Integrating experimental feedback improves generative models for biological sequences

Nucleic Acids Research, 2025, **53**, gkaf832
https://doi.org/10.1093/nar/gkaf832
Computational Biology

Integrating experimental feedback improves generative models for biological sequences

Francesco Calvanese[1,2,†], Giovanni Peinetti[1,3,†], Polina Pavlinova[2], Philippe Nghe[2], Martin Weigt[1,*]

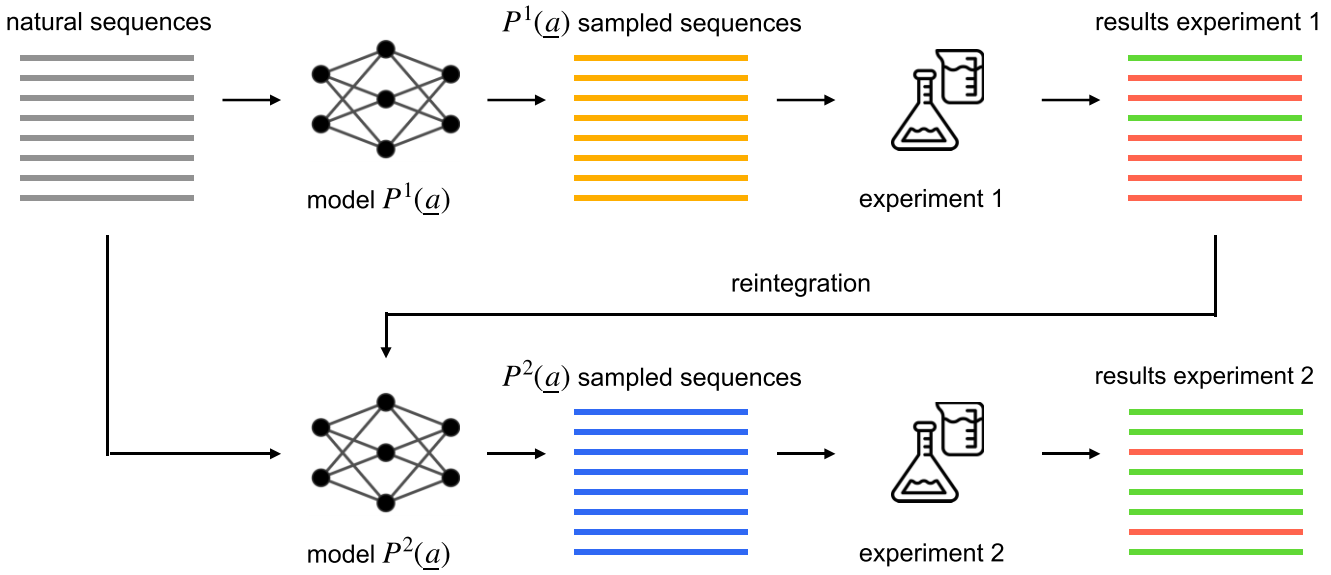
[1]Sorbonne Université, CNRS, Department of Computational, Quantitative and Synthetic Biology—CQSB, 75005 Paris, France
[2]Laboratoire de Biophysique et Evolution, UMR CNRS-ESPCI 8231 Chimie Biologie Innovation, PSL University, 75005 Paris, France
[3]Politecnico di Torino, DISAT, Torino 10129, Italy

[*]To whom correspondence should be addressed. Email: martin.weigt@sorbonne-universite.fr
[†]The first two authors should be regarded as Joint First Authors.

Abstract

Generative probabilistic models have shown promise in designing artificial RNA and protein sequences but often suffer from high rates of false positives, where sequences predicted as functional fail experimental validation. To address this critical limitation, we explore the impact of reintegrating experimental feedback into the model design process. We propose a likelihood-based reintegration scheme, which we test through extensive computational experiments on both RNA and protein datasets, as well as through wet-lab experiments on the self-splicing ribozyme from the Group I intron RNA family where our approach demonstrates particular efficacy. We show that integrating recent experimental data enhances the model's capacity of generating functional sequences (e.g. from 6.7% to 63.7% of active designs at 45 mutations). This feedback-driven approach thus provides a significant improvement in the design of biomolecular sequences by directly tackling the false-positive challenge.

Graphical abstract



Introduction

Generative probabilistic models for biological sequences, such as proteins and RNA, have recently emerged as promising tools for designing artificial biomolecules [1–9] and for addressing biologically relevant questions related to evolution, function, and sequence diversity [10–12]. These models, particularly family-specific ones like those built using direct coupling analysis (DCA) [1–3, 6, 7], as well as more advanced architectures like restricted Boltzmann machines [9], variational autoencoders [4, 5, 7], and protein language models [8], have shown notable success in generating functional sequences. However, a persistent challenge remains: these

Received: April 23, 2025. **Revised:** July 12, 2025. **Editorial Decision:** August 1, 2025. **Accepted:** August 7, 2025

models often produce a high rate of false positives—sequences generated as potentially functional by the model but failing in experimental tests.

These models are trained on sets of homologous sequences, representing families of sequences with shared evolutionary ancestry. Such families are typically characterized by highly conserved structures and functions, though the sequences themselves may diverge significantly. Multiple-sequence alignments (MSAs) [13–16], containing presumably functional sequences from different species, serve as the foundation for training. As a consequence, these models are trained in an unsupervised manner on unlabeled functional sequences, which limits their capacity to differentiate between functional and non-functional variants.

A significant issue in these generative models is the high rate of false positives—sequences deemed functional by the model that fail experimental validation [2, 7]. This limitation arises not only from the scarce sampling of the viable sequence space in the MSAs, which leads to an underrepresentation of functional diversity, but also from a more general issue inherent to all statistical generative approaches. The fitness landscape learned from natural sequence data reflects general evolutionary constraints acting across native environments and host organisms, which may differ substantially from the specific conditions used in experimental assays. As a result, even naturally occurring sequences can fail to exhibit function in laboratory settings (cf. [2, 8]). This fact limits the model's ability to accurately generate sequences, which successfully pass experimental testing.

Recent work demonstrates that labeled feedback on model generations can be leveraged to improve generative model design in both biological sequences [17] and natural language generation tasks [18]. In this study, we focus on DCA-based Potts models [2, 19, 20] and demonstrate that integrating experimental feedback including false-positive sequences into the training procedure can enhance model accuracy and reduce false-positive rates. By incorporating this feedback through an extension of the maximum-likelihood inference procedure [20], which makes explicit use of the experimental results, we show that false positives from the initial model play a critical role in refining the boundaries of the viable sequence space, thereby improving the model's performance. In principle, any experimentally labeled dataset can be used within the presented reintegration procedure, regardless of whether it was generated by the original model or not. However, sequences that are assigned high probability by the model but fail the experimental test are particularly informative, as they provide direct and informative feedback on sequence regions explored by the model.

An intriguing ingredient to this approach is that the underlying mathematical structure of the model remains unchanged, but the reintegration of experimental data significantly improves parameter learning. This highlights an important insight: the current limitations of generative models stem not necessarily from the limited expressivity of their architectures but also from the insufficient information content in the original training data. These alignments, representing natural sequences that have diverged through evolution, offer a sparse and incomplete sampling of the functional sequence landscape. Enhancing this landscape with experimental feedback allows for a reliable model, generating a higher fraction of functional sequences (i.e. true positives). Augmenting data quantity and quality at unchanged model complexity turns out to be an efficient strategy.

In the following "Materials and methods" section, we outline the main ideas of the reintegration approach and detail its application to DCA models. We also describe the datasets used to evaluate our approach. In the "Results and discussion" section, we present extensive tests on diverse RNA and protein families, progressing from purely computational settings to experimental validation. Artificial sequences sampled from our models and tested experimentally represent the most rigorous possible validation of a generative approach to biomolecules. At the end of the article, we present our main conclusion and show an outlook of possible extensions of our approach.

