What is a meaningful representation of protein sequences?

Nicki Skafte Detlefsen
Section for Cognitive Systems
Technical University of Denmark
nsde@dtu.dk

Søren Hauberg
Section for Cognitive Systems
Technical University of Denmark
sohau@dtu.dk

Wouter Boomsma
Department of Computer Science
University of Copenhagen
wb@di.ku.dk

Abstract

How we choose to represent our data has a fundamental impact on our ability to subsequently extract information from them. Machine learning promises to automatically determine efficient representations from large unstructured datasets, such as those arising in biology. However, empirical evidence suggests that seemingly minor changes to these machine learning models yield drastically different data representations that result in different biological interpretations of data. This begs the question of what even constitutes the most meaningful representation. Here, we approach this question for representations of protein sequences, which have received considerable attention in the recent literature. We explore two key contexts in which representations naturally arise: transfer learning and interpretable learning. In the first context, we demonstrate that several contemporary practices yield suboptimal performance, and in the latter we demonstrate that taking representation geometry into account significantly improves interpretability and lets the models reveal biological information that is otherwise obscured.

Introduction

Data representations play a crucial role in the statistical analysis of biological data, and results have lingered on appropriate choices of representation. It is therefore not surprising to see an uprise in biology of *representation learning* [1], a subfield of machine learning where the representation is estimated alongside the statistical model. In the analysis of protein sequences in particular, the last years have produced a number of studies that demonstrate how representations can help extract important biological information automatically from the millions of observations acquired through modern sequencing technologies [2–14]. While these promising results indicate that learned representations can have substantial impact on scientific data analysis, they also beg the question: *what is a good representation?* This elementary question is the focus of this paper.

At its core, a *representation*[1] is a distillation of data into an abstract and often lower dimensional space that captures the essential features of the original data. This can subsequently be used for data exploration, e.g. visualization, or task-specific predictions where limited data is available. The classical principal component analysis (PCA) [15] learns features that are linearly related to the original data, while contemporary techniques seek highly non-linear relations [1]. This has been particularly successful in natural language processing (NLP), where representations of word sequences are learned from vast online textual resources, extracting general properties of language that support specific language tasks [16–18]. The success of such *word sequence models* has inspired its use for modelling *biological sequences*, leading to impressive results in application areas such as remote homologue detection [19], function classification [20] and prediction of mutational effects [6].

Since representations are becoming an important part of biological sequence analysis, we should think critically about whether the constructed representations efficiently capture the information we desire. This paper discusess this topic, with focus on protein sequences, although many of the insights apply to other biological sequences as well [13]. Our work consists of two parts. First, we consider representations in the transfer-learning setting. We investigate the impact of network design and training protocol on the resulting representation, and find that several current practices are suboptimal. Second, we investigate the use of representations for the purpose of data interpretation. We show that explicit modeling of the *representation geometry* allows us to extract robust and identifiable biological conclusions. Our results demonstrate a clear potential for *designing* representations actively, and for *analyzing* them appropriately.

[1] We note that 'representation', 'feature' and 'embedding' all seem to be used interchangeably in the literature.

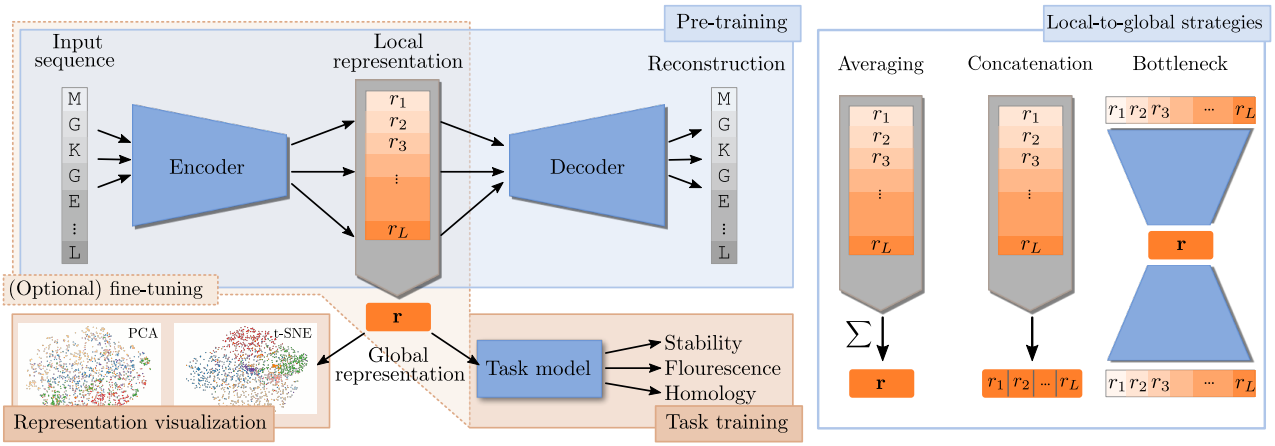

Figure 1: Representations of protein sequences. During the *pre-training* phase, a model is trained to *embed* or *encode* input protein sequences $(s_1, s_2, ..., s_L)$, to a local representation $(r_1, r_2, ..., r_L)$, after which it is *decoded* to be as similar as possible to the original sequence. After the pre-training stage, the learned representation can be used as a proxy for the raw input sequence, either for direct visual interpretation, or as input to a supervised model trained for a specific task (transfer-learning). When working in the transfer-learning setting, it is possible to also update the parameters of the encoder while training on the specific task, thereby fine-tuning the representation to the task of interest. For interpretation or for prediction of global properties of proteins, the local representations r_i, are aggregated into a global representation, often using a simple procedure such as averaging over the sequence length. For visualization purposes these global representations are then often dimensionality reduced using standard procedures such as PCA or t-SNE.

Results

Representation learning has at least two uses: In *transfer learning* we seek a representation that improves a downstream task, and in *data interpretation* the representation should reveal the patterns underlying data, e.g. through visualization. Since the first has been at the center of recent literature [4, 5, 8–10, 20, 21], we place our initial focus there, and return later to data interpretation.

Representations for transfer learning

Transfer learning addresses the problems caused by limited access to labelled data. For instance, when predicting the stability of a given protein, we only have limited training data available as it is experimentally costly to measure stability. The key idea is to leverage the many available unlabelled protein sequences to learn (*pre-train*) a general protein representation through an *embedding model*, and then train a problem-specific *task model* on top using the limited labelled training data (Figure 1).

In the protein setting, learning representations for transfer-learning can be implemented at different scopes. It can be addressed at a global scope, where representations are learned to reflect general properties of all proteins, or it can be implemented at the scope of an individual protein family, where an embedding model is pre-trained only on closely related sequences. Initially, we will focus on global setting, but will return to family specific models in the second half of the paper.

When considering representations in the transfer-learning setting, the quality, or *meaningfulness*, of a representation is judged merely by the level of predictive performance obtained by one or more downstream tasks. Our initial task will therefore be to study how this performance depends on common modelling assumptions. A recent study established a benchmark set of predictive tasks for protein sequence representations [5]. For our experiments below, we will consider three of these tasks, each reflecting a particular global protein property: 1) classification of protein sequences into a set of 1,195 known folds [22], 2) fluorescence prediction for variants of the green fluorescent protein in *Aequorea victoria* [23], and 3) prediction of the stability of protein variants obtained in high throughput experimental design experiments [24].

Fine-tuning can be detrimental to performance In the transfer-learning setting, the pre-training phase and the task learning phase are conceptually considered separately (Figure 1, left), but it is common practice to fine-tune the embedding model for a given task, which implies that the pa-

	Remote Homology			Fluorescence			Stability		
	Resnet	LSTM	Trans	Resnet	LSTM	Trans	Resnet	LSTM	Trans
Pre+Fix	0.27	**0.37**	0.27	0.23	**0.74**	0.48	0.65	0.70	0.62
Pre+Fin	0.17	0.26	0.21	0.21	0.67	0.68	**0.73**	0.69	**0.73**
Rng+Fix	0.03	0.10	0.04	0.25	0.63	0.14	0.21	0.61	-
Rng+Fin	0.10	0.12	0.09	-0.28	0.21	0.22	0.61	0.28	-0.06
Baseline	0.09 (Accuracy)			0.14 (Correlation)			0.19 (Correlation)		

Table 1: The impact of fine-tuning and initialization on downstream model performance. The embedding models were either randomly initialized (Rng) or pre-trained (Pre), and subsequently either fixed (Fix) or fine-tuned to the task (Fin). Although fine-tuning is beneficial on some task/model combinations, we see clear signs of overfitting in the majority of cases (best results in bold).

rameters of both models are in fact optimized jointly [5]. Given the large number of parameters typically employed in embedding models, we hypothesize that this can lead to overfitted representations, at least in the common scenario where only limited data is available for the task learning phase.

To test this hypothesis we train three models, an LSTM [25], a Transformer [26], and a dilated residual network (Resnet) [27], where we either keep the embedding model fixed (Fix) or fine-tune it to the task (Fin). To evaluate the impact of the representation model itself, we consider both a pre-trained version (Pre) and randomly initialized representation models that are not trained on data (Rng). Such models will map similar inputs to similar representations, but should otherwise not perform well. Finally, as a naive baseline representation, we consider the frequency of each amino acid in the sequence. In all cases, we extract global representations using an attention-based averaging over local representations (Figure 1, right).

Table 1 shows that fine-tuning the embedding clearly reduces test performance in two out of three tasks, confirming that fine-tuning can have significant detrimental effects in practice. Incidentally, we also note that the randomly initialized representation performs remarkably well in several cases, which echos results known from random projections [28].

Implication: fine-tuning a representation to a specific task carries the risk of overfitting, since it often increases the number of free parameters substantially, and should therefore take place only under rigorous cross validation. Fixing the embedding model during task-training should be the default choice.

Constructing a global representation as an average of local representations is suboptimal. One of the key modelling choices for biological sequences is how to handle their sequential nature. Inspired by developments in natural language processing, most of the recent representation learning advances for proteins use language models, which aim to reproduce their own input, either by predicting the next character given the sequence observed so far, or by predicting the entire sequence from a partially obscured input sequence. The representation learned by such models is a sequence of *local representations* $(r_1, r_2, ..., r_L)$ each corresponding to one amino acid in the input sequence $(s_1, s_2, ..., s_L)$. To successfully predict the next amino acid, r_i should contain information about the local neighborhood around s_i, together with some global signal reflecting properties of the complete sequence. In order to obtain a global representation of the entire protein, the variable number of local representations must be aggregated into a fixed-size global representation. *A priori*, we would expect this choice to be quite critical to the nature of the resulting representation. Standard approaches for this operation include averaging with uniform [4, 10] or learned attention [5, 29, 30] weights or just using the maximum activations. However, the complex non-local interactions known to occur in a protein suggest that it could be beneficial to allow for more complex aggregation functions. To investigate this issue, we consider two alternative strategies (Figure 1, right):

The first strategy (Concat) avoids aggregation altogether by *concatenating* the local representations $r = [r_1, r_2, ..., r_L, p, p, p]$ (with additional padding p to adjust for variable sequence-length). This approach preserves all information stored in the local r_is. To make a fair comparison to the averaging strategy, we maintain the same overall representation size by scaling down the size of of the local representations r_i. In our case, with a global representation size of 2048, and a maximal sequence length of 512, this means that we restrict the local representation to only four dimensions.

