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 combines evolutionary and physics-grounded modeling. Using a Restricted Boltzmann Machine, we learned a sequence model of a protein family and 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 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. Interestingly, some sequences (6/71) produced by our method showed an improved activity in comparison with the original wild-type proteins sequence. These results demonstrate the interest of 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 sometimes distant 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]. Evolution-based models also struggle to access mutations and combinations of mutations beyond those explored by evolution.

Other considerable efforts were devoted these last years to the modeling of the protein structures. In contrast to evolutionary based methods that rely solely on sequence information [12, 13], structural modeling of protein families incorporates physical and chemical considerations to tackle the lack of available data. Structural modeling approaches have been used to perform multiple important tasks in protein design: inference of structure from a sequence, inference of a sequence from a structure (for example backbone design with [14]), quality assessment of a structure [15–17] or protein interaction with ligands (other proteins, RNA, DNA, metal . . .). For the last two decades, most successful approaches have relied on empirical force fields such as Rosetta [18] or [19], frameworks that encompass multiple physicochemical properties to evaluate and optimize protein tertiary structure. 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. In particular, designing functional proteins interacting with ligands or other proteins requires the relaxation of the protein static structure model. Many protein functions require conformational changes and therefore need to react to perturbation in their environment and contact with ligands.

Beyond what can be learned from natural sequence variation and structural modeling, experimental data on protein activity and specificity is sometimes available that could help in 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 experimental 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.

We evaluate the strategy on the design of Cas9 variants. The Cas9 protein is an RNA guided nuclease from the prokaryotic immune system known as CRISPR-Cas9 [27]. 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 then two guanines (NGG). The PAM requirement limits the 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 [28] for an overview), and engineer Cas9 variants with expanded recognition capabilities [29–31]. It was notably shown to be possible to swap the PID from SpCas9 with the PID of other natural variants [32, 33], suggesting that the PAM-Interacting domain is relatively independent of the rest of the protein. We then choose to model the PID only, without integrating into the design the rest of the enzyme.

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 experimental

data 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

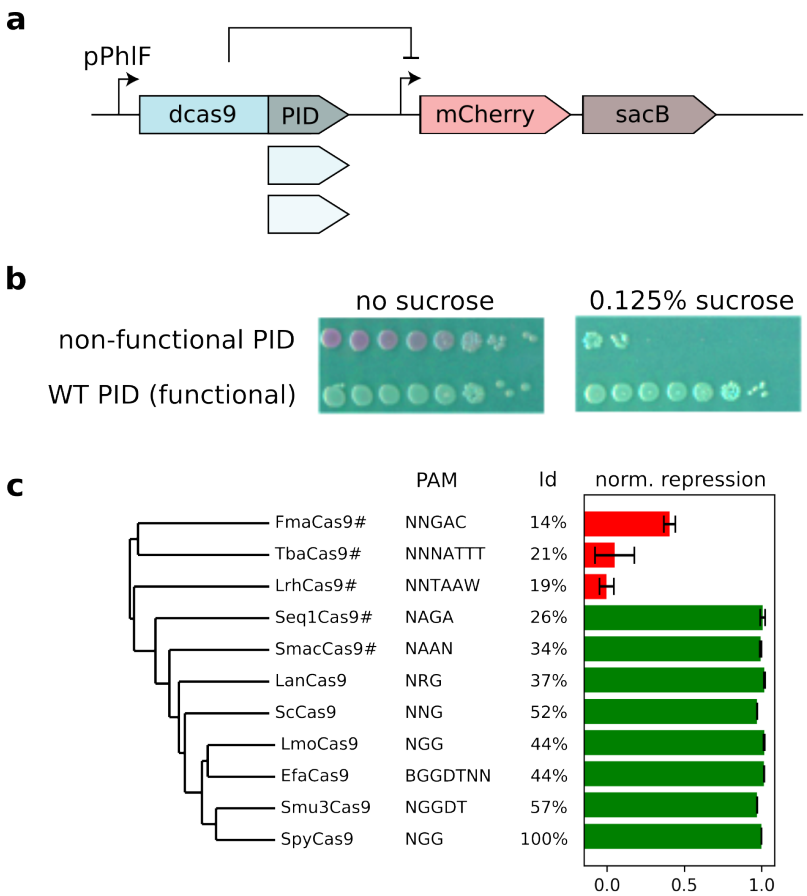

Fig 1. Experimental validation of chimeras between SpyCas9 and the PAM-interacting domain of other natural variants
a: Genetic circuit used to test Cas9 PID variants. **b**: Spot assay of the wild-type SpyCas9 and a negative control PID variant in the circuit, on plates containing the inducer DAPG (left) or DAPG and 0.125% sucrose (right). **c**: Maximum-likelihood tree of alignment of PID sequences of the tested Cas9 PID chimeras. We also display the normalized repression of Cas9 chimera (SpydCas9 + variant PID) targeting a motif recognized by the PID. The experimentally determined PAM are taken from Collias et Beisel (2021) [28] review, and we flagged with a # the PIDs which did not recognize TGG and were therefore tested with another PAM. As we can see, variants who are phylogenetically closer to SpyCas9 PID are functional (in green) while the more distant are not.

Although previous research with Cas9 PID fusions showed there is a relative independence between this domain and the rest of the protein, only a few fusions exist in the literature. To

confirm this modularity and the assumption that the PID can be modelled independently of the rest of the protein, we constructed a system to rapidly characterize Cas9 chimeras. 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-selectable marker *sacB* (see Fig 1**a**). Cells expressing a non-functional variant of dCas9 will yield red fluorescent colonies, and die on plates containing 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**). 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, only the former was moderately active and the two others inactive. 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. For example, LanCas9, the most distant from SpyCas9 functional variant, had 46.7% identity with SpyCas9 PID.

2.2 Sampling proteins under constraints: Langevin Dynamic in Restricted Boltzmann Machine Representation Space

2.2.1 Restricted Boltzmann Machine for protein sequence-based modeling

RBM are two-layer probabilistic graphical models (see 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 [34]. 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 [35] to sample approximately but more efficiently from the model distribution.

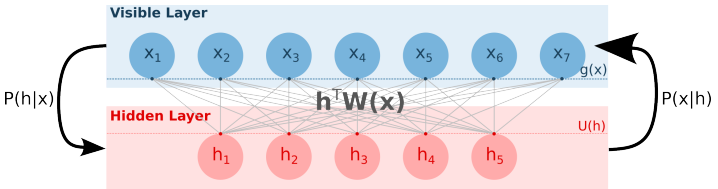

Fig 2. Schema of a standard Restricted Boltzmann Machine
The RBM is a probabilistic graphical model with two layers. The graph is not oriented, allowing to sample from the visible layer to the representation layer and back.

2.2.2 Constrained Langevin Dynamics

RBMs are powerful but relatively light models that have a limited capacity to capture complex and high order interactions. Nonetheless, they are still very well suited to modeling datasets of small size. In particular, RBMs deploy an interesting representation space with proteins with similar properties, structure, sequence, or functionality clustering together [10]. When the quantity and quality of raw data are low, even models of higher complexity are limited by the number of available sequences, as shown recently by McGee et al. [11]. Only $\sim$1070 sequences are found in the SpyCas9 Pam Interacting domain family (PF16595), and the alignment between them is far from perfect.