Materials and methods

In this section, we present the methodological framework of our proposed reintegration method, describe its application to the DCA generative model, and outline the data and procedures used to train and evaluate our models.

Reintegration method

Here, we propose a method to reintegrate experimental test results into a sequence generative model. In the standard approach, the natural data MSA $\mathcal{D}_{\mathrm{N}}$ is used to train an initial probabilistic generative model $P(\underline{a}\,|\,\theta^1) = P^1(\underline{a})$ whose parameters θ^1 are obtained through maximum likelihood estimation (MLE):

$$\theta^1 = \arg\max_{\theta}\ \mathcal{L}(\theta\,|\,\mathcal{D}_{\mathrm{N}}), \tag{1}$$

where

$$\mathcal{L}(\theta\,|\,\mathcal{D}_{\mathrm{N}}) = \frac{1}{|\mathcal{D}_{\mathrm{N}}|}\sum_{\underline{a}\in\mathcal{D}_{\mathrm{N}}} \ln P(\underline{a}\,|\,\theta). \tag{2}$$

Once the model is trained, a set $\mathcal{D}_{\mathrm{T}}$ of artificial sequences can be sampled from $P^1(\underline{a})$ and tested experimentally. However, this approach often suffers from a high rate of false-positive sequences in $\mathcal{D}_{\mathrm{T}}$, i.e. sequences expected to be functional according to $P^1(\underline{a})$, yet failing experimental tests (cf. [2, 7]). To address this issue, we propose reintegrating the experimental feedback contained in $\mathcal{D}_{\mathrm{T}}$ into an updated model, $P(\underline{a}\,|\,\theta^2) = P^2(\underline{a})$. This updated model maintains the same mathematical form and architecture as $P^1(\underline{a})$ but uses recalibrated parameters inferred leveraging the newly labeled data $\mathcal{D}_{\mathrm{T}}$. Consequently, $P^2(\underline{a})$ is expected to generate a higher proportion of functional (true-positive) sequences. To implement this, we update the model parameters optimizing a new objective function:

$$\theta^2 = \arg\max_{\theta}\ \mathcal{Q}(\theta\,|\,\mathcal{D}_{\mathrm{N}}, \mathcal{D}_{\mathrm{T}}), \tag{3}$$

where

$$\mathcal{Q}(\theta\,|\,\mathcal{D}_{\mathrm{N}}, \mathcal{D}_{\mathrm{T}}) = \mathcal{L}(\theta\,|\,\mathcal{D}_{\mathrm{N}}) + \frac{\lambda}{|\mathcal{D}_{\mathrm{T}}|}\sum_{\underline{b}\in\mathcal{D}_{\mathrm{T}}} w(\underline{b})\cdot \ln P(\underline{b}\,|\,\theta)\ , \tag{4}$$

The first contribution to $\mathcal{Q}$ equals the standard log likelihood for the natural data given in equation (2). Maximizing $\mathcal{Q}$ reduces thus to standard MLE if no experimental data is avail-

able, i.e. for $\mathcal{D}_T = \emptyset$. The second contribution is the reintegration term, which acts on the probabilities of the tested sequences in dependence on the adjustment weight w, assigned to every sequence in the tested dataset $\underline{b} \in \mathcal{D}_T$ in function of the experimental test result. We require $w(\underline{b})$ to adhere to the following rules:

- Negative adjustment: $w(\underline{b}) < 0$ for sequences failing the experimental functionality test, such that their probability $P^2(\underline{b})$ is reduced when maximizing $\mathcal{Q}$.
- Positive adjustment: $w(\underline{b}) > 0$ for sequences passing the experimental functionality test, such that their probability $P^2(\underline{b})$ is increased when maximizing $\mathcal{Q}$.

To illustrate the intuition behind this setting, consider an initial DCA model $P^1(\underline{a})$ trained for a protein enzyme family using natural sequences $\mathcal{D}_N$. Moreover, a set of artificial sequences $\mathcal{D}_T$ is sampled from P^1 and experimentally tested for enzymatic activity. Assume that approximately half of the tested sequences exhibit activity while the other half are inactive. In the reintegration step, we assign positive weights $w(\underline{b}) > 0$ to the functional sequences in $\mathcal{D}_T$, effectively reinforcing them in the training objective. This augments the natural dataset $\mathcal{D}_N$ with additional functional examples. In contrast, non-functional sequences receive negative weights $w(\underline{b}) < 0$, forcing the model to reduce their probability in the updated distribution $P^2(\underline{a})$. In this way, the reintegration procedure refines the generative model to better assign high probabilities to provably functional sequences, based on experimental evidence.

The specific values of the weights for individual sequences depend on the specific experimental setting, in the easiest case they can be taken to be all of equal absolute value (see below for some more complicated construction removing at least partially biases in the experimental data). The overall intensity of reintegration is controlled by the hyperparameter λ; the higher is λ the greater the relative importance assigned to the experimentally labeled dataset $\mathcal{D}_T$ compared to the natural sequences $\mathcal{D}_N$. When $\lambda = 0$, the classical MLE is recovered. If we consider only the functional sequences with $w(\underline{b}) > 0$ in $\mathcal{D}_T$, this procedure is similar to adding them to $\mathcal{D}_N$ with a λ-dependent weight and performing the standard MLE inference.

The essential difference arises from the inclusion of non-functional sequences with $w(\underline{b}) < 0$ in $\mathcal{D}_T$, which indicate regions of the sequence space that our model should avoid (cf. Fig. 1). Ideally, $\mathcal{D}_T$ would consist of $P^1(\underline{a})$-generated sequences. Consider a sequence $\underline{b}$ that has been generated by $P^1(\underline{a})$: the sole fact that it was sampled implies that it was assigned a high probability by the $P^1(\underline{a})$ model. If this sequence fails the experimental test, it is assigned a negative $w(\underline{b}) < 0$, and the reintegrated $P^2(\underline{a})$ model will subsequently assign it a lower probability. This procedure enables the model to correct itself based on the experimental feedback and to better infer the limits of functional sequence space.

The reintegration set $\mathcal{D}_T$ is, however, not limited to be comprised of sequences generated by $P^1(\underline{a})$ but can, in principle, be any functionally labeled dataset. It is, however, intuitively important that negative sequences are close to the functional sequence space. Random sequences, for example, are almost surely non-functional, but they typically have already very low values of $P^1(\underline{a})$, and reintegrating them negatively will not improve the description of the positively functional sequence space.

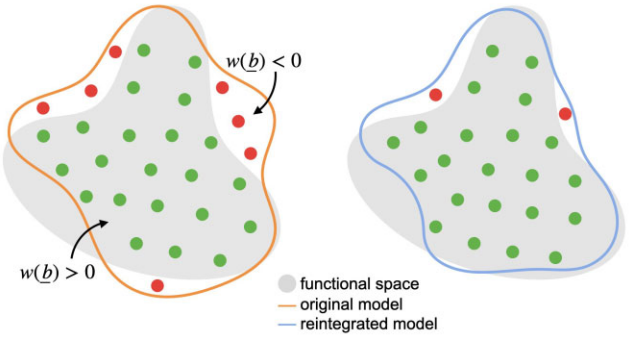


Figure 1. Stylized representation of the effect of the reintegration procedure. Sequences generated by the initial model P^1 (region inside orange line) that fail experimental tests (region outside gray area) are assigned negative adjustment weights ($w < 0$), while those that are functional are assigned positive weights ($w > 0$). The reintegrated model P^2 is then trained using these adjusted weights, resulting in generated sequences that avoid regions associated with non-functional sequences and concentrate in regions associated with functional sequences (region inside blue line), thereby reducing the fraction of false positives among the generated sequences.