As a second strategy (Bottleneck), we investigate the possibility of *learning* the optimal aggregation

operation, using an autoencoder, a simple neural network that as output predicts its own input, but forces it through a low-dimensional bottleneck [31]. The model thus *learns* a generic global representation during pre-training, in contrast to the strategies above in which the global representation arises as a deterministic operation on the learned local representations. We implement the Bottleneck strategy within the Resnet (convolutional) setting, where we have well-defined procedures for down- and upsamling the sequence length.

When comparing the two proposed aggregation strategies on the three protein prediction tasks (Stability, Fluorescence, Remote Homology), we observe a quite dramatic impact on performance (Table 2). The Bottleneck strategy, where the global representation is learned, clearly outperforms the other strategies. This was expected, since already during pre-training this model is encouraged to find a more global structure in the representations. More surprising are the results for the Concat strategy, as these demonstrate that even if we restrict the local representation to be much smaller than in standard sequential models, the fact that there is no loss of information during aggregation has a significant positive influence on the downstream performance.

Implication: if a global representation of proteins is required, it should be learned rather than calculated as an average of local representations.

	Stability (Corr.)	Fluorescence (Corr.)	Homology (Acc.)
Mean	0.42	0.19	0.27
Attention	0.65	0.23	0.27
Light Att.	0.66	0.23	0.27
Maximum	0.02	0.02	0.28
MeanMax	0.37	0.15	0.26
KMax	0.10	0.11	0.27
Concat	0.74	0.69	0.34
Bottleneck	**0.79**	**0.78**	**0.41**

Table 2: Comparison of strategies for obtaining global, sequence-length independent representations on three downstream tasks [5]. The first six are variants of averaging used in the literature, using uniform weights (Mean), some variant of learned attention weights (Attention [5], Light Attention [29]), or averages of the local representation with the highest attention weight (Maximum, MeanMax, KMax(K=5)). They all use the same pretrained and backbone Resnet model, while the last two entries use modified Resnet architectures using either a very low-dimensional feature representation (Concat), or an autoencoder-like structure downsample the representation length. In all cases, training proceeded without fine-tuning. The results demonstrate that simple alternatives such as concatenating smaller local representations (Concat) or changing the model to directly learn a global representation (Bottleneck) can have a substantial impact on performance (best results in bold).

Reconstruction error is not a good measure of representation quality. Any choice of embedding model will have a number of hyper parameters, such as the number of nodes in the neural network or the dimensionality of the representation itself. How do we choose such parameters? A common strategy is to make these choices based on the reconstruction capabilities of the embedding model, but is it reasonable to expect that this is also the optimal choice from the perspective of the downstream task?

As an example, we will consider the task of finding the optimal representation size. We trained and evaluated several Bottleneck Resnet models with varying representation dimensions and applied them to the three downstream tasks. The results show a clear pattern where the reconstruction accuracy increases monotonically with latent size, with the sharpest increase in the region of 10 to 500, but with marginal improvements all the way up to the maximum size of 10000 (Figure S1). However, if we consider the three downstream tasks, we see that the performance in all three cases starts decreasing at around size 500-1000, thus showing a discrepancy between the optimal choice with respect to the reconstruction objective and the downstream task objectives. It is important to stress that the reconstruction accuracy is measured on a validation set, so what we observe is not a matter of overfitting to the training data. We employed a carefully constructed train/validation/test partition of UniProt [33] provided by Armenteros et al [21], to avoid overlap between the sets. The results thus show that there is simply enough data for the embedding model to support a large representation size, while the downstream tasks prefer a smaller input size. The exact behavior will depend on the task and the available data for the two training phases, but we can conclude that there is generally no reason to believe that reconstruction accuracy and the downstream task accuracy will agree on the optimal choice of hyperparameters. Similar findings were reported by [5].

Implication: in transfer-learning, optimal values for hyperparameters (e.g. representation size) can in general not be estimated during pre-training. They

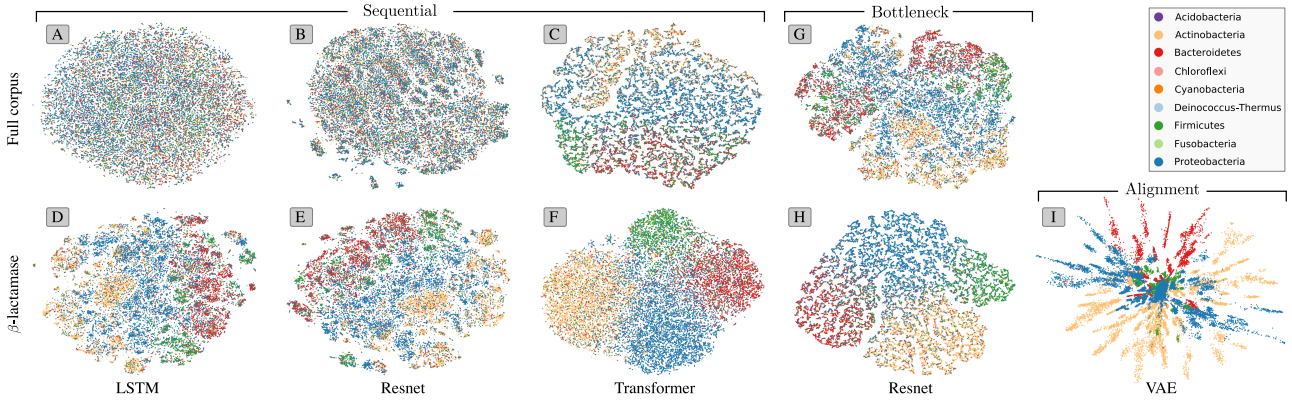

Figure 2: Latent embedding of the protein family of β-lactamase, color-coded by taxonomy at the phyla level. In the upper row, we embed the family using sequential models (LSTM, Resnet, Transformer) trained on the full corpus of protein families. In the lower row we train the same sequential models again only on the β-lactamase family (PFAM PF00144 [32]). For the models in the first three columns, a simple mean strategy is employed to extract a global representation from local representations, while the fourth column uses the Bottleneck aggregation method. Finally, in the last column, we show the result of prepossessing the sequences in a multiple sequence alignment and applying a dense variational autoencoder (VAE) model. We see clear differences in how well the different phyla is separated, which demonstrates the impact of model choice and data preprocession can have on the learned representation.

must be tuned for the specific task.

Representations for data interpretation

We now return to the use of representations for data interpretation. If a representation accurately describes structure in the underlying dataset, we might expect it to be useful not only as input to a downstream model, but also as the basis for direct interpretation, for instance through visualization. In this context, it is important to realize that different modelling choices can lead to dramatically different interpretations of the same data. More troubling, even when using the same model assumptions, repeated training instances can also deviate substantially, and we must therefore analyze our interpretations with care. In the following, we explore these effects in detail.

Representation manifolds are shaped by scope, model architecture and data preprocessing. Recent generative models for proteins tend to learn universal, cross-family representations of protein space. In bioinformatics, there is, however, a long history of analyzing proteins per family. Since the proteins in the same family share a common three-dimensional structure, an underlying correspondence exists between positions in different sequences, which we can approximate using multiple sequence alignment techniques. After establishing such an alignment, all input sequences will have the same length, making it possible to use simple fixed-size input models, rather than the sequential models discussed previously. One advantage is that models can now readily detect patterns at and correlations between absolute positions of the input, and directly observe both conservation and coevolution. In terms of interpretability, this has clear advantages. An example of this approach is the DeepSequence model [2, 12], in which the latent space of a Variational Autoencoder (VAE) was shown to clearly separate the input sequences into different phyla, and capture covariance among sites on par with earlier coevolution methods. We reproduce this result using a VAE on the β-lactamase family PF00144 from PFAM [32], using a 2 dimensional latent space (Figure 2I).

If we use the globally trained sequence models (LSTM, Resnet, Transformer) and the newly introduced Bottleneck Resnet from the previous sections to embed the same set of proteins from the β-lactamase family and use t-SNE [34] to embed the protein representations into a two dimensional space, we see no clear phylogenetic seperation in the case of LSTM and Resnet, and very little for the Transformer and the Bottleneck Resnet (Figure 2, top row).

The fact that the phyla are much less clearly resolved in these sequential models is perhaps unsurprising, since these representation have been trained to represent the space of all proteins, and therefore do not have the same capacity to separate details of a single protein family. Indeed, to compensate for

this, recent work has introduced the concept of *evo-tuning*, where a global representation is fine-tuned on a single protein family [4, 35]. When training exclusively on β-lactamase sequences (Figure 2, bottom row) we observe more structure for all models, but only the Transformer and Bottleneck Resnet are able to separate the different phyla. Comparing this to an alignment-based VAE model, we still see large differences in protein representations, despite the fact that all models now are trained on the same corpus of proteins.

The results above can be explained either by the inherent differences in the model architectures, or by the domain-specific knowledge inserted through preprocessing sequences when constructing an alignment. The different inductive biases in the four model architectures certainly play a role. For instance, sequential models have struggled to recover covariance signals in unaligned protein sequences, although recent progress has been made in this area [7, 36, 37]. We stress that it is not given that the alignment-based VAE always provides better representations. In particular, if our focus was to compare proteins across families, a global representation would be more fruitful than one trained only on a single alignment. However, for the particular objective of capturing the phylogenetic history, the VAE seems to better reflect our prior biological knowledge. Some evidence suggests that other properties, related to structure or function might be better captured with other modelling or preprocessing choices [36, 38]. Finally, we note that with exception of the VAE, all representations in Figure 2 are post-processed using t-SNE to make them amenable to plotting in 2D, which itself can have an impact on the visualization (Figure S3-S4).

Implication: the scope of data (all proteins vs. single protein families), whether data is preprocessed into alignments, the model architecture, and potential post-hoc dimensionality reduction all have fundamental impact on the resulting representations, and the conclusions we can hope to draw from them.

Representation space topology carries relevant information. The star-like structure of the VAE representation in Figure 2I, and the associated phyla color-coding strongly suggest that the topology of this particular representation space is related to the tree topology of the evolutionary history underlying the protein family [39]. As an example of the potential and limits to representation interpretability, we will proceed with a more detailed analysis of this space.