On the other hand, other modeling frameworks that rely on physical principles can provide useful information (protein stability, binding affinity, etc.) without large datasets. We explore here a novel strategy based on a Langevin dynamic process that enables direct sampling

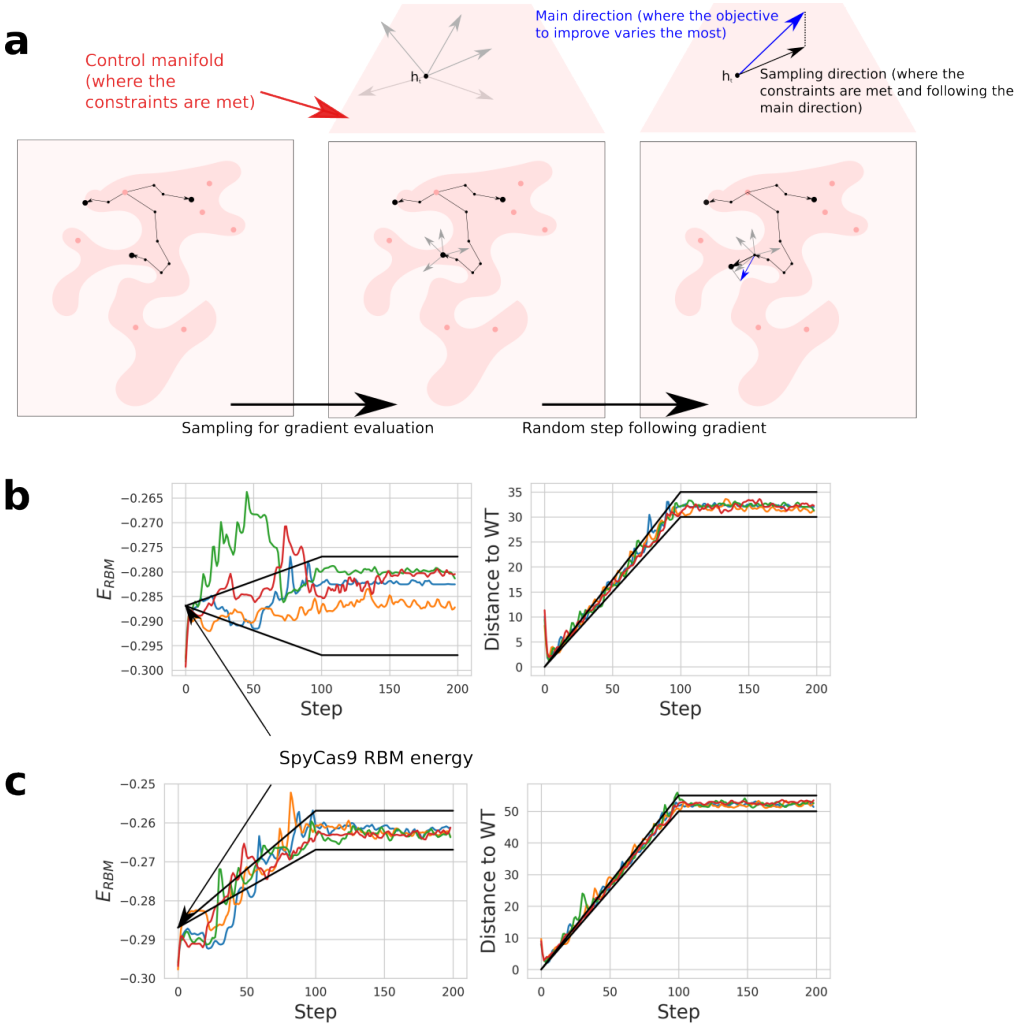

Fig 3. Constrained Langevin Dynamics as a sampling method for Restricted Boltzmann Machine
a: The Constrained Langevin Dynamics in the representation space consists in two steps. The first step is a round of sampling to evaluate the gradient of the main criterion and the gradient of the control criterion through the expectation formula (see methods), the second is a random step following the direction drawn by these two gradient (projection of the main direction vector on the orthogonal of the control direction). A Brownian noise is added to create randomness and diversity in the samples. **b** and **c**: Typical Random Walks for designing sequences with between different RBM energy and similarity targets

according to external inputs 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 doesn't guarantee that the full distribution of sequences will be sampled (see Roussel et al. [36]) and this approach does not easily unlock the possibility of conditional sampling of sequences.

We propose another sampling strategy (see Fig 3**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 use a simple operator that allows to embed in the representation space any function f that applies to the input data (the protein sequence) of the RBM in the representation space. In the case of a protein sequence, such function might be a predictor of biochemical properties or of structural similarity. We use:

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

This function and its gradient are then easy to estimate through sampling (see methods in section 4.3)

This open the way to use a generalized Langevin dynamic [37], to both optimize the objective under constraints and create diversity. Step by step, we perform a walk in the representation space by following the gradient and a Brownian perturbation.

$$h_{t+1} = h_t + \alpha \vec{\nabla}_h \phi(f)(ht) + \epsilon \mathcal{N}(0, I)$$

We also use the embedding operator on a set of criterion to maintain. The embedded functions are then a support to project back our samples into the space where the constraints are met.

$$h_{t+1} = \text{Proj}\left[h_t + \alpha \vec{\nabla}_h \phi(f)(ht) + \epsilon \mathcal{N}(0, I)\right]$$

More details about the method can be found in section 4.3.

To illustrate the control of Constrained Langevin Dynamics through RBM, we tried to perform random walks targeting different RBM energies and Hamming distance to the reference SpyCas9 PID, as we did to propose sequences verified experimentally in section 2.5. Constrained Langevin Dynamic linearly drives the sequences to a target similarity and energy. In Fig 3**b** and Fig 3**c**, we show two examples of random walks with 5 different chains. We also display in supplementary 4 some additional results on MNIST.

2.3 Structural Quality Assessment of sequences with FoldX and AlphaFold2

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. Taking in input a protein sequence with its structure. FoldX can repair and optimize the side chains of each amino acids to evaluate protein stability based on backbones angles, hydrogen bonds, van der Wall interactions and more. Using an experimentally or computationally determined structure file, we can score a sequence related to these scores. FoldX was recently successful in various tasks, including the prediction of the effect of mutations [38–40] or protein variant design [41].

We retrieved from PDB a structure (PDB-ID : 5y36) of the Cas9-PID in complex with DNA. Using FoldX, we replace amino-acids in the structure by the sequences to evaluate, then we repair the structures to fix the side chains and compute the FoldX energy. Functional sequences are expected to have a FoldX energy close to that of the WT sequence.

We also used AlphaFold2 to assess the quality of the structure. For a few selected sequences, we predicted structures using ColabFold [42]. For each sequence, the structure obtained was then aligned to one of the SpyCas9 PID (predicted in the same manner). We then computed the Root Mean Square Deviation (RMSD) between the two aligned structures as a measure of how well-preserved the tertiary structure of the domain was.