Our tests of this procedure have been carried out on DCA models, since DCA has proved capable of generating functional proteins and RNAs [1, 2, 6]. Another advantage is that we can naturally implement the new objective function $\mathcal{Q}$ in the DCA framework without altering its classical training procedures. A detailed discussion is provided in the next section with all analytical derivations detailed in Supplementary Section S1. Note that, in a more generic machine-learning context, the DCA model can be replaced by other generative model architectures, and MLE by the optimization of any loss function, which is additive in data-point specific losses as, for example, cross entropies or reconstruction losses.

Reintegration method for DCA models

The presented reintegration procedure is particularly well suited for application to DCA models, as it does not require modifying their standard inference methods (differently from previous reintegration attempts [21]). In its conventional implementation, DCA assumes that the natural data distribution is described by a probability $P^1(\underline{a})$ in the form of a Potts model,

$$P^1(\underline{a}) = \frac{1}{Z^1} \exp\left\{ \sum_i h_i^1(a_i) + \sum_{i<j} J_{ij}^1(a_i, a_j) \right\}, \qquad (5)$$

with $\underline{a} = (a_1, \ldots, a_L)$ denoting, according to the problem under study, an aligned nucleotide or amino acid sequence of length L. The optimal parameters $\{h^1, J^1\}$ for $P^1(\underline{a})$ are inferred via MLE [20],

$$\{h^1, J^1\} = \arg\max_{h,J} \mathcal{L}(h, J \mid \mathcal{D}_N), \qquad (6)$$

where $\mathcal{D}_N$ is an MSA of all sequenced homologs of the family under consideration.

We omit here the standard sequence reweighting procedure [22] commonly used in DCA, to simplify notation, but it can be included straightforwardly. Our publicly available implementation does contain it. The reintegration data consist of a second alignment, $\mathcal{D}_T$, of experimentally tested sequences,

together with experimental outcomes encoded by the adjustment weight $w(\underline{b})$. The reintegrated model $P^2(\underline{a})$ remains in the Potts model form,

$$P^2(\underline{a}) = \frac{1}{Z^2} \exp\left\{ \sum_i h_i^2(a_i) + \sum_{i<j} J_{ij}^2(a_i, a_j) \right\}, \quad (7)$$

but with updated parameters $\{h^2, J^2\}$ chosen to maximize the new objective function

$$\{h^2, J^2\} = \arg\max_{h,J} \mathcal{Q}(h, J \mid \mathcal{D}_N, \mathcal{D}_T). \quad (8)$$

In standard DCA inference, the MLE conditions require the one-point marginals $P_i^1(a)$ and two-point marginals $P_{ij}^1(a, b)$ of $P^1(\underline{a})$ to match the empirical frequencies $f_i(a)$ and $f_{ij}(a, b)$ observed in $\mathcal{D}_N$:

$$P_i^1(a) = \frac{1}{|\mathcal{D}_N|} \sum_{\underline{a} \in \mathcal{D}_N} \delta_{a_i,a} = f_i(a),$$

$$P_{ij}^1(a, b) = \frac{1}{|\mathcal{D}_N|} \sum_{\underline{a},\underline{b} \in \mathcal{D}_N} \delta_{a_i,a} \cdot \delta_{a_j,b} = f_{ij}(a, b), \quad (9)$$

where $\delta_{a,b}$ is the Kronecker delta.

In contrast, maximizing $\mathcal{Q}$ imposes choosing $\{h^2, J^2\}$ so that the one-point and two-point marginals of $P^2(\underline{a})$ match the $\mathcal{D}_T$-corrected frequencies $\tilde{f}$ (see Supplementary Section S1):

$$\tilde{f}_i(a) = \frac{1}{z}\left[f_i(a) + \frac{\lambda}{|\mathcal{D}_T|} \sum_{\underline{b} \in \mathcal{D}_T} w(\underline{b}) \cdot \delta_{b_i,a} \right],$$

$$\tilde{f}_{ij}(a, b) = \frac{1}{z}\left[f_{ij}(a, b) + \frac{\lambda}{|\mathcal{D}_T|} \sum_{\underline{b} \in \mathcal{D}_T} w(\underline{b}) \cdot \delta_{b_i,a} \cdot \delta_{b_j,b} \right], \quad (10)$$

where

$$z = 1 + \frac{\lambda}{|\mathcal{D}_T|} \sum_{\underline{b} \in \mathcal{D}_T} w(\underline{b}). \quad (11)$$

Hence, the maximization condition becomes

$$P_i^2(a) = \tilde{f}_i(a),$$

$$P_{ij}^2(a, b) = \tilde{f}_{ij}(a, b) \quad (12)$$

for all i, j and all a, b. Thus, any standard DCA inference method may be used simply by replacing the original empirical frequencies f with the corrected frequencies $\tilde{f}$ from equation (10). Consequently, implementing reintegration in a DCA pipeline requires only altering the frequency targets used during parameter inference. In this work, we applied the reintegration procedure to two implementations of DCA: Boltzmann machine DCA [2, 23] and edge activation DCA (eaDCA) [24]. Further details about these models can be found in Supplementary Section S2.

It is important to note, however, that negative values of $w(\underline{b})$ (i.e. experimentally tested non-functional sequences) can break the convexity of the problem, removing guaranteed uniqueness or even existence of a solution. Additionally, the corrected frequencies $\tilde{f}$ may sometimes lie outside the interval [0, 1]. As a result, convergence to a consistent DCA model is not strictly guaranteed, especially for strong reintegration strengths λ. To facilitate convergence, we adopted several strategies, as discussed in Supplementary Section S2 and showcased in the "Results and discussion" section. Nonetheless, in all of our applications, we found that moderate settings of λ do converge reliably toward an improved model P^2.

Data, RNA fitness proxy, and Group I intron experimental activity

Here, we present the datasets used to evaluate our reintegration procedure. Computational tests are performed using RNA MSAs from RFAM [13], as well as a protein alignment of the chorismate mutase (CM) enzyme [2]. Experimental validation is carried out with Group I intron ribozymes, for which high-throughput assays are available to quantify the activity of a large number of artificial designs [7].

RFAM sequence data

The RNA datasets used in this paper are MSA of three RNA families. These families are the transfer RNA (tRNA) family RF00005 (number of sequences $|\mathcal{D}_N| = 30\,000$, number of residues $L = 71$), the SAM (S-adenosylmethionine) riboswitch family RF0162 ($|\mathcal{D}_N| = 6113$, $L = 108$), and the glycine riboswitch family RF0504 ($|\mathcal{D}_N| = 4600$, $L = 94$). For all three RNA families, the corresponding consensus secondary structure is available from the Rfam database [13] and is shown in Fig. 2, visualized with the Forna software [25]. The datasets are taken from [24].

RNA fitness proxy

To study the reintegration procedure, we need a fast and reliable way to label our datasets. Performing experiments on RNA sequence functionality is both expensive and time-consuming, so we initially used computational tools as fitness proxies to label our datasets. Since we have access to the RNA families' consensus secondary structures, we employed the `RNAeval` function provided by the ViennaRNA Package [26]. `RNAeval` calculates the thermodynamic free energy F (using the Turner 2004 energy model [27]) of an RNA sequence folded onto a given secondary structure.

Our underlying assumption is that well-generated sequences will, on average, exhibit low free energy F when folded onto the family's consensus structure, indicating structural stability. In contrast, poorly generated sequences may lack structural stability or fold into alternative structures and are expected to have a higher free energy. Therefore, we will use $-F$ as a fitness proxy. This approach leverages the established structure/function relationship in RNA [28, 29] and enables us to computationally label sequences for a given target secondary structure.