To explore the topological origin of the representation space, we estimate a phylogenetic tree of a subset of our input data ($n = 200$), and encode the inner nodes of the tree to our latent space using a standard ancestral reconstruction method (see Methods). Although the fit is not perfect – a few phyla are split and placed on opposite sides of the origin – there is generally a good correspondence (Figure 3). We see that the reconstructed ancestors to a large extent span a meaningful tree, and it is thus clear that the representation topology in this case reflects relevant topological properties from the input space.

Implication: Although neural networks are high capacity function estimators, we see empirically that topological constraints in input space are maintained in representation space. The latent manifold is thus meaningful and should be respected when relying on the representation for data interpretation.

Geometry gives robust representations. Perhaps the most exciting prospect of representation learning is the possibility of gaining new insights through interpretation and manipulation of the learned representation space. In NLP, the celebrated `word2vec` model [40] demonstrated that simple arithmetic operations on representations yielded meaningful results, e.g. "Paris - France + Italy = Rome", and similar results are known from image analysis. The ability to perform such operations on proteins would have substantial impact on protein engineering and design. However, the lack of such results in

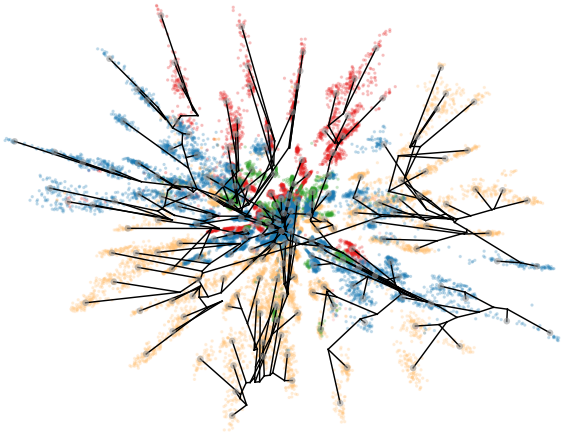

Figure 3: A phylogenetic tree encoded into the latent representation space. The representation and colors correspond to Fig. 2I. The internal nodes were determined using ancestral reconstruction after inferring a phylogenetic tree (branches encoded in black, leaf-nodes in gray).

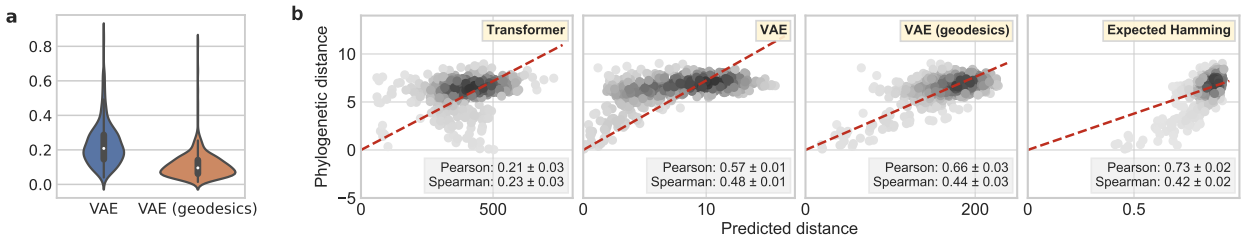

Figure 4: Geodesics provide more robust and meaningful distances in latent space. a) robustness of distances in latent space when calculated between the same data points embedded using models trained with different seeds. The plots show the distribution of standard deviations of normalized distances over five different models. b) Correlation between distances in latent space and phylogenetic distances, where standard deviations are calculated over 5 subsets of distances sampled with different seeds. Latent points were selected with probability proportial to their norm, to ensure a selection of distances covering the full range of latent space. The values for Transformer and VAE were calculated as Euclidean distances in their representation space (512 and 2 dimensional, respectively).

the literature suggests that some aspects of learned representations remain poorly understood.

To qualify the discussion, we note that standard arithmetic operations such as addition and subtraction rely on the assumption that the learned representation space is Euclidean. The star-like structure observed for the alignment-based VAE representation in Figure 3 suggests that a Euclidean interpretation may be misleading: If we define similarities between pairs of points through the Euclidean distance between them, we implicitly assume straight-line interpolants that pass through uncharted territory in the representation space when moving between 'branches' of the star-like structure. This does not seem fruitful.

Mathematically, the Euclidean interpretation is also problematic. In general, the latent variables of a generative model are not statistically identifiable, such that it is possible to deform the latent representation space without changing the estimated data density [41, 42]. The Euclidean topology is also known to cause difficulties when learning data manifolds with different topologies [43, 44]. With this in mind, the Euclidean assumption is difficult to justify beyond arguments of simplicity, as Euclidean arithmetic is not invariant to general deformations of the representation space. It has recently been pointed out that shortest paths (geodesics) and distances between representation pairs can be made identifiable even if the latent coordinates of the points themselves are not [42, 45]. The trick is to equip the learned representation with a Riemannian metric which ensures that distances are measured in data space along the estimated manifold. This result suggests that perhaps a Riemannian set of operations is more suitable for interacting with learned representations than the usual Euclidean arithmetic operators.

To investigate this hypothesis, we develop a suitable Riemannian metric, such that geodesic distances correspond to expected distances between one-hot encoded proteins, which are integrated along the manifold. The VAE defines a generative distribution $p(\mathbf{X}|\mathbf{Z})$ that is governed by a neural network. Here $\mathbf{Z}$ is a latent variable, and $\mathbf{X}$ a one-hot encoded protein sequence. To define a notion of distance and shortest path we start from a curve c in latent space, and ask what is its natural length? We parametrize the curve as $c : [0, 1] \to \mathcal{Z}$, where $\mathcal{Z}$ is the latent space, and write c_t to denote the latent coordinates of the curve at time t. As the latent space can be arbitrarily deformed it is not sensible to measure the curve length directly in the latent space, and the classic geometric approach is to instead measure the curve length after a mapping to input space [42]. For proteins, this amount to measuring latent curve lengths in the one-hot encoded protein space. Shortest paths can then be found by minimizing curve length, and a natural distance between latent points is the length of this path.

An issue with this approach is that the VAE decoder is stochastic, such that the decoded curve is stochastic as well. To arrive at a practical solution, we recall that shortest paths are also curves of minimal *energy* [42] defined as

$$\mathcal{E}(c) = \sum_{t=1}^{T-1} \|\mathbf{X}_{t+1} - \mathbf{X}_t\|^2 , \tag{1}$$

where $\mathbf{X}_t \sim p(\mathbf{X}|\mathbf{Z} = c_t)$ denote the protein sequence corresponding to latent coordinate c_t. Due to the

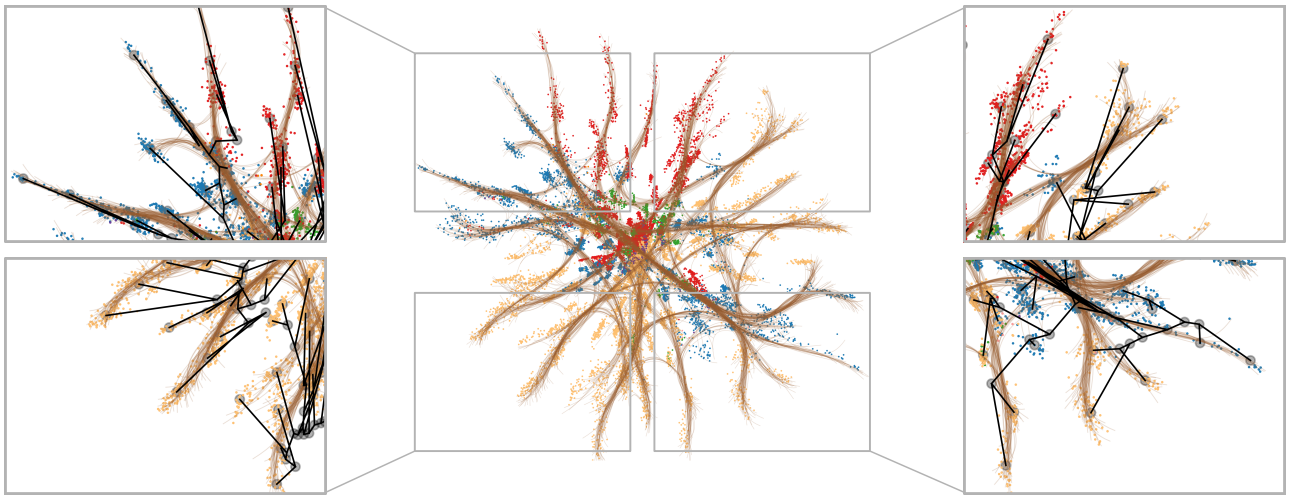

Figure 5: Shortest paths (geodesics) between representations of β-lactamase in a VAE. The Riemannian metric corresponds to measuring the expected distance between one-hot encoded proteins measured along the estimated manifold. The geodesics generally move along the star-shaped structure of the data similarly to the estimated phylogenetic tree, suggesting that the geodesics are well-suited for interpolating proteins.

stochastic decoder, the energy of a curve is a random variable. For continuous $\mathbf{X}$, recent work [45] has shown promising results when defining shortest paths as curves with minimal *expected* energy. In the Methods section we derive a similar approach for discrete one-hot encoded $\mathbf{X}$ and provide the details of the resulting optimization problem and its numerical solution.

To study the potential advantages of using geodesic over Euclidean distances, we analyze the robustness of our proposed distance. Since VAEs are not invariant to reparametrization we do not expect pairwise distances to be perfectly preserved between different initialization of the same model, but we hypothesize that the geodesics should provide greater robustness. We train the model 5 times with different seeds (see Figure S7) and calculate the same subset of pairwise distances. We normalize each set of pairwise distances by their mean and compute the distance standard deviation across trained models. When using normalized Euclidean distance we observe a mean standard deviation of 0.23, while for normalized geodesics distances we obtain a value of 0.11 (Fig. 4(a)). This significant difference indicates that geodesic distances are more robust to model retraining than their Euclidean counterparts.

Implication: distances and interpolation between points in representation space can be made robust by respecting the underlying geometry of the manifold.

Geodesics give meaning to representations. To further investigate the usefulness of geodecics, we revisit the phylogenetic analysis of Figure 3, and consider how well distances in representation space correlate with the corresponding phylogenetic distances. The first two panels of Fig. 4(b) show the correlation between 500 subsampled Euclidean distances and phylogenetic distances in a Transformer and a VAE representation, respectively. We observe very little correlation in the Transformer representation, while the VAE fares somewhat better. The third panel of Fig. 4(b) shows the correlation between *geodesic* distances and phylogenetic distances for the VAE. We observe that the geodesic distances significantly increases the linear correlation for particular short-to-medium distances. Finally, in the last panel, we include as a baseline the expected Hamming distance, i.e. latent points decoded into their categorical distribution from which we draw 10 samples/sequences and calculate the average Hamming distance. We observe that the geodesics in latent space are a reasonable proxy for this expected distance in output space.