To conclude, we then run our analysis with 2 structural metrics : the FoldX score of the complex PID-DNA (FoldX DNA) and generated AlphaFold structure (AF2 RMSD) the TM-score from the alignment of natural and predicted LDDT score from AlphaFold (pLDDT) were also tested but did not bring improvements to the predictive performances.

2.4 Introducing supervision in Restricted Boltzmann Machine

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

The standard RBM is an unsupervised model that does not require any other information than the amino-acids sequence input. When available, experimental data on protein properties can greatly enhance the ability of models to capture meaningful information (see Biswas et al. [43]). 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. We incorporate a Neural Network C_θ to the RBM model, which receives as input the hidden representation of an RBM and outputs a prediction of the supervision data (see Fig 4**a**).

For labelled data, the energy of the RBM is changed to incorporate a cross-entropy loss that is not used for unlabeled samples. This approach is similar to the one proposed by Larochelle et al. 2012 [25] :

$$E(x, y, h) = E_{\text{RBM}}(x, h) + \gamma \text{CE}(y, C_\theta(h)) \tag{1}$$

We have to weight the cross entropy loss (CE) and the RBM loss to find an equilibrium between modeling of the data (minimization of the log likelihood of the RBM) and performance of the classifier. If γ is small, the model will converge to a standard RBM if γ is big, the model will start to 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.

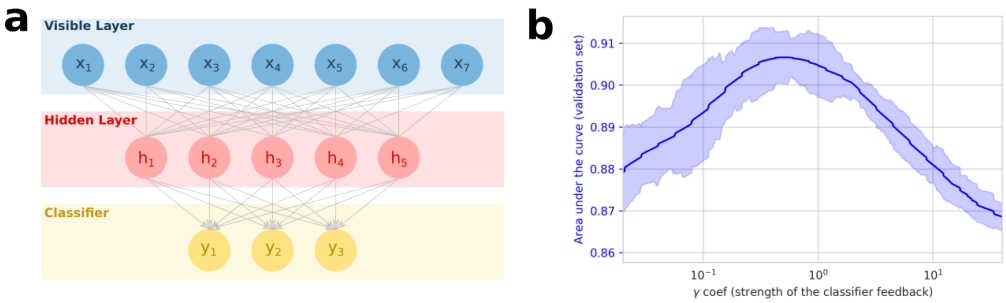

Fig 4. Semi-supervision helps the RBM learn useful representations of the Cas9 PID **a**: Our Semi-Supervised Learning RBM with a one-layer classifier. This classifier is oriented and take as input the representation of the sequence in the hidden layer. **b**: Area under the ROC curve for the prediction of the PAM on the validation set. The curve was smoothed, and we display the standard deviation over 20 consecutive values of γ. Balance between the classifier and the RBM seem to both stabilize the training and improve the accuracy of the classifier.

2.4.2 Semi-supervision helps the RBM learn useful representations of the Cas9 PID

We evaluated this strategy by training an RBM on the 1071 available sequences for the Cas9 PID (PF16595 PFAM family). As supervision data, we use a dataset of PAM recognition motifs predicted from genomics data and available for 154 PIDs [44]. We then evaluate the predictive capabilities of our model on an held-out validation set. We also tested its generative capacities

on the more challenging task of generating novel variants of Cas9 PID using Constrained Langevin Dynamics. We assessed the stability of the proposed variants using FoldX.

We used MMseq2 to cluster the sequences with a threshold of 70% similarity, and randomly extracted a set of clusters corresponding to $\tilde{1}5\%$ of the sequences as our validation set. We trained 600 models with progressive γ (from 10^{-6} to 10^{6}). For labelled data, the positions in the PAM motif are modelled independently by using as a target a binary vector encoding the recognized motifs (see methods). 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 was made of 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, we computed the area under the ROC curve (AUC) for the predictions on the validation set. We then reduced it to a single mean AUC by averaging over all position and nucleotide. We show in Fig 4**b** this mean AUC over γ. As we see, we reach a maximum for intermediate value of γ that outperform 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.

We also proposed to test the generative capacities of Semi-Supervised Learning Restricted Boltzmann Machine (SSL-RBM). To do so, we used Constrained Langevin Dynamics to generate 120 protein sequences per RBM with an RBM energy equivalent to the reference sequence (SpyCas9) and different number of mutations, spanning from 10 to 50. We then used FoldX (with the SpyCas9-PID/DNA) complex to get a fast evaluation of the quality of the sequences generated (see Fig 5**a**). Sequences were generated in this manner using values of γ spanning from 10^{-2} to 10^{2}.

We show the distribution of FoldX DNA energy below certain threshold on Fig 5**b** as well as some statistics of these distributions (FoldX energy of the top 5%, 10%, 25% as well as the median. We observe a balanced regime with intermediate value of γ that outperforms both the regime with weak feedback (small values of γ) or the classifier regime with strong feedback (high values of γ). This suggests that the use of classification data in a semi-supervised way improves the generative capacities of the model, since generated samples have a better structure according to FoldX.

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 assess the interest of our method, we designed several sequences through Langevin Dynamics, targeting different similarity to SpyCas9, different RBM score and different protein-DNA complex stability according to their FoldX energy.

Overall, we adopted two methods for sampling :

- The first method (and batch) was to improve the distance to SpyCas9 WT (in terms of number of mutations) 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) was to constrain the RBM score and the similarity to SpyCas9 within a certain interval using Langevin Dynamics, and then to select sequences with good FoldX score *a posteriori* (sequences referenced dgfx30 to dgfx75).