Group I intron ribozymes data

The Group I intron datasets utilized in this paper are taken from the study by Lambert *et al.* [7]. These datasets consist of MSAs of Group I intron ribozymes. The natural data $\mathcal{D}_N$ consist of an MSA of $N = 817$ natural Group I introns aligned against the reference sequence, the *Azoarcus* Group I intron (number of residues $L = 197$). The reintegration dataset $\mathcal{D}_T$ consists of a subset of 14 099 out of the original 24 071 experimentally labeled sequences. Specifically, $\mathcal{D}_T$ includes the sequences with mutational distance from 3 to 60, excluding those near the activity threshold. Sequences with mutational distances exceeding 60 were not reintegrated due

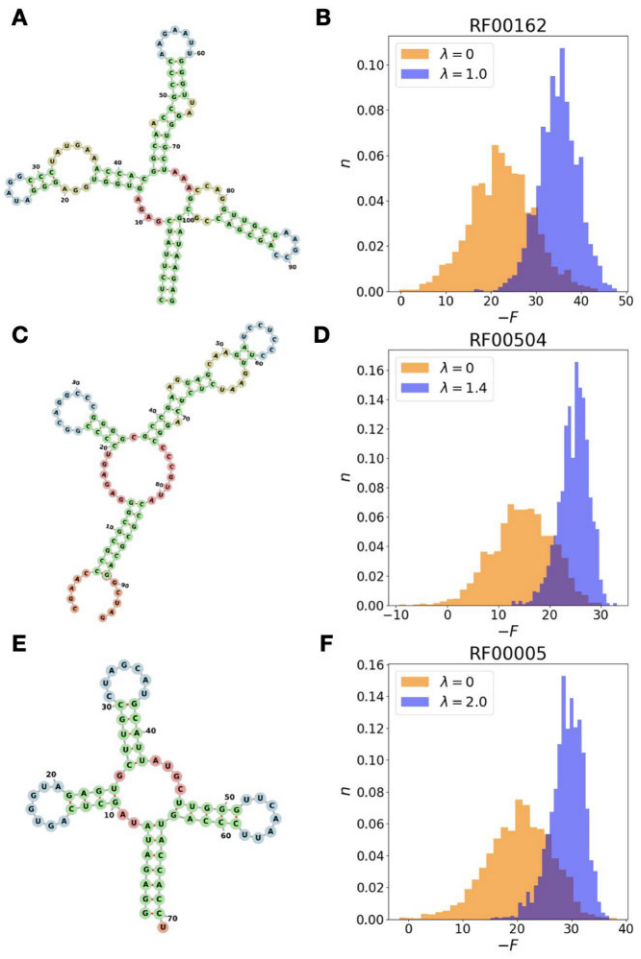


Figure 2. (**A**) Consensus secondary structure of the RF00162 RNA family (SAM riboswitch). (**B**) Distribution of the `RNAeval` proxy fitness for the RF00162 DCA model $P^1(\underline{a})$ (orange; $\lambda = 0$) and the reintegrated $P^2(\underline{a})$ model (blue; $\lambda = 1$). (**C**) Consensus secondary structure of the RF00504 RNA family (glycine riboswitch). (**D**) Distribution of the `RNAeval` proxy fitness for the RF00504 DCA model $P^1(\underline{a})$ (orange; $\lambda = 0$, $N = 2000$) and the reintegrated $P^2(\underline{a})$ model (blue; $\lambda = 2$, $N = 2000$). (**E**) Consensus secondary structure of the RF00005 RNA family (tRNA). (**F**) Distribution of the `RNAeval` proxy fitness for the RF00005 DCA model $P^1(\underline{a})$ (orange; $\lambda = 0$, $N = 2000$) and the reintegrated $P^2(\underline{a})$ model (blue; $\lambda = 1$, $N = 2000$). The secondary structure diagrams were generated using the Forna software [25].

to the unreliable detection of activity signals beyond this range (see Supplementary Section S3 and Supplementary Fig. S1).

Group I experimental activity

Group I introns are catalytic RNA sequences capable of self-splicing, a process in which the intron autonomously excises itself from precursor RNA and subsequently ligates the adjacent exons. To experimentally validate the reintegration procedure, we conducted experiments on artificially designed Group I intron ribozymes. This self-splicing activity was experimentally determined using the same methodology as the tested ribozymes in Lambert *et al.* [7]. The experimental assay consists of a high-throughput screening of self-splicing catalytic activity. Additional details about the procedure and its replicability are provided in Supplementary Section S3, Supplementary Figs S2 and S3, and Supplementary Table S1.

Protein data

The protein datasets used in this paper are taken from the study by Russ *et al.* [2]. These datasets consist of MSAs of the CM enzyme. $\mathcal{D}_N$ consists of the MSA of the natural CM homologs (number of sequences $N = 1130$, number of residues $L = 96$), and the reintegration dataset $\mathcal{D}_T$ is the alignment ($N = 1003$, $L = 96$) of DCA-generated artificial CM variants. In [2], both datasets are labeled based on experimental testing: all protein sequences were expressed in genetically engineered *Escherichia coli* strains, each modified to produce one of the CMs from $\mathcal{D}_N$ or $\mathcal{D}_T$ instead of their natural wild-type variants. These *E. coli* strains were then tested for growth under selective conditions; sequences enabling *E. coli* growth were labeled as functional, whereas those that did not were labeled as non-functional. It is noteworthy that many natural sequences in $\mathcal{D}_N$ were non-functional in *E. coli* under the experiment's conditions: the natural CM homologs have undergone evolutionary adaptations specific to their native hosts, and may fail to provide growth in the *E. coli* environment.

Results and discussion

In this section, we present computational tests of the reintegration procedure on RNA and protein data, along with experimental validation using Group I intron ribozymes.

The effect of the reintegration strength λ in Rfam RNA families

To understand the action of the proposed reintegration method, we need to study its performance systematically in function of the reintegration strength λ. For this aim, we use three RNA-family MSAs from the Rfam database (cf. the "Materials and methods" section). Before performing resource and time-consuming experiments (cf. below), we first assess the role of λ via a fully computational approach. As a proxy for the experimental fitness, we employ the negative free energy $-F$ of folding a given sequence onto the family's consensus secondary structure computed from the Turner model [27] implemented in the `RNAeval` function of the ViennaRNA package [26]. Further details on the data and fitness proxy are provided in the "Materials and methods" section.

For the statistical models, we use our recent time-efficient RNA-tailored eaDCA [24]. It provides easy access to the inferred models' Shannon entropy S [30] in function of λ, and thereby allows to quantify the potential diversity of the sequences generated by the model.

To this end, for each of the three RNA families, we used the Rfam MSA as $\mathcal{D}_N$ and trained our initial model $P^1(\underline{a})$ via eaDCA. From this model, we sampled the $P^1(\underline{a})$ dataset comprising 2000 sequences, to be used as $\mathcal{D}_T$ in the reintegration procedure. We measured the `RNAeval` proxy fitness $-F$ [26] of these sequences (cf. the "Materials and methods" section) and decided (somewhat arbitrarily) to consider all sequences with above-average $-F$ as functional and below-average $-F$ as non-functional—the training objective for $P^2(\underline{a})$ thus being the generation of highly thermo-stable sequences. We thus define a simple adjustment weight $w(\underline{b})$ for all $\underline{b} \in \mathcal{D}_T$:

$$w(\underline{b}) = \begin{cases} +1, & \text{if } -F(\underline{b}) \geq -F_{\text{avg}}, \\ -1, & \text{if } -F(\underline{b}) < -F_{\text{avg}}. \end{cases} \tag{13}$$

where F_{avg} is the average `RNAeval` folding free energy evaluated for the $\mathcal{D}_{\text{T}}$ dataset.

A reasonable range for the reintegration strength λ can be estimated by considering the product $\lambda \cdot |w|$. As previously noted, the case $\lambda = 0$ corresponds to standard DCA inference on $\mathcal{D}_{\text{N}}$ alone. When $\lambda \cdot |w| \approx 1$, the contributions of $\mathcal{D}_{\text{T}}$ and $\mathcal{D}_{\text{N}}$ to the objective function $\mathcal{Q}$ become comparable. Therefore, it is reasonable to explore values of λ such that this product lies in the range of 0 to slightly above 1. For larger values, convergence issues may arise during model training (see the "Materials and methods" section).