Visually, the correspondence is also striking (Fig. 5). Well optimized geodesics follow the manifold very closely, and to a large extent preserve the underlying tree structure. We see that the irregularities described before (e.g. the incorrect placement of the yellow subtree in the top right corner) are recognized by both the phylogenetic reconstruction and our geodesics, which is visually clear by the thick bundle of geodesics running diagonally to connect these regions.

Implication: Analysing geodesics distances instead of euclidean distances in representation space better reflects the underlying manifold allowing us to extract

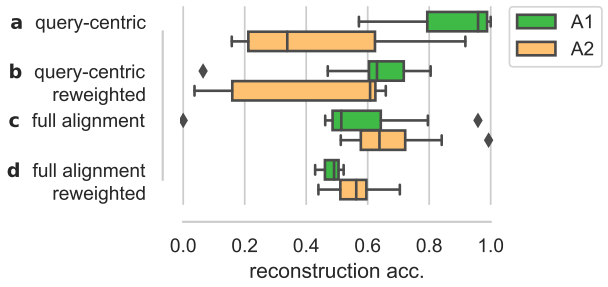

Figure 6: The effect of alignment preprocessing on the ability of representations to reliably decode back to protein sequences. Box plots (median, upper/lower quartiles, 1.5 inter-quartile range) show the distribution of reconstruction accuracies across the 14 class A1 and 13 class A2 sequences. Query-centric denotes an alignment where columns in the alignment have been removed if they contain a gap in the query sequence of interest. Reweighted refers to the standard practice of reweighting protein sequences based on similarity to other sequences. All four cases contain the same protein sequences. A1 and A2 are subclasses of beta-lactamase. A2 sequences have substantially worse representations when alignments are focused on a query from the A1 class.

biologically distances that are more meaningful.

Data preprocessing affects the geometry We have established that the preprocessing of protein sequences into an alignment has a strong effect on the learned representation. But how do alignment quality and sequence selection biases affect the learned representations? To build alignments, it is common to start with a single *query* sequence, and iterative search for sequences similar to this query. If the intent is to make statements only about this particular query sequence (e.g. predicting effects of variants relative to this protein) then a common practice is to remove columns in the alignment for which the query sequence has a gap. This query-centric bias is further enhanced by the fact that the search for relevant sequences occurs iteratively based on similarity, and is thus bound to have greater sequence coverage for sequences close to the query. These effects would suggest that representations learned from query-centric alignments might be better descriptions of sequences close to the query.

To test this hypothesis, we look at a more narrow subset of the β-lactamase family, covering only the class A β-lactamases. This subset was included as part of the DeepSequence paper [2] and will serve as our representative example of a query-centric alignment. The class A β-lactamases consist of two subclasses, A1 and A2, which are known to display consistent differences in multiple regions of the protein. The query sequence in this case is the TEM from Escherichia coli, which belongs to subclass A1. Following earlier characterization of the differences between the subclasses, we consider a set of representative sequences from each of the subclasses, and probe how they are mapped to representation space (Class A1: TEM-1, SHV-1, PSE-1, RTG-2, CumA, OXY-1, KLUA-1, CTX-M-1, NMCA, SME-1, KPC-2, GES-1, BEL-1, BPS-1. Class A2: PER-1, CEF-1, VEB-1, TLA-2, CIA-1, CGA-1, CME-1, CSP-1, SPU-1, TLA-1, CblA, CfxA, CepA). When training a representation model on the original alignment (Figure 6a), we indeed see that the ability to reconstruct (decode) meaningful sequences from representation values differs dramatically between the A1 and A2 classes.

It is common practice to weigh input sequences in alignments by their density in sequence space, which compensates for the sampling bias mentioned above [46]. While this is known to improve the quality of the model for the variant effect prediction [2], it only partially compensates for the underlying bias between the classes in our case (Figure 6b). If we instead retrieve full length sequences for all proteins, redo the alignment using standard software (Clustal Omega [47]), and maintain the full alignment length, we see that the differences between the classes becomes much smaller (Figure 6c-d). The reason is straightforward: as the distance from the query sequence increases, larger parts of a protein will occur within the regions corresponding to gaps in the query sequence. If such columns are removed, we discard more information about the distant sequences, and therefore see larger uncertainty (i.e. entropy) for the decoder of such latent values. Note that these differences in representation quality are not immediately clear through visual inspection alone (Figure S6).

Implication: Alignment-based representations depend critically on the nature of the multiple sequence alignment. In particular, training on query-centric alignments results in representations that primarily describe sequence variation around a single query sequence. In general, density based reweighting of sequences should be used to counter selection bias.

Geodesics provide more meaningful interpolation. The output distributions obtained by decoding from representation space provide interpretable insights into the nature of the represen-

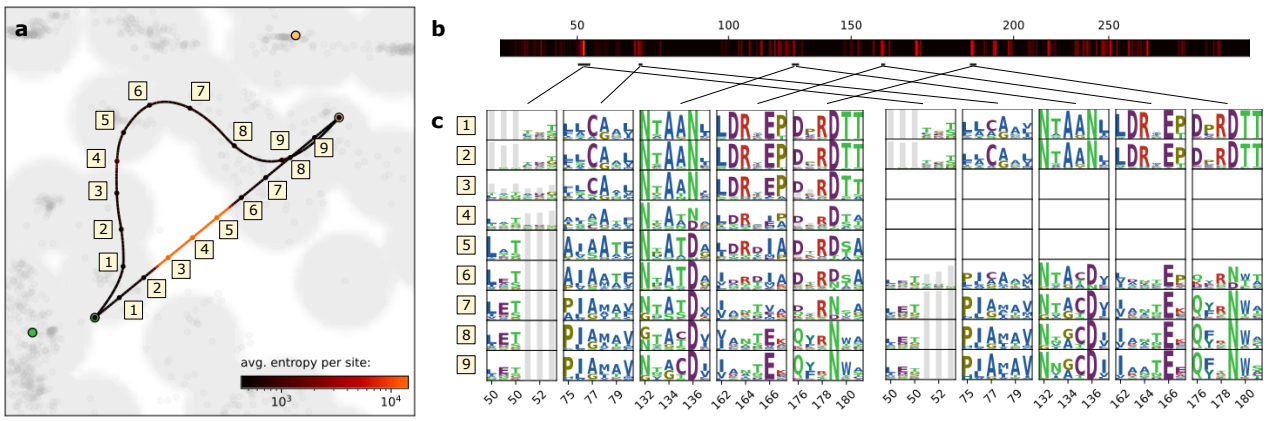

Figure 7: Interpolation between two protein sequences. a) The latent space corresponding to bottom-right representation in Fig. 6, where we have selected a sequence from β-lactamase subclass A1 and subclass A2, and the considering two interpolation paths between the points: a direct linear interpolation and one following the geodesic. The curves are color coded by the entropy of the amino acid output distribution along the path. b) The Kullback-Leibler (relative entropy) of the target sequence relative to the source for each position in the alignment. Regions with large difference (i.e. high relative entropy) are highlighted in red. c) The output distributions corresponding to several of the high-entropy regions, for the different points along the interpolant. Distributions are encoded using the weblogo standard, where the height of the letter-column at each position encodes how peaked the distribution is.

tation. We illustrate this by constructing an interpolant along the geodesic from a subclass A1 member to a subclass A2 member (Figure 7a). We calculate the entropy of the output distribution (summed over all sequence positions) along the interpolant and observe that there is a clear transition with elevated entropy around point 5 (highlighted in red). To investigate which regions of the protein are affected, we calculate the Kullback-Leibler divergence between the output distributions of the end points (Figure 7b). Zooming in on these particular regions (Figure 7c, left), and following them along the interpolant, we see that the representation naturally captures transitions between amino acid preferences at different sites. Most of these correspond to sites already identified in prior literature, for instance disappearance of the cysteine at position 77, the switch between N $\rightarrow$ D at position 136, and D $\rightarrow$ N at position 179 [48]. We also see an example where a region in one class aligns to a gap in the other (position 50-52). The linear interpolation (Figure 7c, right), has similar statistics at the endpoints, but displays an almost trivial interpolation trajectory, which effectively interpolates linearly between the probability levels of the output classes at the end points (note for instance the minor preference for cysteine in the A2 region at position 77).

Implication: Geodesics provide natural interpolants between points in representation space, avoiding high entropy regions, and thereby providing interpolated values that are better supported by data.

Discussion

Learned representations of protein sequences can substantially improve systems for making biological predictions, and may also help to reveal previously uncovered biological information. In this paper, we have illuminated parts of the answer to the titular question of what constitutes a meaningful representation of proteins. One of the conclusions is that the question itself does not have a single general answer, and must always be qualified with a specification of the purpose of the representation. A representation that is suitable for making predictions may not be optimal for a human investigator to better understand the underlying biology, and vice versa. The enticing idea of a single protein representation for all tasks thus seems unworkable in practice.

Designing purposeful representations

Designing a representation for a given task requires reflection over which biological properties we wish the representation to encapsulate. Different biological aspects of a protein will place different demands on the representations, but it is not straightforward to enforce specific properties in a representation. We can, however, steer the representation learning by 1) picking appropriate model architectures, 2) preprocessing the data, 3) choosing suitable objective

functions, and 4) placing prior distributions on parts of the model. We discuss each of these in turn.

Informed network architectures can be difficult to construct as the usual neural network 'building blocks' are fairly elementary mathematical functions that are not immediately linked to high-level biological information. Nonetheless, our discussion of length-invariant sequence representations is a simple example of how one might inform the model architecture of the biology of the task. It is generally acknowledged that global protein properties are not linearly related to local properties. It is therefore not surprising when we show that the model performance significantly improves when we allow the model to learn such a nonlinear relationship instead of relying on the common linear average of local representations. It would be interesting to push this idea beyond the Resnet architecture that we explored here, in particular in combination with the recent large scale transformer-based language models. We speculate that while similar 'low-hanging fruit' may remain in currently applied network architectures, they are limited, and more advanced tools are needed to encode biological information into network architectures. The internal representations in attention-based architectures have been shown to recover known physical interactions between proteins [36, 37], opening the door to the incorporation of prior information about known physical interactions in a protein. Recent work on permutation and rotation invariance/equivariance in neural networks [49, 50] hold promise, though they have yet to be explored exhaustively in representation learning.