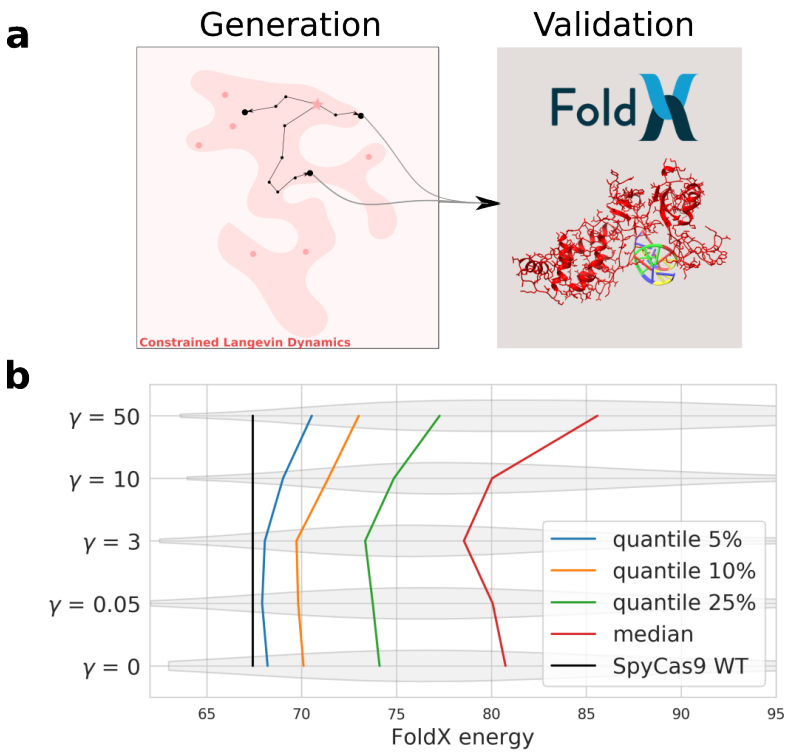

Fig 5. Generative capacities of Semi-Supervised Learning Restricted Boltzmann Machine **a**: We test the generative capacities of our models by generating 120 sequences with Constrained Langevin Dynamics using the trained Restricted Boltzmann Machine. We then use FoldX to compute the energy of the protein-DNA complex for the generated sequences and evaluate their quality. **b**: For different value of γ, we display the distribution of FoldX energy of the generated protein sequences through Constrained Langevin Dynamic using the SSL-RBMs trained with given value of γ. We also the share of sequences generated that had a FoldX energy below certain thresholds. Intermediate values of γ have the best ratio of sequence generated and consistently outperform generation with a standard Restricted Boltzmann Machine: 16% of the sequences generated with a good SSL-RBM had a FoldX energy below 70 (equivalent to the energy of the wild type) against less than 10% with standard RBM.

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 Fig 5**b**).

We selected a total of 71 sequences that spanned different predicted scores to be assessed experimentally (29 for method 1 and 42 for method 2). DNA sequences encoding the desired proteins were synthesized and introduced in E. coli, where we evaluated their ability to bind a target position with a NGG PAM motif and silence the expression of a mCherry reporter gene as described in section 4.4 (see methods). The activity of dCas9 is reported as the logarithm of the repression fold compared to an inactive dCas9 with no PID (f_0) and normalized to the WT dCas9 (f_1) :

$$\text{normalized repression}(f) = \text{nr}(f) = \frac{\log f/f_0}{\log f_1/f_0} \tag{2}$$

An inactive sample will have a score of 0 and an active sample will have a score closed to 1

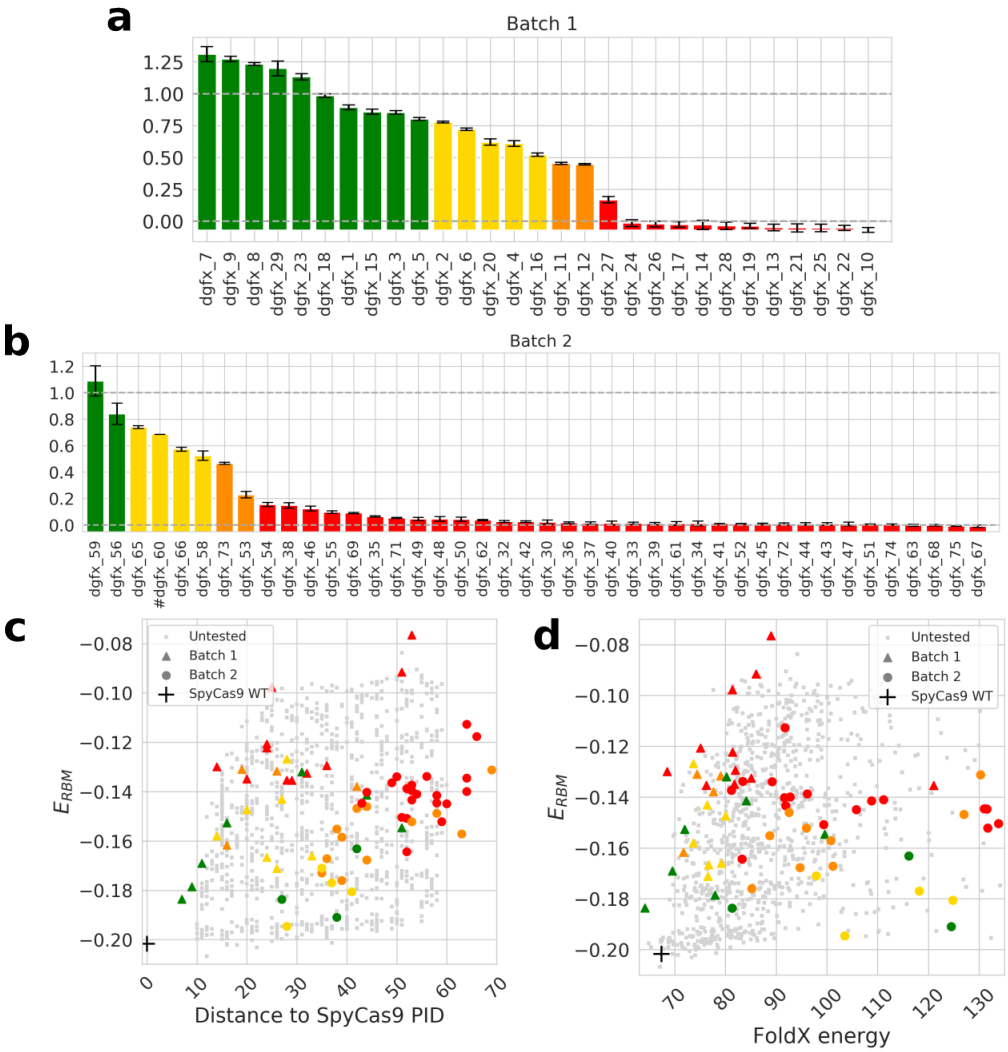

Fig 6. Experimental evaluation of generated sequences and relationship with Hamming distance, RBM and FoldX scores
a and **b**: The activity of the dCas9 chimera carrying generated PIDs was measured using the mCherry fluorescence assay. Arbitrary thresholds were used to classify proteins as very functional (green), functional (yellow), marginally functional (orange), not functional (red). Barplot **a** represent the sequence in the first batch of tested sequences and bar plot **b** represent the sequences in the second batch of tested sequences. Error bars are plotted using 2 σ, where σ in the standard deviation between independent measurement (see methods) (# dgfx_60 large standard deviation obtained from experimental measurement was removed from the figure for the sake of 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.

if it is as active as the WT Cas9 PID. In Fig 6**a** and Fig 6**b** we display the normalized repression of the sequences in batch 1 and 2 experimentally tested. For visualization, we define four levels of activity : not functional (nr < 0.2), marginally functional (nr < 0.5), functional (nr < 0.8), very functional (nr > 0.8). 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 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. The second batch is composed of sequences designed with relaxed objectives of distance to SpyCas9 PID, RBM and FoldX energy compared to the first batch (see below). 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). The drop in the share of functional variants is a reflection of these relaxed constraints, and cannot be interpreted as a lower performance of the second method compared to the first (which consisted in postfiltering using FoldX instead of performing a FoldX computation at every Langevin Dynamics Step).

Experimental results are displayed in supplementary 3. 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 on the first figure, 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 the Fig 6**d** (good FoldX score, good RBM score) displayed a high proportion of active or very active sequences. We also display non experimentally tested sequences generated by Constrained Langevin Dynamics (with different target in Hamming distance and RBM energy) in gray. We notice again that a substantial part of the sequences generated display a good FoldX score, which motivated the use of the score as a post-processing step rather than a constraint during Langevin Dynamic sampling.