To assess the effect of the reintegration procedure, we monitor the following quantities:

(1) Average Proxy Fitness: $-\overline{F}$ is the average proxy fitness of the sequences in the P^2 dataset. This measures how well our reintegration is pushing the generation toward "functional" sequences according to equation (13).
(2) True Positives: TP is defined as the fraction of sequences in the P^2 dataset that exhibit a fitness score $F(\underline{b}) \geq -F_{\text{avg}}$. In other words, these sequences would have been assigned a positive weight during the reintegration procedure, indicating predicted functionality. This metric quantifies the effectiveness of the approach in reducing false positives.
(3) Model entropy: S quantifies the diversity of sequences generated by the reintegrated model P^2. A higher entropy suggests a more diverse sequence space.
(4) Average intra-dataset distance: $D_{P^2-P^2}$ is the average Hamming distance (number of mutations) between pairs of sequences in the P^2 dataset. It quantifies the diversity between generated sequences. This value is to be compared with $D_{P^1-P^1}$, i.e. the average distance in the non-reintegrated $P^1(\underline{a})$ dataset.
(5) Average minimum distance to functional sequences in $\mathcal{D}_{\text{T}}$: For each sequence in the P^2 dataset, we calculate the Hamming distance to the closest functional sequence in the reintegration dataset $\mathcal{D}_{\text{T}}$. The average of these is reported as $D_{P^2-\mathcal{D}_{\text{T}}^+}$. This metric ensures that our model is not merely replicating functional sequences from $\mathcal{D}_{\text{T}}$.
(6) Average minimum distance to non-functional sequences in $\mathcal{D}_{\text{T}}$: For each sequence in the P^2 dataset, we calculate the Hamming distance to the closest non-functional sequence in the reintegration dataset $\mathcal{D}_{\text{T}}$. The average of these is reported as $D_{P^2-\mathcal{D}_{\text{T}}^-}$. This metric allows us to test whether generated sequence avoid the vicinity of non-functional sequences from $\mathcal{D}_{\text{T}}$, as is to be expected by their negative contribution to the objective $\mathcal{Q}$.

By tracking these quantities, we can evaluate whether the reintegration leads to a better model without overfitting the reintegrated data. The results of these analyses are presented in Table 1 for the RF00162 RNA family. Similar results are observed also for the other families (cf. Supplementary Section S4, Supplementary Tables S2–S4, and Supplementary Fig. S4).

In general, we observe that higher values of λ yield a higher average proxy fitness $-\overline{F}$, demonstrating that the reintegration procedure effectively enhances the fitness of the generated sequences. For example, for the RF00162 RNA family (Table 1), the average proxy fitness increases from 22.4 ($\lambda = 0$) to 34.9 ($\lambda = 1$), and the TP fraction improves from 49.9% to 99.4%. However, this gain comes at the expense of model entropy S and overall sequence diversity: S decreases from 63.4 ($\lambda = 0$) to 53.3 ($\lambda = 1$), while the average intra-dataset distance $D_{P^2-P^2}$ drops from 46.0 ($\lambda = 0$) to 39.7 ($\lambda = 1$). Additionally, generated sequences become somewhat closer to the functional sequences in the reintegration dataset $\mathcal{D}_{\text{T}}^+$, as indicated by a reduction in the average minimum distance $D_{P^2-\mathcal{D}_{\text{T}}^+}$ from 28.1 ($\lambda = 0.1$) to 25.3 ($\lambda = 1$). The average minimum distance to non-functional sequences $D_{P^2-\mathcal{D}_{\text{T}}^-}$ slightly increases from 29.6 ($\lambda = 0.1$) to 31.8 ($\lambda = 1$).

Table 1. Effect of the reintegration strength for the RF00162 RNA family

λ	$-\overline{F}$	TP	S	$D_{P^2-P^2}$	$D_{P^2-\mathcal{D}_{\text{T}}^+}$	$D_{P^2-\mathcal{D}_{\text{T}}^-}$
0	22.4	49.9%	63.4	46.0	–	–
0.1	24.3	60.4%	65.3	45.8	28.1	29.6
0.5	30.2	91.3%	63.2	42.2	26.7	30.5
1.0	34.9	99.4%	53.3	39.7	25.3	31.8

Note that the values reported for $\lambda = 0$ correspond to the DCA model without reintegration, i.e. $-F$avg $= 22.4$ is the threshold value chosen for functional sequences and the entropy $S = 63.4$ equals the one of P^1.

We were able to perform these tests because we can readily compute the proxy fitness also on the sequences generated from the reintegrated model. In more realistic scenarios, this is not possible without experiments. However, our observations guide the choice of λ before doing experiments: it seems reasonable to select a value before a significant loss in diversity occurs or before the training procedure fails to converge.

The effect of reintegration on the proxy fitness $-F$ distributions for the three Rfam families is shown in Fig. 2. In all cases, the P^2 dataset samples exhibit a significant shift to higher proxy fitness compared to the P^1 dataset (indicated by $\lambda = 0$). We find, in particular, that "non-functional" sequences with proxy fitnesses below $-F_{\text{avg}}$ become very rare.

A deeper analysis of the effect of the size of the reintegration dataset $\mathcal{D}_{\text{T}}$, as well as of the reintegration of deep mutational scanning-like data, was performed in the RNA context and is presented in Supplementary Section S4, Supplementary Fig. S5, and Supplementary Table S5.

In addition to generating sequences with improved proxy fitness, another effect of the reintegration is that the DCA model score becomes a more reliable predictor of fitness; an in-depth analysis of this phenomenon is provided in Supplementary Section S4 and Supplementary Fig. S6

Reintegrating experimental activity of a protein family

Our case study for applying the reintegration procedure to proteins is the CM enzyme, which plays an essential role in the biosynthesis of aromatic amino acids. This enzyme serves as an ideal setting to test our procedure because Russ *et al.* [2] have already trained a DCA model P^1 on an MSA $\mathcal{D}_{\text{N}}$ of natural CM homologs, and they have experimentally tested the natural sequences of $\mathcal{D}_{\text{N}}$ as well as a dataset $\mathcal{D}_{\text{T}}$ of P^1-designed CM variants using an *in vivo* growth assay (see the "Materials and methods" section). As a result, we have access to experimentally labeled datasets indicating sequence functionality. Moreover, Russ *et al.* demonstrated that it is possible to train a simple logistic regression (LR) classifier on $\mathcal{D}_{\text{N}}$ to predict whether an artificial CM variant is estimated to be functional or not. We can leverage these findings for our reintegration procedure.