Data preprocessing and feature engineering is frowned upon in contemporary 'end-to-end' representation learning, but it remains an important part of model design. In particular, preprocessing using the vast selection of existing tools from computational biology is a valuable way to encode existing biological knowledge into the representation. We saw a significant improvement in the representation capabilities of unsupervised models when trained on aligned protein sequences, as this injects prior knowledge about comparable sequence positions in a set of sequences. While recent work is increasingly working towards techniques for learning such signals directly from data [7, 36, 37, 51], it remains unclear if the advantages provided by multiple alignments can be fully encapsulated by these methods. Other preprocessing techniques, such as the reweighing of sequences, are currently also dependent on having aligned sequences. These examples suggests that if we move too fast towards 'end-to-end' learning, we risk throwing the baby out with the bathwater, by discarding years of experience endowed in existing tools.

Relevant objective functions are paramount to any learning task. Although representation learning is typically conducted using a reconstruction loss, we demonstrate that optimal representations according to this objective are generally sub-optimal for any specific transfer-learned task. This suggests that hyper-parameters of representations should be chosen based on downstream task-specific performance, rather than reconstruction performance on a hold-out set. This is, however, a delicate process, as optimizing the *parameters* of the representation model on the downstream task is associated with a high risk of overfitting. We anticipate that principled techniques for combining reconstruction objectives on the large unsupervised data sets with task specific objectives in a semi-supervised learning setting will provide substantial benefits in this area [52].

Informative priors can impose softer preferences than those encoded by hard architecture constraints. The Gaussian prior in VAEs is such an example, though its preference is not guided by biological information, which appears to be a missed opportunity. In the studies of β-lactamase, we, and others [2, 39], observe a representation structure that resembles the phylogenetic tree spanned by the evolution of the protein family. Recent hyperbolic priors [53] that are designed to emphasize hierarchies in data may help to more clearly bring forward such evolutionary structure. Since we observe that the latent representation better reflects biology when endowed with a suitable Riemannian metric, it may be valuable to use corresponding geometric priors [54].

Analysing representations appropriately

Even with the most valiant efforts to incorporate prior knowledge into our representations, they must still be interpreted with great care. We highlight the particular example of distances in representation space, and emphasize that the seemingly natural Euclidean distances are misleading. The non-linearity of encoders and decoders in modern machine learning methods means that representation spaces are generally non-Euclidean. We have demonstrated that by bringing the expected distance from the observation space into the representation space in the form of a Riemannian metric, we obtain geodesic distances that correlate significantly better with phylogenetic distances than what can be attained through the usual Euclidean view. This is an exciting result as the Riemannian view comes with a set of natural operators akin to addition and subtraction, such that

the representation can be engaged with operationally. We expect this to be valuable for e.g. protein engineering, since it gives an operational way to combine representations from different proteins.

In this study, we employed our geometric analysis only on the latent space of a variational autoencoder, which is well-suited due to its smooth mapping from a fixed dimensional latent space to a fixed dimensional output space. Expanding beyond single protein families is hindered by the fact that we cannot decode from an aggregated global representation in a sequential language model. A natural question is whether Bottleneck strategies like the one we propose could make such analysis possible. If so, it would present new possibilities for defining meaningful distances between remote homologues in latent space [19], and potentially allow for improved transfer of GO/EC annotations between proteins.

Finally, the geometric analysis comes with several implications that are relevant beyond proteins. It suggests that the commonly applied visualizations where latent representations are plotted as points on a Euclidean screen may be highly misleading. We therefore see a need for visualization techniques that faithfully reflect the geometry of the representations. The analysis also indicates that downstream prediction tasks may gain from leveraging the geometry, although standard neural network architectures do not yet have such capabilities.

Methods

Variational autoencoders. A variational autoencoder assumes that data $\mathbf{X}$ is generated from some (unknown) latent factors $\mathbf{Z}$ though the process $p_\theta(\mathbf{X}|\mathbf{Z})$. The latent variables $\mathbf{Z}$ can be viewed as the compressed representation of $\mathbf{X}$. Latent space models try to model the joint distribution of $\mathbf{X}$ and $\mathbf{Z}$ as $p_\theta(\mathbf{X}, \mathbf{Z}) = p_\theta(\mathbf{Z})p_\theta(\mathbf{X}|\mathbf{Z})$. The generating process can then be viewed as a two step procedure: first a latent variable $\mathbf{Z}$ is sampled from the prior and then data $\mathbf{X}$ is sampled from the conditional $p_\theta(\mathbf{X}|\mathbf{Z})$ (often called the decoder). Since $\mathbf{X}$ is discrete by nature, $p_\theta(\mathbf{X}|\mathbf{Z})$ is modelled as a Categorical distribution $p_\theta(\mathbf{X}|\mathbf{Z}) \sim Cat(C, l_\theta(\mathbf{Z}))$ with C classes and $l_\theta(Z)$ being the log-probabilities for each class. To make the model flexible enough to capture higher order amino acid interactions, we model $l_\theta(Z)$ as a neural network. Even though data $\mathbf{X}$ is discrete, we use continuous latent variables $\mathbf{Z} \sim N(0, 1)$.

Construction of entropy network. To ensure that our VAE decodes to high uncertainty in regions of low data density, we construct an explicit network architecture with this property. That is, the network $p_\theta(\mathbf{X}|\mathbf{Z})$ should be certain about its output in regions where we have observed data, and uncertain in regions where we have not. This has been shown to be important to get well-behaved Riemannian metrics [42, 55]. In a standard VAE with posterior modelled as a normal distribution $\mathcal{N}(\mu_\theta(\mathbf{Z}), \sigma^2_\theta(\mathbf{Z}))$, this amounts to constructing a variance network $\sigma^2_\theta(\mathbf{Z})$ that increases away from data [45, 56]. However, no prior work has been done on discrete distributions, such as the Categorical distribution $C(\mu_\theta(\mathbf{Z}))$ that we are working with. In this model we do not have a clear division of the average output (mean) and uncertainty (variance), so we control the uncertainty through the entropy of the distribution. We remind that for a categorical distribution, the entropy is

$$H(\mathbf{X}|\mathbf{Z}) = \sum_{i=1}^{C} p_\theta(\mathbf{X}|\mathbf{Z})_i \cdot \log p_\theta(\mathbf{X}|\mathbf{Z})_i.$$

The most uncertain case corresponds to when $H(\mathbf{X}|\mathbf{Z})$ is largest i.e. when $p(\mathbf{X}|\mathbf{Z})_i = 1/C$ for $i = 1, ..., C$. Thus, we want to construct a network $p_\theta(\mathbf{X}|\mathbf{Z})$ that assigns equal probability to all classes when we are away from data, but is still flexible when we are close to data. Taking inspiration from [56] we construct a function $\alpha = T(z)$, that maps distance in latent space to the zero-one domain ($T : [0, \text{inf}) \mapsto [0, 1]$). T is a trainable network of the model, with the functional form $T(z) = \texttt{sigmoid}\big(\frac{-6.9077\beta \cdot V(z)}{\beta}\big)$ with $V(z) = \min_{j=\{1,..,K\}} \|z - c_j\|_2^2$, where c_j are trainable cluster centers (initialized using k-means). This function essentially estimates how close a latent point z is to the data manifold, returning 1 if we are close and 0 when far away. Here K indicates the number of cluster centers (hyperparameter) and β is a overall scaling (trainable, constrained to the positive domain). With this network we can ensure a well-calibrated entropy by picking

$$p_\theta(\mathbf{X}|\mathbf{Z})_i = \alpha \cdot p_\theta(\mathbf{X}|\mathbf{Z})_i + (1 - \alpha) \cdot \mathbb{L},$$

where $\mathbb{L} = \frac{1}{C}$. For points far away from data, we have $\alpha = 0$ and return $\mathbb{L}$ regardless of category (class), giving maximal entropy. When near the data, we have $\alpha = 1$ and the entropy is determined by the trained decoder $p_\theta(\mathbf{X}|\mathbf{Z})_i$.

Figure 8 shows the difference in entropy of the likelihood between a standard VAE (top) and a VAE equipped with our developed entropy network (bottom). The standard VAEs produce arbitrary entropy, and is often more confident in its predictions far away from the data. Our network increases entropy as we move away from data.

Distance in sequence space. To calculate geodesic distances we first need to define geodesics over the random manifold defined by $p(\mathbf{X}|\mathbf{Z})$. These geodesics are curves c that minimize expected *energy* [42] defined as

$$\bar{\mathcal{E}}(c) = \mathbb{E}[\mathcal{E}(c)] = \int_0^1 \mathbb{E}\left[\|\partial_t \mathbf{X}_t\|^2\right] \, \mathrm{d}t, \tag{2}$$

where $\mathbf{X}_t \sim p(\mathbf{X}|\mathbf{Z} = c_t)$ is the decoding of a latent point c_t along the curve c. This energy requires a meaningful (squared) norm in data space. We remind here that

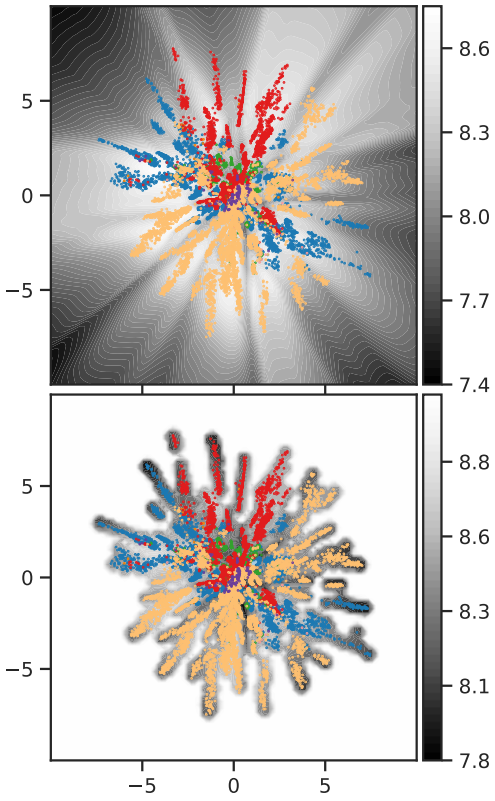

Figure 8: Construction of the entropy network for our geodesic calculations. Top: latent representations of β-lactamase with the background color denoting the entropy of the output posterior for a standard VAE. Bottom: as top but using a VAE equipped with our developed entropy network.

protein sequence data x, y is embedded into a one-hot space i.e.

$$x, y \in \left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\},$$

where we assume that $p(x_d = 1) = a_d$, $p(y_d = 1) = b_d$ for $d = 1, ..., C$. It can easily be shown that the squared norm between two such one-hot vectors can either be 0 or 2:

$$\Delta^2 = \|x - y\|^2 = \{0, 2\}.$$

The probability of these two events are given as

$$\begin{aligned} P(\Delta^2 = 0) &= P(x = y) \\ &= P(x_1 = y_1) + P(x_2 = y_2) + ... + P(x_D = y_D) \\ &= \sum_{d=1}^{C} a_d b_d, \end{aligned}$$

$$P(\Delta^2 = 2) = 1 - P(\Delta^2 = 0) = 1 - \sum_{d=1}^{C} a_d b_d.$$

The expected squared distance is then given by

$$\begin{aligned} \mathbb{E}(\Delta^2) &= \int_{\{0,2\}} \Delta^2 \cdot P(\Delta^2) d\Delta^2 \\ &= 0 \cdot P(\Delta^2 = 0) + 2 \cdot P(\Delta^2 = 2) \\ &= 2 \left(1 - \sum_{d=1}^{C} a_d b_d \right), \end{aligned}$$

Extending this measure to two sequences of length L is then

$$\mathbb{E}(\Delta^2) = \sum_{l=1}^{L} 2 \left(1 - \sum_{d=1}^{C} a_{l,d} b_{l,d} \right). \tag{3}$$

The energy of a curve, can then be evaluated by integrating this sequence measure (3) along the given curve,

$$\bar{\mathcal{E}}(c) \approx 2 \sum_{i=1}^{N-1} \sum_{l=1}^{L} \left(1 - \sum_{d=1}^{C} p(c_i)_{l,d}\, p(c_{i+1})_{l,d} \right) \Delta t, \tag{4}$$

where $\Delta t = \|c_{i+1} - c_i\|_2$. Geodesics can then be found by minimizing this energy (4) with respect to the unknown curve c. For an optimal curve c, its length is given by $\sqrt{\bar{\mathcal{E}}(c)}$.

