Language Models via Langevin Dynamics in Embedding Spaces; 2022. (Back to p.7)

46. Mirdita M, Schütze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M. ColabFold: Making Protein Folding Accessible to All. Nature Methods. 2022; p. 1–4. (Back to p.12)

47. Klausen MS, Jespersen MC, Nielsen H, Kjærgaard Jensen K, Jurtz VI, Sønderby CK, et al. NetSurfP-2.0: Improved Prediction of Protein Structural Features by Integrated Deep Learning. Bioinformatics; 2018. (Back to p.14)

48. Asgari E, Poerner N, McHardy AC, Mofrad MRK. DeepPrime2Sec: Deep Learning for Protein Secondary Structure Prediction from the Primary Sequences; 2019. (Back to p.14)

49. Wu Q, Peng Z, Zhang Y, Yang J. COACH-D: Improved Protein–Ligand Binding Sites Prediction with Refined Ligand-Binding Poses through Molecular Docking. Nucleic acids research. 2018;46(W1):W438–W442. (Back to p.14)

50. Hekkelman ML, de Vries I, Joosten RP, Perrakis A. AlphaFill: Enriching the AlphaFold Models with Ligands and Co-Factors; 2021. (Back to p.14)

51. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. BioRxiv. 2022;. (Back to p.14)

52. Salomon-Ferrer R, Case DA, Walker RC. An Overview of the Amber Biomolecular Simulation Package. WIREs Computational Molecular Science. 2013;3(2):198–210. doi:10.1002/wcms.1121. (Back to p.14)

53. Nijkamp E, Ruffolo J, Weinstein EN, Naik N, Madani A. Progen2: exploring the boundaries of protein language models. arXiv preprint arXiv:220613517. 2022;. (Back to p.14)

54. Rao RM, Liu J, Verkuil R, Meier J, Canny J, Abbeel P, et al. MSA transformer. In: International Conference on Machine Learning. PMLR; 2021. p. 8844–8856. (Back to p.14)

55. Elnaggar A, Essam H, Salah-Eldin W, Moustafa W, Elkerdawy M, Rochereau C, et al. Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling. bioRxiv. 2023; p. 2023–01. (Back to p.14)

56. Gligorijević V, Berenberg D, Ra S, Watkins A, Kelow S, Cho K, et al. Function-guided protein design by deep manifold sampling. bioRxiv. 2021; p. 2021–12. (Back to p.14)

57. Chan A, Madani A, Krause B, Naik N. Deep extrapolation for attribute-enhanced generation. Advances in Neural Information Processing Systems. 2021;34:14084–14096. (Back to p.14)

58. Madani A, Krause B, Greene ER, Subramanian S, Mohr BP, Holton JM, et al. Large language models generate functional protein sequences across diverse families. Nature Biotechnology. 2023; p. 1–8. (Back to p.14)

59. Malbranke C, Bikard D, Cocco S, Monasson R, Tubiana J. Machine learning for evolutionary-based and physics-inspired protein design: Current and future synergies. Current Opinion in Structural Biology. 2023;80:102571. (Back to p.14)

60. Dauparas J, Anishchenko I, Bennett N, Bai H, Ragotte RJ, Milles LF, et al. Robust deep learning–based protein sequence design using ProteinMPNN. Science. 2022;378(6615):49–56. (Back to p.15)

61. Krapp LF, Abriata LA, Cortés Rodriguez F, Dal Peraro M. PeSTo: parameter-free geometric deep learning for accurate prediction of protein binding interfaces. Nature Communications. 2023;14(1):2175. (Back to p.15)

62. Hauser M, Steinegger M, Söding J. MMseqs Software Suite for Fast and Deep Clustering and Searching of Large Protein Sequence Sets. Bioinformatics. 2016;32(9):1323–1330. doi:10.1093/bioinformatics/btw006. (Back to p.15)

63. Glorot X, Bengio Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings; 2010. p. 249–256. (Back to p.16)

64. Loshchilov I, Hutter F. Fixing Weight Decay Regularization in Adam. open review. 2018;. (Back to p.17)

65. Rousset F, Cabezas-Caballero J, Piastra-Facon F, Fernández-Rodríguez J, Clermont O, Denamur E, et al. The impact of genetic diversity on gene essentiality within the Escherichia coli species. Nature microbiology. 2021;6(3):301–312. (Back to p.20)

66. Tamura K, Stecher G, Kumar S. MEGA11: Molecular Evolutionary Genetics Analysis Version 11. Molecular biology and evolution. 2021;38(7):3022–3027. (Back to p.21)