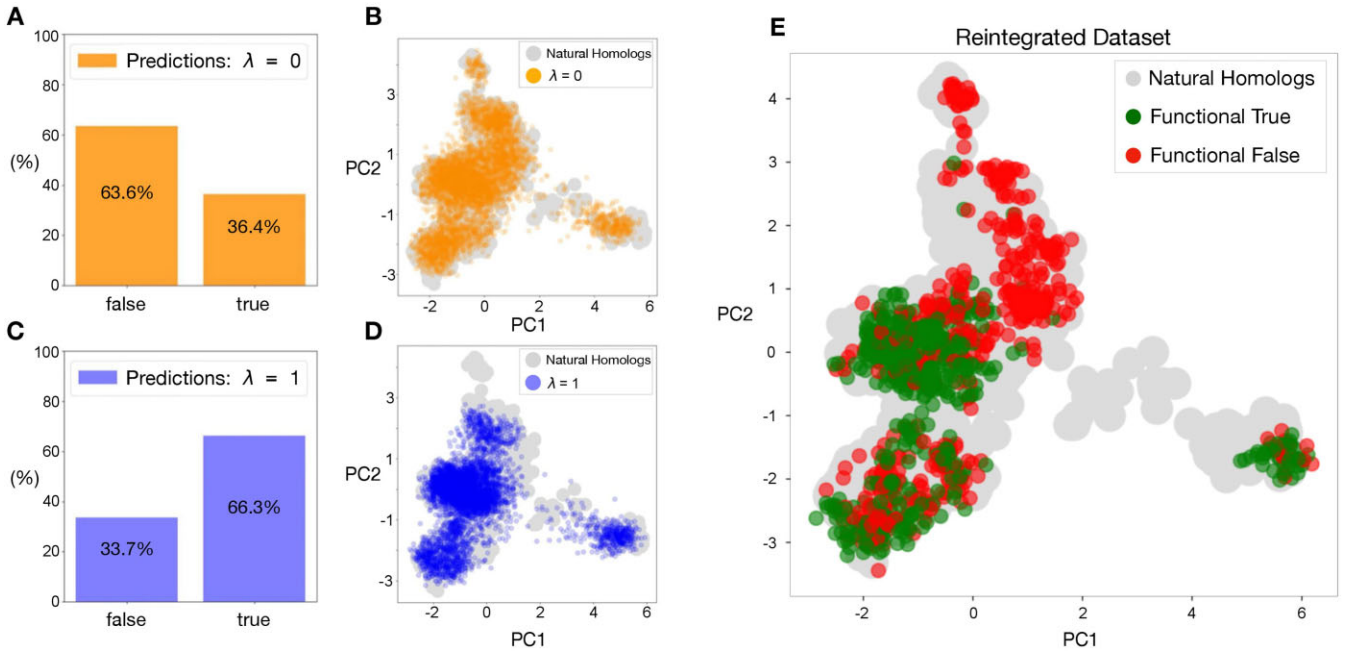


Figure 3. (**A**) Classifier predictions for protein-estimated functionality in the $P^1(\underline{a})$ dataset ($N = 8000$): "True" indicates estimated functional, "False" indicates estimated non-functional. (**B**) Principal-component analysis (PCA) projection of the $P^1(\underline{a})$ dataset (orange); natural MSA homologs are represented by the gray cloud. (**C**) Classifier predictions for protein functionality in the $P^2(\underline{a})$ dataset ($N = 8000$): "True" indicates predicted functional, "False" indicates predicted non-functional. (**D**) PCA projection of the $P^2(\underline{a})$ dataset (blue); natural MSA homologs are represented by the gray cloud. (**E**) PCA projection of the experimentally labeled dataset $\mathcal{D}_T$ ($N = 1003$) of artificially generated CM sequences. Functional sequences are shown in green, non-functional ones in red. The natural sequences in $\mathcal{D}_N$ ($N = 1130$) are shown in gray (note that not all of them are functional in the experimental test performed in *E. coli*).

Our approach begins by training an LR classifier using the labeled natural CM variants in $\mathcal{D}_N$ to estimate experimental functionality. This classifier achieves an accuracy of ∼80% in estimating the functionality of artificially generated sequences (cf. Supplementary Section S5), which is consistent with the results reported [2]. Since experimental testing of our artificial sequences is not feasible within this study, we will use this classifier to evaluate the performance of our reintegration procedure. Note that the labels used for training the classifier are not used in our reintegration procedure but are complementary information exploitable for posterior sequence evaluation.

We first trained our initial DCA model $P^1(\underline{a})$ using the natural MSA $\mathcal{D}_N$ as training data and the adabmDCA (Adaptive Boltzmann Machine Direct Coupling Analysis) implementation of DCA [31]. From this model, we sampled the P^1 dataset comprising 8000 artificial CM variants.

To train our reintegrated DCA model $P^2(\underline{a})$, we used dataset $\mathcal{D}_T$, which is experimentally labeled in [2], allowing us to avoid relying on proxy fitness measures. We chose again a binary adjustment function $w(\underline{b})$ for all sequences in $\mathcal{D}_T$

$$w(\underline{b}) = \begin{cases} +1, & \text{if } \underline{b} \text{ is functional}, \\ -1, & \text{if } \underline{b} \text{ is non-functional}, \end{cases} \tag{14}$$

We set the reintegration strength parameter λ to 1, which is the highest value that converged within an acceptable time frame. The reintegration procedure is significantly slower for proteins compared to RNA, requiring 3 h of runtime on an L4 GPU using the most advanced DCA GPU implementation available. Also, here we sampled from the resulting model a P^2 dataset containing 8000 artificial CM sequences. Results for lower values of λ are provided in Supplementary Table S6.

To assess the effectiveness of our reintegration procedure, we employed the LR classifier and determined the percentage of variants estimated to be functional in both the P^1 and the reintegrated P^2 dataset. The results of the analysis indicate that the reintegration procedure was successful, leading to a substantial increase in the LR-estimated functional fraction from 36.6% to 66.3%, as shown in Fig. 3A and C.

To investigate how reintegration affects the distribution of the generated designs in sequence space, we performed a PCA analysis. Each $\mathcal{D}_N$ amino acid sequence was first one-hot encoded, representing each residue as a 21-dimensional binary vector (20 standard amino acids in alphabetical order, plus a 21st component for gaps), with a 1 in the position corresponding to the observed amino acid and 0 elsewhere. These high-dimensional vectors were used to compute the principal components based on the natural MSA $\mathcal{D}_N$. All other datasets (P^1, P^2, and $\mathcal{D}_T$) were one-hot encoded in the same way and projected onto the first two principal components derived from $\mathcal{D}_N$ to visualize their relative positions in sequence space.

In the PCA plots shown in Fig. 3B and D, we project the P^1- and P^2-generated datasets (colored) onto the first two principal components of the natural MSA $\mathcal{D}_N$ (gray). Notably, the P^2 dataset avoids certain regions in PCA space that are occupied by sequences from both the natural dataset and the P^1 model. To better understand this behavior, Fig. 3E projects the reintegration dataset $\mathcal{D}_T$ onto the same PCA, with functional sequences shown in green and non-functional ones in red. We observe that the regions avoided by the P^2 dataset correspond to the non-functional areas in $\mathcal{D}_T$. These findings suggest that the observed reduction in diversity of sequences generated after reintegration results from the exclusion of non-functional regions, effectively refining the functional sequence space in the P^2 model compared to the standard DCA model P^1.

Experimental validation on Group I intron ribozymes

Group I intron ribozymes are catalytic RNA molecules that possess the ability to self-splice, meaning they can excise themselves from precursor RNA transcripts without the assistance of proteins or additional enzymes. Our work leverages on that of Lambert *et al.* [7], who started with a reference wild type, the *Azoarcus* Group I intron ribozyme and designed artificial mutations of this reference sequence using MSA-based generative models (DCA [24] and VAE [4]), structure-based methods (Turner model [27]), and combinations of the two. Through high-throughput assays, they tested these mutations for self-splicing-like activity and evaluated the percentage of active designs from each model at varying distances from the wild type. Here, we present experimental evidence supporting the presented reintegration method applied in this context.

We start with the DCA model of [7], utilizing it as our non-reintegrated P^1 model. This model was trained using as training set $\mathcal{D}_N$ an MSA of 817 Group I introns aligned against the *Azoarcus* Group I intron ribozyme ($L = 197$). For the training of our reintegrated models, we employ as $\mathcal{D}_T$ a subset of 14 099 experimentally labeled RNA sequences (see the "Materials and methods" section and details in Supplementary Section S3). These sequences were previously generated and tested in [7] and are used solely for the training of our models.