Optimizing geodesics. In principal, the geodesics could be found by direct minimization of the expected energy. However, empirically we observed that this strategy was prone to diverge, since the optimization landscape is very flat near the initial starting point. We therefore instead discretise the entropy landscape into a 2D grid, and form a graph based on this. In this graph each node will be a point in the grid, which is connected to its eight nearest neighbours, with the edge weight being the distance weighted with the entropy. Then using Dijkstra's algorithm [57] we can very fast get a robust initialization of each geodesic. To get the final geodesic curve we fit a cubic spline [58] to the discretized curve found by Dijkstra's algorithm, and afterwards do 10 gradient steps over the spline coefficients with respect to the curve energy (4) to refine the solution.

Phylogeny and ancestral reconstruction. The $n = 200$ points used for the ancestral reconstruction where chosen as latent embeddings from the training set that were closest to the trainable cluster centers $\{c_i\}_{i=1}^{n}$ found during the estimation of the entropy network. We used FastTree2 [59] with standard settings for estimation of phylogenetic trees and subsequently applied the codeml program [60] from the PAML package for ancestral reconstruction of the internal nodes of the tree.

Acknowledgements. This work was funded in part by the Novo Nordisk Foundation through the Center for Basic Machine Learning Research in Life Science (NNF20OC0062606). It also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (757360). NSD and SH were supported in part by a research grant (15334) from VILLUM FONDEN. WB was

supported by a project grant from the Novo Nordisk Foundation (NNF18OC0052719). We thank Ole Winther, Jesper Ferkinghoff-Borg, and Jesper Salomon for feedback on earlier versions of this manuscript. Finally, we gratefully acknowledge the support of NVIDIA Corporation with the donation of GPU hardware used for this research.

Author contributions. NSD, SH and WB jointly conceived and designed the study. NSD and WB conducted the experiments. All authors contributed to the writing of the paper.

Data availability. All data used in this manuscript originates from publicaly available databases. The specific sequence data for pre-training and data for the different protein task can be found online: `https://github.com/songlab-cal/tape`. Data for the β-lactamase familie can be found here: `https://pfam.xfam.org/family/PF00144`. Preprocessed data is available through the scripts provided in our code repository.

Code availability. The source code for the paper is freely available online under an open source licence: `https://github.com/MachineLearningLifeScience/meaningful-protein-representations`.

References

1. Bengio, Y., Courville, A. & Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35,** 1798–1828 (2013).
2. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods* **15,** 816–822 (2018).
3. Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. in *International Conference on Learning Representations* (2019).
4. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods* **16,** 1315–1322 (2019).
5. Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P. & Song, Y. Evaluating protein transfer learning with TAPE. in *Advances in Neural Information Processing Systems* (2019), 9689–9701.
6. Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J. & Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv,* 622803 (2019).
7. Riesselman, A., Shin, J.-E., Kollasch, A., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. & Marks, D. Accelerating Protein Design Using Autoregressive Generative Models. *bioRxiv,* 757252 (2019).
8. Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F. & Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics* **20,** 723 (2019).
9. Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S. & Socher, R. Progen: Language modeling for protein generation. arXiv: `2004.03497` (2020).
10. Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Bhowmik, D. & Rost, B. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. arXiv: `2007.06225` (2020).
11. Lu, A. X., Zhang, H., Ghassemi, M. & Moses, A. Self-Supervised Contrastive Learning of Protein Representations By Mutual Information Maximization. bioRxiv: `2020.09.04.283929` (2020).
12. Frazer, J., Notin, P., Dias, M., Gomez, A., Brock, K., Gal, Y. & Marks, D. Large-scale clinical interpretation of genetic variants using evolutionary data and deep learning. bioRxiv: `2020.12.21.423785` (2020).
13. Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics* (2021).
14. Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J., Poviloniene, S., Laurynenas, A., Viknander, S., Abuajwa, W., Savolainen, O., Meskys, R., Engqvist, M. K. M. & Zelezniak, A. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence,* 1–10 (2021).
15. Jolliffe, I. Principal Component Analysis (Springer, 1986).

16. Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving Language Understanding by Generative Pre-Training. Tech. rep. (OpenAI, 2018).

17. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (2019), 4171–4186.

18. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv: 1907.11692 (2019).

19. Morton, J., Strauss, C., Blackwell, R., Berenberg, D., Gligorijevic, V. & Bonneau, R. Protein Structural Alignments From Sequence. bioRxiv: 2020.11.03.365932 (2020).

20. Gligorijevic, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C., Fisk, I. M., Vlamakis, H., Xavier, R. J., Knight, R., Cho, K. & Bonneau, R. Structure-based function prediction using graph convolutional networks. bioRxiv: 10.1101/786236 (2020).

21. Armenteros, J. J. A., Johansen, A. R., Winther, O. & Nielsen, H. Language modelling for biological sequences–curated datasets and baselines. *bioRxiv* (2020).

22. Hou, J., Adhikari, B. & Cheng, J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* **34,** 1295–1303 (2018).

23. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533,** 397–401 (2016).

24. Rocklin, G. J., Chidyausiku, T. M., Goreshnik, I., Ford, A., Houliston, S., Lemak, A., Carter, L., Ravichandran, R., Mulligan, V. K., Chevalier, A., Arrowsmith, C. H. & Baker, D. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* **357,** 168–175 (2017).

25. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Compututation* **9,** 1735–1780 (1997).

26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. Attention is all you need. in *Advances in neural information processing systems* (2017), 5998–6008.

27. Yu, F., Koltun, V. & Funkhouser, T. Dilated residual networks. in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), 472–480.

28. Bingham, E. & Mannila, H. Random projection in dimensionality reduction: applications to image and text data. in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (2001), 245–250.

29. Stärk, H., Dallago, C., Heinzinger, M. & Rost, B. Light Attention Predicts Protein Location from the Language of Life. bioRxiv: 2021.04.25.441334 (2021).

30. Monteiro, J., Alam, M. J. & Falk, T. On The Performance of Time-Pooling Strategies for End-to-End Spoken Language Identification. English. in *Proceedings of the 12th Language Resources and Evaluation Conference* (European Language Resources Association, 2020), 3566–3572.

31. Kramer, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal* **37,** 233–243 (1991).

32. El-Gebali, S., Mistry, J., Bateman, A., Eddy, S. R., Luciani, A., Potter, S. C., Qureshi, M., Richardson, L. J., Salazar, G. A., Smart, A., Sonnhammer, E. L. L., Hirsh, L., Paladin, L., Piovesan, D., Tosatto, S. C. E. & Finn, R. D. The Pfam protein families database in 2019. *Nucleic Acids Research* **47,** D427–D432. eprint: https://academic.oup.com/nar/article-pdf/47/D1/D427/27436497/gky995.pdf (2018).

33. Consortium, T. U. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research* **47,** D506–D515. eprint: https://academic.oup.com/nar/article-pdf/47/D1/D506/27437297/gky1049.pdf (2018).

34. Van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).

35. Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M. & Church, G. M. Low-N protein engineering with data-efficient deep learning. *Nature Methods* **18,** 389–396 (2021).

36. Rao, R., Ovchinnikov, S., Meier, J., Rives, A. & Sercu, T. Transformer protein language models are unsupervised structure learners. bioRxiv: 2020.12.15.422761 (2020).

37. Vig, J., Madani, A., Varshney, L. R., Xiong, C., Socher, R. & Rajani, N. F. BERTology Meets Biology: Interpreting Attention in Protein Language Models. arXiv: 2006.15222 (2020).

38. Hawkins-Hooker, A., Depardieu, F., Baur, S., Couairon, G., Chen, A. & Bikard, D. Generating functional protein variants with variational autoencoders. *PLoS computational biology* **17,** e1008736 (2021).

39. Ding, X., Zou, Z. & Brooks, C. L. Deciphering protein evolution and fitness landscapes with latent space models. *Nature Communications* **10** (2019).

40. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. in (2013). arXiv: 1301.3781.

41. Bishop, C. M. Pattern Recognition and Machine Learning (Springer, 2006).

42. Hauberg, S. Only Bayes should learn a manifold (on the estimation of differential geometric structure from data). arXiv: 1806.04994 (2018).

43. Falorsi, L., de Haan, P., Davidson, T. R., Cao, N. D., Weiler, M., Forré, P. & Cohen, T. S. Explorations in Homeomorphic Variational Auto-Encoding. in *ICML18 Workshop on Theoretical Foundations and Applications of Deep Generative Models* (2018).

44. Davidson, T. R., Falorsi, L., Cao, N. D., Kipf, T. & Tomczak, J. M. Hyperspherical Variational Auto-Encoders. in *Uncertainty in Artificial Intelligence* (2018).

45. Arvanitidis, G., Hansen, L. K. & Hauberg, S. Latent space oddity: On the curvature of deep generative models. in *6th International Conference on Learning Representations, ICLR 2018* (2018).

46. Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M. & Aurell, E. Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E* **87** (2013).

47. Sievers, F., Wilm, A., Dineen, D., Gibson, T., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., Thompson, J. & Higgins, D. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology* **7.**

48. Philippon, A., Slama, P., Dény, P. & Labia, R. A structure-based classification of class A β-lactamases, a broadly diverse family of enzymes. *Clinical Microbiology Reviews* **29,** 29–57 (2016).