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

We then investigated if semi-supervision with the PAM information improved the ability of our RBM to predict the activity of Cas9 variants. 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 8) that the correlation between the RBM score and Cas9 activity improve when unsupervised and supervised loss are balanced (with values of γ between 1 and 50). The optimal value of γ falls in the same range as its optimal value for the classification of PAM recognition sequences (see Fig 4**b**), providing further evidence that semi-supervision improved both the ability of the RBM to separate Cas9 PIDs according to their recognition sequence, and the ability of the RBM to recognize functional sequences. This is interesting considering that the activity of Cas9 variants here was only evaluated on the canonical NGG PAM motif, while the supervision data was the nature of the PAM motif recognized by a small subset of our training data. This suggests that the additional information provided to the model in this manner generally improved its performance, even if the information did not relate to activity prediction directly.

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 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 [42]) and compared this structure to the one predicted by the same AlphaFold model for the wild-type.

On Fig 7**a**, we display the correlation with the structural metrics and the RBM score. We

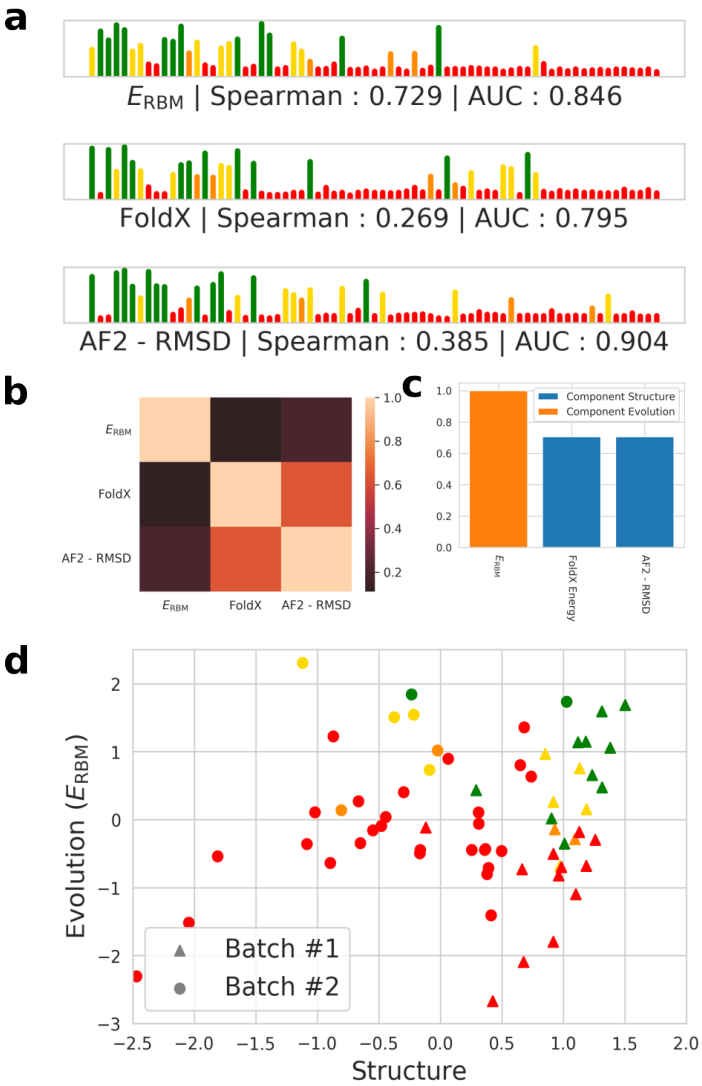

Fig 7. Evolution and structure scores predict sequence activity
a: Sequences ordered by their score according to each of the metrics, the height of the bars is the activity measured by mCherry. The RBM score displayed the best Spearman correlation, but is beaten by AlphaFold in terms of detection of very functional samples (AUROC curve for detecting sample with activity over 0.8). **b**: Inter correlation between the different metric, the FoldX energy and the AlphaFold score, are very correlated as we can. **c**: The strong correlation translates into two components using a sparse PCA: the evolution and the structure component. **d**: On the bottom, we displayed the two components inferred by sparse PCA. We labelled them Evolution and Structure component, and we display the sequences colored by their activity. This representation separates well active and inactive samples. The correlation matrix shows that the structural scores are very correlated but almost orthogonal to the RBM score.

also computed the Area Under the ROC Curve (AUROC) for detecting 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. In terms of detecting very functional samples (mCherry activity over 0.8), AlphaFold displayed the best results, but the RBM on its own still performed very well and seemed to capture complementary information as

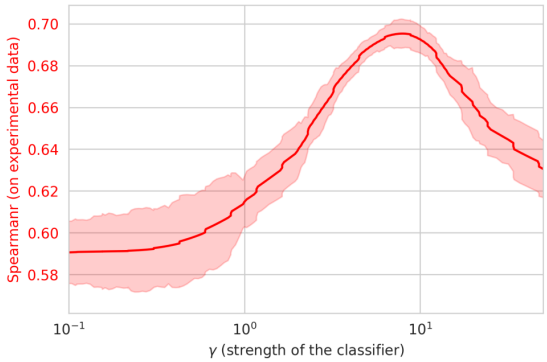

Fig 8. Correlation between SSL-RBM energy and experimental activity
The correlation between the experimental activity and the SSL-RBM score is plotted as a function of the classifier strength (γ) (red curve). The curve was smoothed, and we display the standard deviation over 20 consecutive values of γ. Note that the values of γ that yielded the best performing models with regard to experimental data fall in the same range as the values of γ that yielded the best performing models when evaluating the accuracy of PAM prediction on a validation set.

	$\mathbf{E_{RBM} < \infty}$	$\mathbf{E_{RBM} < -0.23}$	$\mathbf{E_{RBM} < -0.20}$	$\mathbf{E_{RBM} < -0.18}$
$\mathbf{fx < inf}$	0.17 [12/71]	0.18 [12/67]	0.22 [12/54]	0.40 [10/25]
$\mathbf{fx < 90}$	0.29 [10/34]	0.32 [10/31]	0.43 [10/23]	0.62 [08/13]
$\mathbf{fx < 85}$	0.36 [10/28]	0.37 [10/27]	0.48 [10/21]	0.73 [08/11]
$\mathbf{fx < 80}$	0.41 [07/17]	0.41 [07/17]	0.54 [07/13]	0.78 [07/09]
$\mathbf{fx < 78}$	0.47 [07/15]	0.47 [07/15]	0.64 [07/11]	0.88 [07/08]

Table 1. The combination of structural and RBM scores best discriminates functional sequences
Since the FoldX score in complex is among the most efficient, we displayed the share (and number) of very functional sequences with different thresholds of the RBM energy and the FoldX score. For a fixed RBM threshold, we see that the share of active sequences decrease when we relax the threshold of for the FoldX energy score and the same goes for a fixed FoldX threshold

can be seen in the failure of AlphaFold to spot inactive samples which are correctly predicted as active by the RBM and vice-versa. To push this analysis a bit further, we noticed that AlphaFold and FoldX scores were highly correlated (Fig 7**b**) which motivate the use of a Sparse PCA on the different scores we collected to reveal two components on the vectors of the different scores associated to each sequence (Fig 7**c**). What emerged from this Sparse PCA was one component encompassing the structural scores from FoldX and AlphaFold and one component corresponding to the RBM score alone (Fig 7**d**). This matches the observation we made that the structural scores are strongly correlated, while the RBM brings another view of the data.