An important detail is that $\mathcal{D}_T$ in this case is significantly different from all other reintegration instances in this study. All tested designs are mutations of a single reference sequence, with up to 60 mutations from wild type, so their distribution in the sequence space is localized around one single point. Furthermore, as expected, designs with fewer mutations generally exhibited higher activity than those with a larger number of mutations. Thus, a significant amount of information about sequence functionality in $\mathcal{D}_T$ is trivially contained in the mutational distance from wild type. Unlike all the other instances presented, where $\mathcal{D}_T$ designs are exclusively derived from the non-reintegrated model P^1, in this case, they originate from all the different models tested in [7]. Details about $\mathcal{D}_T$ are provided in Supplementary Section S3 and Supplementary Fig. S1.

Building on this setting, our goal was to generate and experimentally test new designs to validate the reintegration procedure. We implemented two reintegration strategies:

(1) Standard reintegration procedure (REINT): All active $\mathcal{D}_T$ sequences ($\mathcal{D}_T^+$) were reintegrated with a weight $w(\underline{b}) = 1/|\mathcal{D}_T^+|$, and all non-active $\mathcal{D}_T$ sequences ($\mathcal{D}_T^-$) were reintegrated with $w(\underline{b}) = -1/|\mathcal{D}_T^-|$.
(2) Bin sum zero reintegration (REINT SB0): $\mathcal{D}_T$ sequences were grouped into bins based on their mutational distance from the reference sequence, with each bin covering four mutational steps. For sequences in bin i, the weight $w(\underline{b})$ was set to $1/|b_i^+|$ for active sequences, where $|b_i^+|$ is the number of active sequences in the bin, and $w(\underline{b}) = -1/|b_i^-|$ for non-active sequences, where $|b_i^-|$ is the number of non-active sequences in the bin. This ensured that the sum of $w(\underline{b})$ in each bin was zero, mitigating bias toward the *Azoarcus* reference sequence by balancing positive and negative signals at each distance.

For further details and the choice of the parameter λ, refer to Supplementary Section S3.

Table 2. Number of tested designs of the two reintegrated models at various mutational Hamming distances (header row)

Model	30	45	55	60	65	70	75
REINT	–	80	180	180	180	180	–
REINT SB0	100	150	–	275	–	–	275

For the DCA P1 model, this number is 150 for each bin [7].

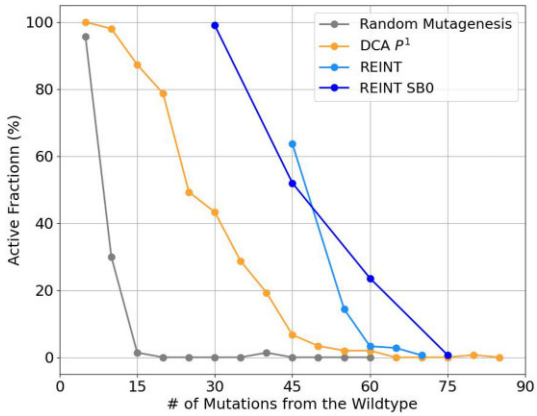


Figure 4. Active fraction of the generated designs as a function of the mutational distance from the reference sequence for the P^1 DCA model and the two reintegrated models. The number of tested designs at each bin of distance can be found in Table 2.

The two models were trained using the same GPU DCA implementation used for the CM case.

From the two reintegrated models, we generated designs at various bins of mutational distance from the reference wild type (Table 2). These designs were then experimentally tested for self-splicing activity using the same experimental assay that was used for the designs from the non-reintegrated P^1 model [7]. Details about the experimental procedures are provided in the "Materials and methods" section and in Supplementary Section S3.

The results of the experimental assays are displayed in Fig. 4 and detailed in Supplementary Section S6 and Supplementary Table S7.

Both reintegrated models outperformed the non-reintegrated one. The REINT SB0 model maintained an active fraction of 23.6% at a mutational distance of 60 residues, compared to the non-reintegrated P^1 model, which had a similar active fraction of 19.3% at 40 mutations but dropped to just 2.0% at 60 mutations. The REINT model successfully generated functional sequences at 65 mutations, further away than all the $P^1(\underline{a})$ tested in [7], where the furthest active sequence was found at 60 mutations.

This increase in performance came at the expense of generated sequence diversity, which was significantly more impacted than in the previous reintegration examples. This is most likely due to the highly localized nature of the reintegrated dataset $\mathcal{D}_T$. The intra-sample and $\mathcal{D}_T^+$ distance distributions for mutational distances of 45, 60, and 65 are shown in Table 3, with the complete analysis provided in Supplementary Table S7 and Supplementary Figs S7 and S8.

Table 3. Active fraction, average intra-dataset distance ($D_{P^2\_P^2}$), and average minimum distance from the positively reintegrated dataset ($D_{P^2\_\mathcal{D}_T^+}$) for mutational distances 60, 70, and 75 from the reference sequence

Model (distance)	Active %	$D_{P^2\_P^2}$	$D_{P^2\_\mathcal{D}_T^+}$
DCA P^1 (45)	6.7	51.3	31.8
REINT (45)	63.7	8.7	6.5
REINT SB0 (45)	52.0	10.0	7.6
DCA P^1 (60)	2.0	62.8	43.1
REINT (60)	3.3	12.5	18.6
REINT SB0 (60)	23.6	15.1	13.8
DCA P^1 (65)	0.0	70.4	50.3
REINT (65)	2.8	18.1	28.0

Conclusion and outlook

Advances in high-throughput experimentation and machine learning are rapidly reshaping our ability to design functional biomolecular sequences. In this work, we have demonstrated that reintegrating experimental feedback into generative models markedly improves the reliability of predicted sequences. Our results show that even when the underlying mathematical model remains unchanged, the incorporation of well-characterized experimental data—including both functional and non-functional sequences—serves as a powerful corrective mechanism. Experimental tests therefore do not only validate predictions of models entirely trained on natural sequence data but also guide the refinement of the modeled sequence space by penalizing non-functional regions and reinforcing areas associated with activity.

A key insight from our study is that enhanced performance can be achieved by training on more informative data rather than solely by increasing model complexity. The same DCA framework, when trained with a balanced integration of natural sequence data and experimental outcomes, produces a model that yields a significantly higher fraction of functional sequences. This observation underscores that the limitations of conventional generative models are not necessarily due to the inadequacy of their architectures but rather stem from the sparsity and incomplete sampling inherent in natural sequence databases. By reintegrating experimental feedback, our method compensates for this deficiency, thereby sharpening the model's discriminative power.

At the same time, our approach has inherent limitations. Because the reintegration procedure relies on experimental data that are necessarily sampled from regions already explored by nature, the model is not readily extended to completely novel areas of sequence space. In other words, while our method is highly effective at refining and navigating the known functional landscape, it remains dependent on the availability and quality of experimental measurements. This dependency emphasizes that the generation of truly novel sequences will continue to require a comprehensive experimental framework to guide and validate model predictions.

Looking ahead, the iterative interplay between experimental feedback and model refinement presents a promising route toward more accurate and targeted design strategies. Future studies could explore how successive rounds of data integration may gradually expand the functional sequence space, while also addressing potential trade-offs between accuracy and diversity. Ultimately, our results suggest that a synergistic integration of experimental and computational approaches can overcome the false-positive limitations of current generative models and pave the way for more reliable biomolecular design.

Acknowledgements

We acknowledge helpful discussions with Camille Lambert, Roberto Netti, Vaitea Opuu, Andrea Pagnani, Lorenzo Rosset, and Francesco Zamponi during the preparation of this work; and we thank particularly Lorenzo Rosset for providing his GPU-based adabmDCA implementation prior to publication and Vaitea Opuu for his help with the initial analysis of the experimental data using the scripts of [7].