49. Cohen, T. S., Geiger, M. & Weiler, M. Intertwiners between Induced Representations (with Applications to the Theory of Equivariant Neural Networks). 2018. arXiv: 1803.10743 [cs.LG].

50. Weiler, M., Geiger, M., Welling, M., Boomsma, W. & Cohen, T. S. 3D Steerable CNNs: Learning rotationally equivariant features in volumetric data. in *Advances in Neural Information Processing Systems* (2018).

51. Rao, R., Liu, J., Verkuil, R., Meier, J., Canny, J. F., Abbeel, P., Sercu, T. & Rives, A. MSA Transformer. bioRxiv: 10.1101/2021.02.12.430858 (2021).

52. Min, S., Park, S., Kim, S., Choi, H.-S. & Yoon, S. Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information. arXiv: 1912.05625 (2019).

53. Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R. & Teh, Y. W. Continuous hierarchical representations with poincaré variational auto-encoders. in *Advances in neural information processing systems* (2019).

54. Kalatzis, D., Eklund, D., Arvanitidis, G. & Hauberg, S. Variational Autoencoders with Riemannian Brownian Motion Priors. in *Proceedings of the 37th International Conference on Machine Learning* **119** (2020).

55. Tosi, A., Hauberg, S., Vellido, A. & Lawrence, N. D. Metrics for Probabilistic Geometries. in *The Conference on Uncertainty in Artificial Intelligence (UAI)* (Quebec, Canada, 2014).

56. Skafte, N., Jørgensen, M. & Hauberg, S. Reliable training and estimation of variance networks. in *Advances in Neural Information Processing Systems* (2019).

57. Dijkstra, E. W. *et al.* A note on two problems in connexion with graphs. *Numerische mathematik* **1,** 269–271 (1959).

58. J.H. Ahlberg, E. N. & Walsh, J. L. The theory of splines and their applications. *Canadian Mathematical Bulletin* **11,** 507–508 (1968).

59. Price, M. N., Dehal, P. S. & Arkin, A. P. FastTree 2 - Approximately maximum-likelihood trees for large alignments. *PLoS One* **5,** e9490 (2010).

60. Adachi, J. & Hasegawa, M. MOLPHY version 2.3: programs for molecular phylogenetics based on maximum likelihood. **28** (Institute of Statistical Mathematics Tokyo, 1996).

Supplementary Material: What is a meaningful representation of protein sequences?

Nicki Skafte Detlefsen
Section for Cognitive Systems
Technical University of Denmark
nsde@dtu.dk

Søren Hauberg
Section for Cognitive Systems
Technical University of Denmark
sohau@dtu.dk

Wouter Boomsma
Department of Computer Science
University of Copenhagen
wb@di.ku.dk

Experimental details

Datasets. In the transfer-learning experiments we use 31 million protein sequences extracted from the Pfam database [1], following the procedure described for the TAPE benchmark set [2]. We use data for remote homology detection from [3], fluorescence landscape prediction from [4] and for stability landscape prediction from [5], all spanning multiple protein families. See Table S1 for the specific dataset sizes.

For the experiments regarding the analysis of reconstruction error as a measure of downstream performance (S1), we use the UniLanguage dataset [6]. UniLanguage consists of samples from the UniProt [7] database, where splits were constructed to minimize the overlap between families.

For the study of latent space structure on single protein families we consider the β-lactamase sequences extracted from Pfam [1], family PF00144, where we also obtain a sequence alignment. For the study of subclasses of beta-lactamase, we use the beta-lactamase alignment provided in the DeepSequence study [8], which is processed in different ways as described in the main paper. The processing scripts are provided as part of the source code repository associated with our manuscript.

Task	Train	Valid	Test
Language Mod.	32,207,059	N/A	44,314
Unilanguage [6]	607,737	98,907	295,161
Remote Homol.	12,312	736	718
Fluorescence	21,446	5,362	27,217
Stability	53,679	2,447	12,839

Table S1: Data set sizes used for the prediction tasks (i.e. the transfer-learning setting).

Predictive tasks. In the transfer-learning experiments, the Transformer and Resnet based models were pre-trained using a masked token prediction task [9] where 15% of the amino acids in a sequence are masked out and the task is to predict the identity of the masked amino acids from the non-masked. The LSTM model were trained using next-token prediction, where the task is to predict the next amino acid in a sequence given the amino acids processed until now. Lastly, the autoencoder (bottleneck) models were trained with standard reconstruction tasks. Details for the three downstream tasks are listed below:

1. Fluorescence: An input protein sequence $\mathbf{s}$ is mapped to a label $y \in \mathbb{R}$ corresponding to the log-fluorescence intensity of $\mathbf{s}$, that expresses a models ability to distinguish between similar sequences. The models are optimized using the mean squared loss and performance is measured using Spearman correlation.

2. Stability: An input protein sequence $\mathbf{s}$ is mapped to a label $y \in \mathbb{R}$ corresponding to the most extreme value for which the protein keeps its fold. The models are optimized using the mean squared loss and performance is measured using Spearman correlation.

3. Remote homology: An input protein sequence $\mathbf{s}$ is mapped to a label $y \in \{1, ..., 1195\}$, where each class correspond to a specific protein fold. The models are optimized using categorical cross entropy and performance is measured using accuracy.

Network Architectures

Resnet, LSTM, Transformer

The Resnet, LSTM and Transformer architectures used in the first experiments are all directly taken

from TAPE [2]. The Transformer consist of 12-layers with a hidden size of 512 and 8 attention heads, which leads to a 38M-parameter model. The architechtures of the other two models were chosen such that the total number of parameters match that of the Transformer. In this case, the Resnet consist of 35 layers each with 256 filters, a dilution rate of 2 and a kernel size of 9. The LSTM has 3 bidirectional layers each with 1024 hidden units. If not stated otherwise, we use an attention based aggregation function for combining the local representations into a single global representation.

Bottleneck AutoEncoder

For the Bottleneck Resnet autoencoder we used an encoder-decoder architecture where both the encoder and decoder were modelled using resnet blocks. In contrast to the three models above, the bottleneck autoencoder is not a sequential model and requires a fixed size input. All sequences were therefore padded with zeros to the same length (3000). For the encoder we use 30 residual blocks with pooling along the sequence dimension every 5 layer. For the decoder we inverse the process and again use 30 residual blocks, this time with upsampling along the sequence dimension every 5 layer. Between the encoder and decoder we had two fully connected layers that respectively downsampled and upsampled from the global latent space. The AutoEncoder has approximately twice the number of parameters as the sequential models during pre-training, but has the same number parameter when used for transfer-learning as the decoder is disabled in this setting.

VAE

The architecture of the VAE is a simplified version of that used in the DeepSequence paper [8], using an encoder with two fully connected hidden layers (1500, 1500 nodes) with ReLU activations, and a decoder with two fully connected hidden layers (100, 500 nodes) also with ReLU activations.

Training details. We followed the training protocol from [2] for pretraining on Pfam and training of the task specific models. Pre-training was performed on four NVIDIA TITAN V GPUs for 1 week. Hyperparameters were set as follows:

- Adam optimizer was used with default settings for momentum.
- Learning rate: initialized to 10^{-3}, adjusted using a linear warm-up scheduler.
- 10% dropout rate.
- Batch size was dynamically set during training to the largest possible based on model architecture and sequence length.

Task-specific training was performed using the same set of GPUs and hyperparameters, but training was stopped early when no increase in validation performance was observed. If not stated otherwise, we always complete pre-training before task-specific training to get the best possible performing model.

For the training on β-lactamase model we deploy nearly the same training stategy as with the Pfam family, however with extra steps to prevent overfitting, which is more likely on a single family than the full corpus of proteins. In particular we use early stopping monitored on the validation loss with a patience of 10 epochs to ensure that we do not use highly overfitted models.

The VAEs were trained using the same optimizer settings, but with a fixed learning rate, no dropout and using a fixed batch size of 16.

Additional results

Reconstruction accuracy is not a good proxy for downstream performance

In the main paper we observe that reconstruction accuracy may be a poor proxy for the quality of the representation itself, as it does not directly correlate with the downstream performance metrics (Fig. S1. To further investigate the phenomenon, we re-trained a number of embedding models, where we gradually

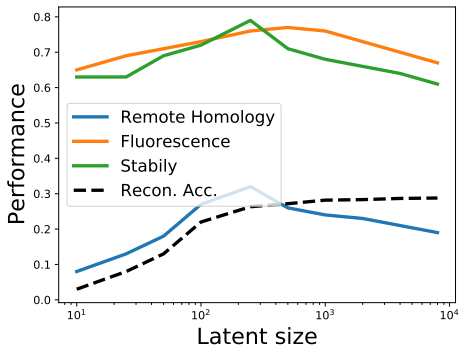

Figure S1: Reconstruction and downstream performance as a function of representation size. Although reconstruction accuracy consistently improves for increasing representation size, the performances on the individual tasks deteriorate for large representations. Performance refers to Spearman correlation (stability, fluorescence) or accuracy (homology, reconstruction).

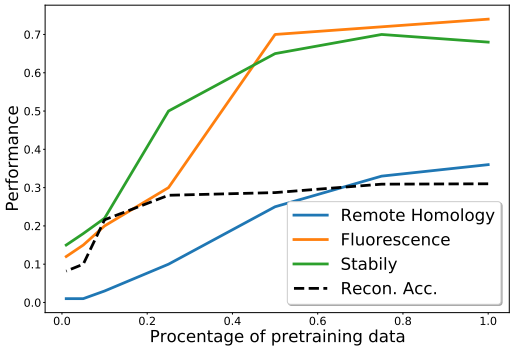

Figure S2: Reconstruction and downstream performance as a function of amount of data used during pre-training (in %). Performance refers to Spearman correlation (stability, fluorescence) or accuracy (homology, reconstruction).

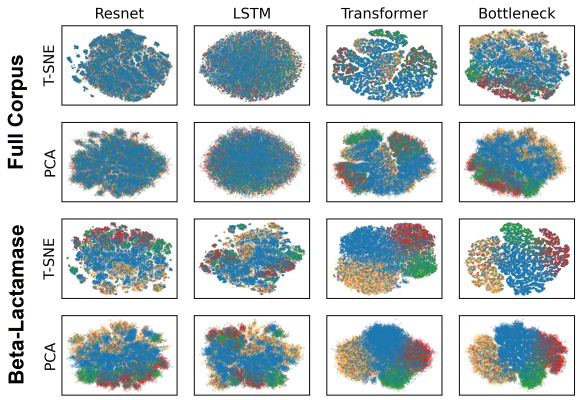

Figure S3: Impact of the choice of dimensionality reduction on the visualizations. This figures expands the results of Fig. 2 in the main paper, showing the influence of two different choices of dimensionality reduction (t-SNE as in the main paper vs PCA).

lowered the amount of pre-training data available to the model (Fig. S2).