In Table 1, we show a numerical confirmation of this complementarity by displaying the share of very functional samples for different thresholds of RBM and FoldX energy. Hard threshold on both FoldX and RBM provide an important success rate, we also notice that for a fixed RBM energy threshold the success rate was improving when we were hardening the threshold on the FoldX energy and the same goes when we choose a more severe RBM threshold with a fix FoldX energy threshold.

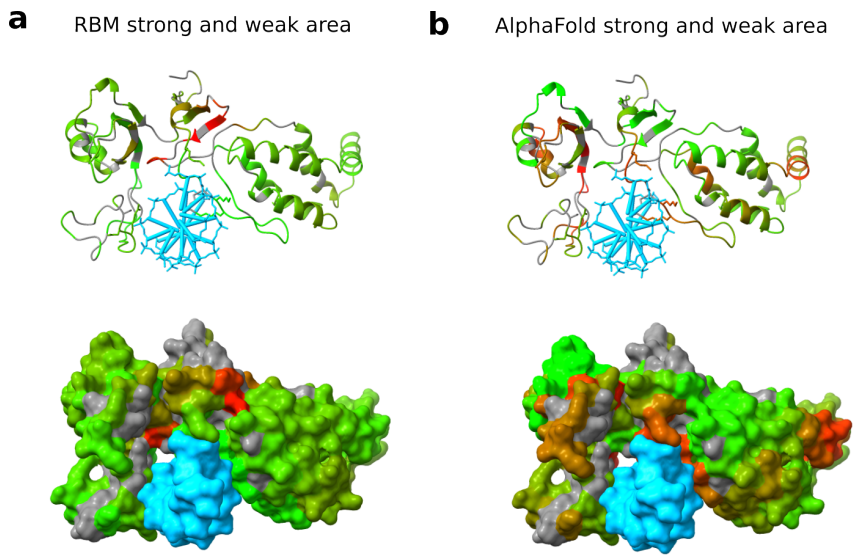

Fig 9. Strength and weaknesses of AlphaFold and the RBM models
We display the share of sequences correctly predicted as non-functional by either the RBM energy (**a**) or the AlphaFold RMSD (**b**) as a function of the localization of the mutations on the protein sequence and structure. DNA is shown in blue. In green are the area where the deleterious effect of mutations are correctly predicted, while in red are the area 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, while the RBM struggle more than AlphaFold on internal parts of the structure.

2.5.4 Weak and strong spots of RBM and structural scores

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 than AlphaFold and conversely. For every residue on the protein sequences, we gathered all the experimentally tested sequences with a mutation at the position and that were inactive (normalized repression below 0.5). We then checked whether the RBM energy and the AlphaFold RMSD could be used to correctly predict that the variant was inactive. For each position, 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 on the protein structure (Fig 9**b**).

One of the key observation 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 take into account interactions with other molecular partners, such as DNA.

3 Discussions

Structural biology has been a field of major breakthrough these last years with the increasing success of Deep Learning models for secondary structure prediction [45,46], ligand or protein binding site prediction [47,48] and of course tertiary structure prediction with frameworks such as AlphaFold2 [20], RoseTTAFold [21] or ESMFold [49]. Alternatively, frameworks like Rosetta [18], Amber [50] or FoldX [19] have been developed around an empirical force field to

encompass molecular and atomistic interactions. These frameworks rely on strong physico-chemical priors made 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 structure are available since their determination is complex and expensive. Force fields can evaluate the compatibility between an amino acids sequence and a given structure and are generally more robust to over-fitting that data-based models.

On the other end, 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, 51–53]. 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 [54, 55]. We proposed 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 [56]

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 despite their great modeling of the crystal structure of the PID respectively alone and in complex with the DNA strands 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 they are 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 improve 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. Some of the chimeras that we constructed using the PID of natural Cas9 variants already showed a stronger activity than SpyCas9, and it might thus be expected that our model explored regions of the sequence space with similarly improved activity. It is also possible 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.

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

Two families of PAM-Interacting domains are available in PFAM ([57]), the PFAM family which contains the SpyCas9 PAM-interacting domain (PF16595) also contains 1102 other sequences (with no annotation of recognized PAM).

For training of our model SSL-RBM, we rely mainly on a dataset from a genomic approach. Vink et al. [44] retrieved the spacers available in CRISPRCasDb ([58]) following a Cas gene to map them with their target in bacteriophage genome. Recognized motifs can be read at the flank of the corresponding target (protospacer). This dataset provides high-throughput data with lower cost than the experimental determination of the accepted PAM. We crossed the data with the corresponding Cas9 genes in the genomes where the motifs were retrieved. With this method, we were able to identify 154 sequences corresponding to the PF16595 PFAM family.

4.1.2 Data engineering

We used data from PF16595 PFAM family (unlabeled) and Vink et al. dataset (labelled) for training and validation of our SSL-RBM. To build a validation set, datasets were grouped together and stratified through MMSEQS2 [59]. We use MMSEQS to cluster sequences that had more than 90% similarities. Clusters were randomly extracted one by one to build a validation of around 10% of the total set, ensuring that sequences in both sets labelled or unlabeled were not too similar. Sequences were also weighted given the size of the cluster they belong, to not give too much importance to data too similar. Concretely, let x be a sequence and $\mathcal{C}(x)$ its cluster built through MMSEQS2 then the sequence was ponderated according to $\frac{1}{\text{card } \mathcal{C}(x)}$

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 + gaps. The hidden layer has 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 ([60]), following the distribution $\mathcal{U}(-\frac{1}{\sqrt{qn_x}}, \frac{1}{\sqrt{qn_x}})$ where $n_x = 736$ is the size of the sequence 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 represent the prevalence of different amino-acids at each position. The hidden layer contains a fixed and arbitrary number of hidden units and associated with a potential $U(h)$. These two layers are linked by a matrix W_{xh} of size $n_x \times n_h$ such that :

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

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

One of the interest of the choice of this potential is to facilitate the sampling of the visible layer from the hidden layer, as well as the reverse process.