Author contributions: Francesco Calvanese (Data curation [equal], Investigation [equal], Methodology [equal], Resources [equal], Software [equal], Visualization [equal], Writing—original draft [equal]), Giovanni Peinetti (Data curation [equal], Investigation [equal], Methodology [equal], Resources [equal], Software [equal], Visualization [equal], Writing—original draft [equal]), Polina Pavlinova (Investigation [supporting], Validation [equal], Writing—original draft [supporting]), Philippe Nghe (Funding acquisition [equal], Resources [equal], Supervision [equal], Validation [equal]), and Martin Weigt (Conceptualization [equal], Funding acquisition [equal], Investigation [equal], Methodology [equal], Supervision [lead], Writing—original draft [lead]).

Supplementary data

Supplementary data is available at NAR online.

Conflict of interest

None declared.

Funding

Funding to pay the Open Access publication charges for this article was provided by the group budget of corresponding author. Philippe Nghe is supported by France 2030 PEPR Origins ANR-22-EXOR-0013 and EU Horizon 2020 Grant ERC AbioEvo (101002075).

Data availability

All data and code used in this study are publicly available at Zenodo: 10.5281/zenodo.15115193.

References

1. Tian P, Louis JM, Baber JL *et al.* Co-evolutionary fitness landscapes for sequence design. *Angew Chem* 2018;**130**:5776–80. https://doi.org/10.1002/ange.201713220
2. Russ WP, Figliuzzi M, Stocker C *et al.* An evolution-based model for designing chorismate mutase enzymes. *Science* 2020;**369**:440–5. https://doi.org/10.1126/science.aba3304
3. McGee F, Hauri S, Novinger Q *et al.* The generative capacity of probabilistic protein sequence models. *Nat Commun* 2021;**12**:6302. https://doi.org/10.1038/s41467-021-26529-9
4. Hawkins-Hooker A, Depardieu F, Baur S *et al.* Generating functional protein variants with variational autoencoders. *PLoS Comput Biol* 2021;**17**:e1008736. https://doi.org/10.1371/journal.pcbi.1008736

5. Lyu S, Sowlati-Hashjin S, Garton M. Variational autoencoder for design of synthetic viral vector serotypes. *Nat Mach Intell* 2024;**6**:147–60. https://doi.org/10.1038/s42256-023-00787-2
6. Alvarez S, Nartey CM, Mercado N *et al.* *In vivo* functional phenotypes from a computational epistatic model of evolution. *Proc Natl Acad Sci* 2024;**121**:e2308895121. https://doi.org/10.1073/pnas.2308895121
7. Lambert CN, Opuu V, Calvanese F *et al.* Exploring the space of self-reproducing ribozymes using generative models. *Nat Commun* 2025;**16**:7836. https://doi.org/10.1038/s41467-025-63151-5
8. Madani A, Krause B, Greene ER *et al.* Large language models generate functional protein sequences across diverse families. *Nat Biotechnol* 2023;**41**:1099–106. https://doi.org/10.1038/s41587-022-01618-2
9. de Cossio-Diaz JF, Hardouin P, du Moutier FXL *et al.* Designing molecular RNA switches with Restricted Boltzmann machines. bioRxiv, https://doi.org/10.1101/2023.05.10.540155, 20 April 2024, preprint: not peer reviewed.
10. Di Bari L, Bisardi M, Cotogno S *et al.* Emergent time scales of epistasis in protein evolution. *Proc Natl Acad Sci USA* 2024;**121**:e2406807121. https://doi.org/10.1073/pnas.2406807121
11. De Juan D, Pazos F, Valencia A. Emerging methods in protein co-evolution. *Nat Rev Genet* 2013;**14**:249–61. https://doi.org/10.1038/nrg3414
12. Ziegler C, Martin J, Sinner C *et al.* Latent generative landscapes as maps of functional diversity in protein sequence space. *Nat Commun* 2023;**14**:2222. https://doi.org/10.1038/s41467-023-37940-w
13. Kalvari I, Nawrocki EP, Ontiveros-Palacios N *et al.* Rfam 14: expanded coverage of metagenomic, viral and microRNA families. *Nucleic Acids Res* 2020;**49**:D192–200. https://doi.org/10.1093/nar/gkaa1047
14. Mistry J, Chuguransky S, Williams L *et al.* Pfam: the protein families database in 2021. *Nucleic Acids Res* 2021;**49**:D412–9. https://doi.org/10.1093/nar/gkaa913
15. Nawrocki EP, Eddy SR. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics* 2013;**29**:2933–5. https://doi.org/10.1093/bioinformatics/btt509
16. Finn RD, Clements J, Eddy SR. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res* 2011;**39**:W29–37. https://doi.org/10.1093/nar/gkr367
17. Frey NC, Hötzel I, Stanton SD *et al.* Lab-in-the-loop therapeutic antibody design with deep learning. bioRxiv, https://doi.org/10.1101/2025.02.19.639050, 15 March 2025, preprint: not peer reviewed.
18. Ouyang L, Wu J, Jiang X *et al.* Training language models to follow instructions with human feedback. *Adv Neural Inf Process Syst* 2022;**35**:27730–44.
19. Muntoni AP, Pagnani A, Weigt M *et al.* adabmDCA: adaptive Boltzmann machine learning for biological sequences. *BMC Bioinformatics* 2021;**22**:528. https://doi.org/10.1186/s12859-021-04441-9
20. Cuturello F, Tiana G, Bussi G. Assessing the accuracy of direct-coupling analysis for RNA contact prediction. *RNA* 2020;**26**:637–47. https://doi.org/10.1261/rna.074179.119
21. Barrat-Charlaix P, Figliuzzi M, Weigt M. Improving landscape inference by integrating heterogeneous data in the inverse Ising problem. *Sci Rep* 2016;**6**:37812. https://doi.org/10.1038/srep37812
22. Cocco S, Feinauer C, Figliuzzi M *et al.* Inverse statistical physics of protein sequences: a key issues review. *Rep Prog Phys* 2018;**81**:032601. https://doi.org/10.1088/1361-6633/aa9965
23. Figliuzzi M, Barrat-Charlaix P, Weigt M. How pairwise coevolutionary models capture the collective residue variability in proteins? *Mol Biol Evol* 2018;**35**:1018–27. https://doi.org/10.1093/molbev/msy007
24. Calvanese F, Lambert CN, Nghe P *et al.* Towards parsimonious generative modeling of RNA families. *Nucleic Acids Res* 2024;**52**:5465–77. https://doi.org/10.1093/nar/gkae289
25. Kerpedjiev P, Hammer S, Hofacker IL. Forna (force-directed RNA): simple and effective online RNA secondary structure diagrams. *Bioinformatics* 2015;**31**:3377–9. https://doi.org/10.1093/bioinformatics/btv372
26. Lorenz R, Bernhart SH, zu Siederdissen CH *et al.* ViennaRNA Package 2.0. *Algorithm Mol Biol* 2011;**6**:26. https://doi.org/10.1186/1748-7188-6-26
27. Mathews DH, Disney MD, Childs JL *et al.* Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc Natl Acad Sci USA* 2004;**101**:7287–92. https://doi.org/10.1073/pnas.0401799101
28. Assmann SM, Chou HL, Bevilacqua PC. Rock, scissors, paper: how RNA structure informs function. *Plant Cell* 2023;**35**:1671–707. https://doi.org/10.1093/plcell/koad026
29. Spitale RC, Incarnato D. Probing the dynamic RNA structurome and its functions. *Nat Rev Genet* 2023;**24**:178–96. https://doi.org/10.1038/s41576-022-00546-w
30. Barton JP, Chakraborty AK, Cocco S *et al.* On the entropy of protein families. *J Stat Phys* 2016;**162**:1267–93. https://doi.org/10.1007/s10955-015-1441-4
31. Rosset L, Netti R, Muntoni AP *et al.* adabmDCA 2.0—a flexible but easy-to-use package for direct coupling analysis. arXiv, https://arxiv.org/abs/2501.18456, 30 January 2025, preprint: not peer reviewed.

Received: April 23, 2025. **Revised:** July 12, 2025. **Editorial Decision:** August 1, 2025. **Accepted:** August 7, 2025