We again observe a discrepancy between the reconstruction performance and the downstream performance metrics, with the reconstruction accuracy flattening out after seeing only 30% of the data, whereas the downstream tasks all increase with more pre-training data. This confirms the result from the main paper that the reconstruction accuracy of a model is a poor proxy for the how well the representation will perform on downstream tasks.

Impact of the choice of dimensionality reduction technique

In the main paper, we chose the latent dimension of the VAE to be 2, such that it could be directly visualized. However, for many models, it is not possible to train a model with such a low internal representation dimension, and it thus becomes necessary to employ a post-hoc dimensionality reduction. The choice of dimensionality reduction technique can have a substantial impact on the visual interpretation. In Fig. S3, we recreate the visualizations from Fig. 2 in the main text, using PCA instead of t-SNE. Although the t-SNE seems to have slightly better separation if phyla locally, the overall patterns produced by the two dimensionality reduction schemes are remarkably robust, in particular in the most specific representations in the bottom right. We also investigated the impact of dimensionality reduction for the VAE, when trained with a higher dimensional latent space of 30, and applying t-SNE or PCA to reduce it to two dimensions (Fig. S4). In this case, the PCA seems to preserve the star-like structure to some extent, while t-SNE organizes the clusters in a completely different topology, presumably due to the distributional assumptions underlying the t-SNE method.

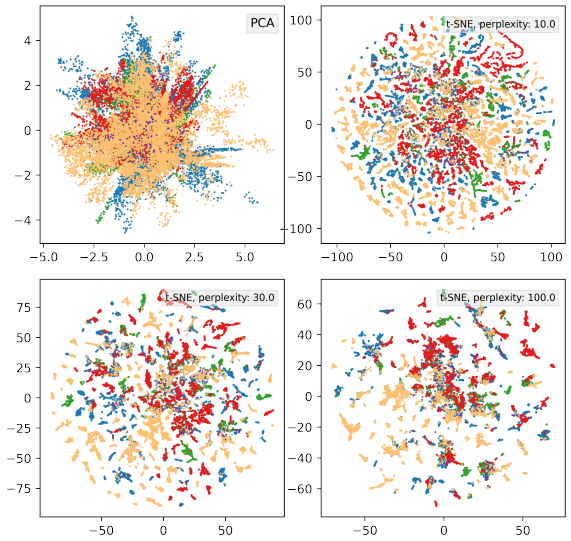

Figure S4: Dimensionality reduction on the latent space of a VAE. In contrast to the two dimensional latent space employed in the main paper, we here train a model with 30 latent dimensions. The two plots show the same latent space reduced by either PCA or t-SNE, the latter with different choices of the perplexity parameter.

VAE trained on reweighted sequence data

The VAE in Fig. 3 in the main paper was trained without conducting sequence reweighting. For completeness, we here include results from training on a dataset where the input sequences are reweighted based on input density (as described in the main

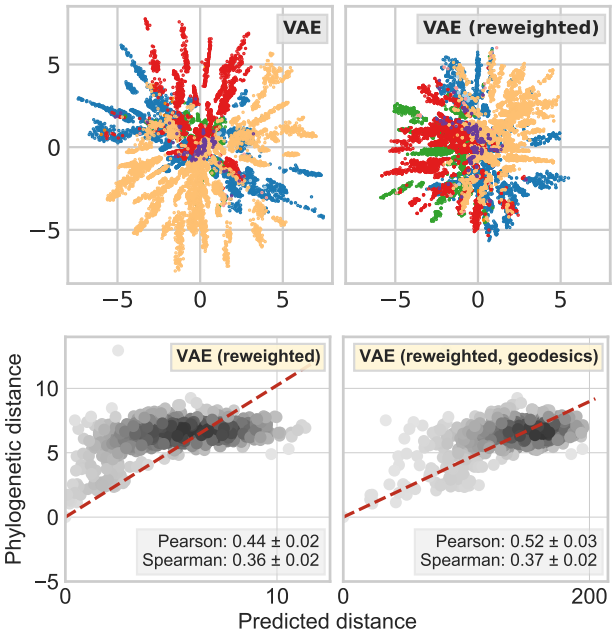

Figure S5: Representation and geodesic correlations for a VAE trained with reweighted inputs. Top row: representation plot of the VAE trained on the raw input vs one trained on reweighted data. Bottom row (left): correlations between Euclidean distances in latent space and phylogenetic distances. Bottom row (right): same as left, but now with geodesics (details in main paper).

text). As expected, we see a better balanced representation of the different phyla, most pronounced in the added weight on the green Firmicutes subtree (Fig. S5) and the even lower populated phyla that are difficult to discern in the original VAE. The trend in the correlations to phylogenetic distances are similar to those reported in the main paper, again demonstrating a benefit of employing geodesics rather than euclidean distances.

Effect of alignment preprocessing

In the main text, we also discuss the how reweighting of input data and column removal in the alignment can lead to representations with different degrees of selection bias towards a particular query sequence (Fig. 6). Visualizations of the four settings are displayed in Fig. S6.

Impact of initialization on the representation

Although it is commonly accepted that initialization of a neural network has some impact on the resulting models, the ultimate behavior and performance of a model is often fairly robust to different initializations. It is important to stress that this is

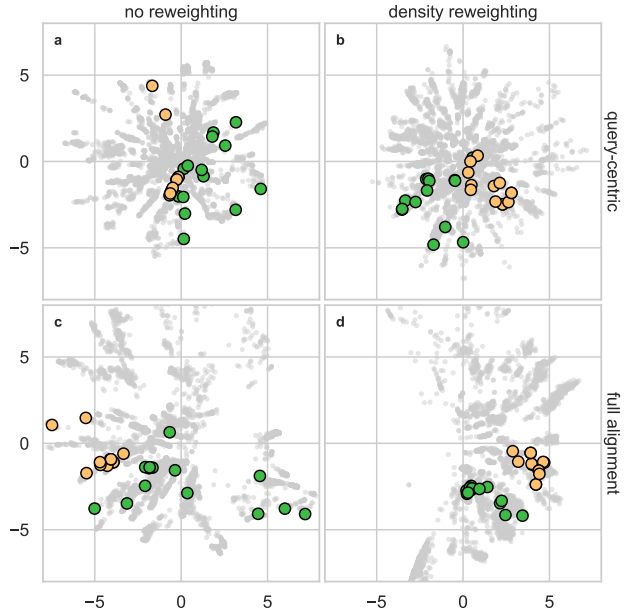

Figure S6: The effect of alignment preprocessing on the learned representation. Top row: query-centric alignment where columns are removed if they contain a gap in the query sequence. Bottom row: standard alignment of the same sequences. Left/Right column: whether the sequences are reweighted during training of the model. The green dots correspond to proteins belonging to the same subclass as the query (A1). The yellow dots belong to subclass A2, which is more distant to the query protein.

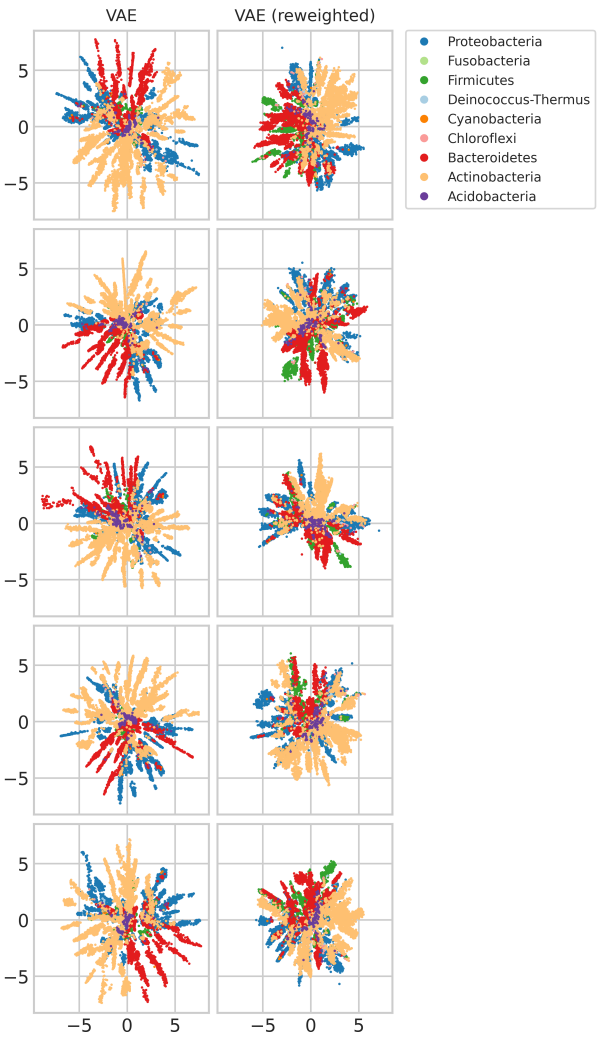

Figure S7: Different initializations of the same model. Left column: trained on raw data; right column: trained on density reweighted data. The top left plot corresponds to the model used in the main paper.

not the case for learned representations, which can change dramatically depending on the initialization, partly due to the many symmetries in parameter space. As an example, in Fig. S7 we show the representations of the β-lactamase protein family for 4 different initial seeds. While they all follow the overall tree structure, we see clear variations in the organisation of the individual branches of the tree, and we especially observe that the orientation in latent space is arbitrary. This further supports the idea that a Euclidean interpretation of the latent space can be misleading.

References

1. El-Gebali, S. *et al.* The Pfam protein families database in 2019. *Nucleic Acids Research* **47,** D427–D432. eprint: `https://academic.oup.com/nar/article-pdf/47/D1/D427/27436497/gky995.pdf` (2018).
2. Rao, R. *et al.* Evaluating protein transfer learning with TAPE. in *Advances in Neural Information Processing Systems* (2019), 9689–9701.
3. Hou, J., Adhikari, B. & Cheng, J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* **34,** 1295–1303 (2018).
4. Sarkisyan, K. S. *et al.* Local fitness landscape of the green fluorescent protein. *Nature* **533,** 397–401 (2016).
5. Rocklin, G. J. *et al.* Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* **357,** 168–175 (2017).
6. Armenteros, J. J. A., Johansen, A. R., Winther, O. & Nielsen, H. Language modelling for biological sequences–curated datasets and baselines. *bioRxiv* (2020).
7. Consortium, T. U. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research* **47,** D506–D515. eprint: `https://academic.oup.com/nar/article-pdf/47/D1/D506/27437297/gky1049.pdf` (2018).
8. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods* **15,** 816–822 (2018).
9. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (2019), 4171–4186.