With this formalism, we have for a Gaussian potential :

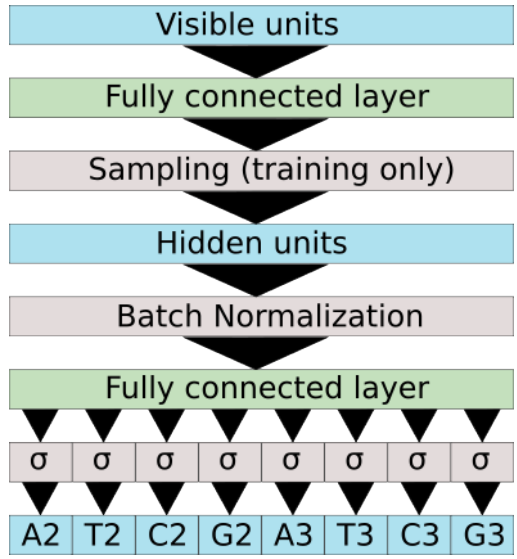

Fig 10. Architecture of the SSL-RBM classifier, in training the classifier take in input the hidden layer after sampling (equivalent to a noisy representation of the input sequence) while in prediction it takes the hidden representation. The hidden representation is then batch normalized, then is fed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

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

Which can be simulated with a Gaussian Law $\mathcal{N}(\left(\sum_i W_{i\mu}^{xh}(x_i)\right)_\mu, I)$

and

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

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 in inputs the output of the hidden layer (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 is positive for the quality of the training.

The classifier (Fig 10) was made to predicted PAM recognized by the classifier. 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 training of RBM is done through the maximization of the likelihood of the training set T.

Let θ be the set trainable parameters we want to optimize. We denote by $p(x|\theta) = \log P_\theta(x) = \log \frac{1}{Z(\theta)} \exp\left[-E(\theta, x)\right] = -E(\theta, x) - \log Z(\theta)$ the log-likelihood of a

sequence x given the set of parameters θ. We perform maximum likelihood estimation on the training set to optimize the parameters. It is to say we want to find :

$$\theta^* = \arg\max_{\theta} \sum_{x \in T} p(\theta, x)$$

One way to do this is to perform Gradient Descent. We have to estimate $\mathbb{E}_{x\sim T}\left[\frac{\partial p(\theta,x)}{\partial\theta}\right] = -\mathbb{E}_{x\sim T}\left[\frac{\partial E(\theta,x)}{\partial\theta}\right] - \frac{\partial \log Z(\theta)}{\partial\theta}$. It is easy to approximate the first term with mini batches of samples from the training set T. The other term, however, is trickier :

$$\begin{aligned}\frac{\partial \log Z(\theta)}{\partial\theta} &= \frac{\frac{\partial Z(\theta)}{\partial\theta}}{Z(\theta)} = \frac{\sum_x \frac{\partial \exp -E(\theta,x)}{\partial\theta}}{Z(\theta)} \\ &= -\frac{\sum_x \frac{\partial E(\theta,x)}{\partial\theta} \exp E(\theta,x)}{Z(\theta)} = -\sum_x \frac{\partial E(\theta,x)}{\partial\theta} P_\theta(x)\end{aligned}$$

With P_θ the distribution of the RBM model, M we then have :

$$\mathbb{E}_{x\sim T}\left[\frac{\partial p(\theta,x)}{\partial\theta}\right] = \mathbb{E}_{x\sim M}\left[\frac{\partial E(\theta,x)}{\partial\theta}\right] - \mathbb{E}_{x\sim T}\left[\frac{\partial E(\theta,x)}{\partial\theta}\right]$$

To approximate the expectation of the model, we then have to sample from the training set (easy) and from the model. This is complicated, and for this, we rely mainly on **Persistent Contrastive Divergence** [35]. We implemented our own RBM in PyTorch (available on https://github.com/CyrilMa/torchpgm-standalone) running on GPU, we rely on the AdamW [61] optimizer as it yielded a more robust training.

4.3 Sampling RBM through Constrained Langevin Dynamic

4.3.1 Function embedding in RBM representation space

Let f be an evaluation function applying in the visible space, it is to say on a sequence x. We are interested in the optimization of this function in the hidden space. To achieve that, 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 show that :

$$\begin{aligned}\vec{\nabla}_h \mathbb{E}_{x|h}(f(x)) &= \vec{\nabla}_h \sum_x P(x|h) f(x) \\ &= \sum_x \vec{\nabla}_h P(x|h) f(x) \\ &= \sum_x f(x) P(x|h) \vec{\nabla}_h \log P(x|h)\end{aligned}$$

And,

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

We conclude that :

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

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

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

Let f be a criterion in the visible space, then by considering $\phi(f)$ and $\partial\phi(f)$ define 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 a Gaussian potential is simply $h_0 = Wx_0$

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

$$h_{t+1} = h_t + \gamma\left(\hat{\partial\phi}(f)(h_t) + \epsilon\mathcal{N}(0, I)\right) \tag{3}$$

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 being the Brownian noise and depending on ϵ and the second one being $(\hat{\partial\phi}(f)(h_t) - \partial\phi(f)(h_t))$ depending on the variance of the estimator (and thus can be lowered by raising the number of samples if needed).

This proposition, however, has been shown to be problematic because the norm of h tend to explode to values that are not encountered for representation of samples in the training set, it is to say that we explore a space that is not use for modeling of our protein family. To fix it, we add a regularization term. We now want to maximize $\phi'(f)(h) = \phi(f)(h) - \frac{1}{2}\lambda\|h\|_2^2$.

This modifies our recurrence slightly, and we now update 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 selected wisely.

We then rewrite this equation :

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

4.3.3 Controlling constraints through Constrained Langevin Dynamic (CLD)

Let g a function in the visible space that we want to keep null (a constraint). We consider $\phi(g)$ and $\partial\phi(g)$ as defined in section 4.3.1.

We propose to update h_{t+1} with :

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

Can we choose b_t to ensure $\partial\phi(g)(h_{t+1}) = 0$?

With Taylor formula, we can approximate $\phi(h_{t+1})$:

$$\begin{aligned}\phi(g)(h_{t+1}) &\approx \hat{\phi}(g)(h_t) + \partial\phi(g)(h_t)\cdot(h_{t+1} - h_t)\\ &\approx \hat{\phi}(g)(h_t) + \gamma\partial\phi(g)(h_t)\cdot[\mathcal{C}_{\epsilon,\lambda}(f)(h_t) + b_t\hat{\partial\phi}(g)(h_t)]\end{aligned}$$

For a better understanding of what is happening here, we rewrite this equation by using the projector on $\hat{\partial\phi}(g)(h_t)$:

$$P_t(\vec{x}) = \frac{\left\langle \hat{\partial\phi}(g)(h_t)|x \right\rangle}{\left\langle \hat{\partial\phi}(g)(h_t)|\hat{\partial\phi}(g)(h_t) \right\rangle} \hat{\partial\phi}(g)(h_t)$$

We then have with this new notation:

$$\begin{aligned}
\phi(h_{t+1}) &\approx \phi(h_t) + \gamma \partial\phi(g)(h_t) \cdot (h_{t+1} - h_t) \\
&\approx \phi(h_t) + \gamma \partial\phi(g)(h_t) \cdot [b'_t P_t(\mathcal{C}_{\epsilon,\lambda}(f)(h_t)) + (\mathcal{C}_{\epsilon,\lambda}(f)(h_t) - P_t(\mathcal{C}_{\epsilon,\lambda}(f)(h_t))] \\
&\approx \phi(h_t) + \gamma b'_t \langle \partial\phi(g)(h_t)|\mathcal{C}_{\epsilon,\lambda}(f)(h_t)\rangle
\end{aligned}$$

and by identifying b_t with b'_t, $b_t = \frac{(b'_t - 1)}{\|\partial\phi(g)(h_t)\|_2^2} \langle \partial\phi(g)(h_t)|\mathcal{C}_{\epsilon,\lambda}(f)(h_t)\rangle$

Then, if we resolve the equation $\phi(g)(h_{t+1}) = 0$, we have an estimation of b'_t

$$b'_t = \frac{\phi(g)(h_t)}{\gamma \langle \partial\phi(g)(h_t)|\mathcal{C}_{\epsilon,\lambda}(f)(h_t)\rangle}$$

From which we derive an estimation of b_t :

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

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 to find sequences distant from the reference sequence due to strong constraints. This then became our main criterion to maximize

$$f(x) = \sum_i \mathbf{1}(x_i = x_i^0) \tag{4}$$

- **PAM prediction** (second batch)

  For the second batch, we choose to maximize the classifier predicting the recognition of targeted NGG sequence. We recall that the classifier predict the PAM over an array $\mathcal{C}(x) = (A_1(x), T_1(x), C_1(x), G_1(x), ..., T_5(x), C_5(x), G_5(x))$ where X_k is the probability that the nucleotide X is recognized in position k according to the classifier. To maximize the likelihood of sequence to recognize motif TGG we then want to maximize:

$$f(x) = \log(T_1(x)) + \log(G_2(x)) + \log(G_3(x)) \tag{5}$$

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 evaluate 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 $e_{\max}$

$$f(x) = \left((e_{\max} - e_{\min}) + \frac{t}{100} e_{\min} - E_{\mathrm{RBM}}(x) \right)^+ + \left(E_{\mathrm{RBM}}(x) - \frac{t}{100} e_{\min} \right)^+ \tag{6}$$

- **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. We then have a control criterion of the form:

$$f(x) = (f_a(x) - c)^+ \tag{7}$$

  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 SpyCas9 x^0 to the expected distance followed by a phase when we maintain the distance. For $t < 100$ and a final similarity in an interval comprised between $c_{\min}$ and $c_{\max}$ we have:

$$f(x) = \left((c_{\max} - c_{\min}) + \frac{t}{100} c_{\min} - \sum_i \mathbf{1}(x_i = x_i^0) \right)^+ + \left(\sum_i \mathbf{1}(x_i = x_i^0) - \frac{t}{100} c_{\min} \right)^+ \tag{8}$$

4.4 Experimental Validation

4.4.1 Cloning and bacterial growth conditions

PCRs and restriction digestions were carried out with Q5 polymerase, T4 ligase and restriction enzymes from NEB. Vectors were constructed in *E. coli* DH10b cells, while PID variants were cloned into pWR5 and characterized in MG1655. All bacterial cultures and plates were grown at 37 °C in LB broth or agar supplemented with 25 µg/mL chloramphenicol, 50 µM DAPG inducer and sucrose when appropriate. To construct pWR1, parts were PCR-amplified with primers WR5/WR6 on template pFR56-ccdB, WR7/WR8 on pFR56-ccbB and WR9/WR10 on pFR148, then assembled by 3-part goldengate with BsmBI-HF. To construct pWR1.1 to 1.4, gRNA spacers were annealed from oligos (see table XX) then cloned into pWR1 by goldengate with BsaI-HFv2. pWR5 was built by goldengate of 4 PCR fragments amplified with WR10/WR29, WR5/WR6, WR7/WR30 on template pWR1.4 and WR31/WR32 on template pFR56-ccdB. pWR6 is derived from pWR5 with the promoter replaced by two BsmBI sites, and was constructed by PCR of pWR5 with WR41/WR42 and insertion of the fragment into pWR5 using SalI and PstI. pWR6.1/6.2/6.3 contain the same gRNA target site but differing PAM sites, for testing non NGG PAMs and were constructed by annealing oligos then inserting them into pWR6. All PID variants were synthesized as a eBlocks by IDT DNA and cloned by goldengate with BsaI-HFv2 into pWR5 or pWR6.1, pWR6.2 or pWR6.3. All constructs were verified by Sanger sequencing by Eurofins.

4.4.2 Sucrose resistance and fluorescence assays

Sucrose assays were conducted 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 chloramphenicol, 50 µM DAPG and sucrose as appropriate, then grown overnight, imaged and colonies counted. Fluorescence ex200-foldperiments were conducted by growing individual 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 M100 plate reader.

4.4.3 Phylogenetic tree of Cas9 PID chimeras

Cas9 sequences were obtained from [28], and whole Cas9 sequences or PID domains only were aligned using MUSCLE then maximum-likelihood phylogenetic trees were built using MEGA11 [62].

Supporting information

Data & Code Available on https://github.com/CyrilMa/DesignCas9WithCLD

S1 file. Experimentally tested sequences Zip. Fasta files, AlphaFold Prediction with PDB file, pLDDT, etc.

S2 file. Sequences with experimental and computational scores Excel File

S3 file. Experimental measurements of the activity through both mCherry and sacB PDF

S4 file. Additional constrained Langevin dynamics results with MNIST PDF

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, Ecole Polytechnique and benefits from financial support from the Centre de Recherche Interdisciplinary (CRI) through ”Ecole Doctorale Frontières de l’Innovation en Recherche et Education – Programme Bettencourt”. 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. arXiv:180308718 [q-bio]. 2018;. (Back to p.1, 4)

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, 4)

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, 15)

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, 15)

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, 6, 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, 7)

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)

27. 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)

28. 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, 3, 22)

29. 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.2)

30. 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.2)

31. 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.2)

32. 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.2)

33. 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.2)

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

35. 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, 18)

36. 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.5)

37. Kumar S, Paria B, Tsvetkov Y. Constrained Sampling from Language Models via Langevin Dynamics in Embedding Spaces; 2022. (Back to p.6)

38. 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.6)

39. 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.6)

40. 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.6)

41. 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.6)

42. 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.6, 11)

43. 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.7)

44. 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.7, 16)

45. 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)

46. 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)

47. 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)

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

49. 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)

50. 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)

51. 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.15)

52. 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.15)

53. 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.15)

54. 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.15)

55. 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.15)

56. Malbranke C, Bikard D, Cocco S, Monasson R, Tubiana J. Computational protein design with evolutionary-based and physics-inspired modeling: current and future synergies. arXiv preprint arXiv:220813616. 2022;. (Back to p.15)

57. Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy SR, et al. The Pfam Protein Families Database. Nucleic Acids Research. 2002;30(1):276–280. doi:10.1093/nar/30.1.276. (Back to p.16)

58. Pourcel C, Touchon M, Villeriot N, Vernadet JP, Couvin D, Toffano-Nioche C, et al. CRISPRCasdb a Successor of CRISPRdb Containing CRISPR Arrays and Cas Genes from Complete Genome Sequences, and Tools to Download and Query Lists of Repeats and Spacers. Nucleic Acids Research. 2020;48(D1):D535–D544. doi:10.1093/nar/gkz915. (Back to p.16)

59. 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.16)

60. 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)

61. Loshchilov I, Hutter F. Fixing Weight Decay Regularization in Adam. open review. 2018;. (Back to p.18)

62. 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.22)