DiffNets: Self-supervised deep learning to identify the mechanistic basis for biochemical differences between protein variants

Michael D. Ward[1], Maxwell I. Zimmerman[1], S.J. Swamidass[2], and Gregory R. Bowman[1,3]*

[1]Department of Biochemistry & Molecular Biophysics, Washington University School of Medicine, St. Louis, MO 63110, USA
[2]Department of Pathology & Immunology, Washington University School of Medicine, St. Louis, MO 63110, USA
[3]Center for the Science and Engineering of Living Systems, Washington University in St. Louis, St. Louis, MO 63110, USA

*To whom correspondence should be sent: g.bowman@wustl.edu

Abstract
A mechanistic understanding of how mutations modulate proteins' biochemical properties would advance our understanding of biology, provide insight for engineering proteins with particular functions, and facilitate efforts in precision medicine. However, such mechanistic insight remains elusive in many cases. For example, experimentally-derived structures of protein variants with dramatically different behaviors are often nearly identical, suggesting that one must consider the entire ensemble of structures that a protein adopts. Molecular dynamics (MD) simulations provide access to such ensembles, but identifying the relevant features of these complex entities remains difficult. Here we develop DiffNets, a deep learning framework that combines supervised autoencoders with expectation maximization to identify the structural preferences that are responsible for the biochemical differences between protein variants. As a proof of principle, we show that DiffNets identify the important structural preferences that confer increased stability to TEM β-lactamase variants without any a priori knowledge of the relevant structural features.

Intro

A mechanistic understanding of the functional differences between protein variants is crucial for advancing our understanding of fundamental biology and for applications in precision medicine. Amino acid substitutions can disrupt protein function leading to morbidity and mortality[1,2], but also have the potential to bring about new functions, like metabolizing new substrates[3]. In either case, identifying how mutations perturb the underlying structure and dynamics of a protein is the key to explaining these functional differences[4–9]. Streamlining this process would make it easier to infer the behavior of new protein variants and to identify essential elements of protein function, which would accelerate protein engineering. Further, devising therapeutic strategies for many human health challenges is easier when researchers have a structural basis to distinguish drug targets (e.g. cancerous protein variant) from other proteins.

Identifying structural features that explain functional differences between protein variants is a difficult challenge. While it is appealing to distinguish protein variants by structural differences in their ground state configurations, like with crystal structures, often the ground states are identical[10]. Moreover, this comparison neglects many thermally accessible configurations that proteins adopt, which are essential for protein function and strongly influence a protein's druggability[11–13]. Experimentally-derived protein structures can be combined with MD simulations[14] to sample conformational ensembles, but it is difficult to compare ensembles since they contain millions of overlapping states, each with thousands of degrees of freedom[15,16]. Adding to this challenge, its often the case that protein variants are only distinguishable by statistical differences in their propensity to adopt specific structural configurations, rather than by gross structural rearrangements. For example, mutations in the bacterial protein, TEM β-lactamase, help it to evolve resistance to new antibiotic drugs by changing how frequently it occupies certain types of excited states in the conformational ensemble, rather than by sampling new configurations, or changing the native folded structure[5].

Because conformational ensembles are extremely high-dimensional, analyzing MD simulation data using dimensionality reduction algorithms is crucial. However, existing dimensionality reduction algorithms make assumptions about what is important in the data that may not always be well-suited for distinguishing protein variants. For example, principal component analysis (PCA)[17,18] finds linear combinations of features that capture the most geometric variance. As a result, PCA learns a low-dimensional projection of data that often highlights large geometric differences in protein structural configurations. When PCA identifies differences between protein variants it is difficult to discern whether they are arbitrary motions, like of a floppy loop in solution, or if they explain functional differences between protein variants. Further, PCA will not uncover functional differences between variants if they are due to subtle geometric rearrangements. Autoencoders[19] are a more powerful alternative since they can find nonlinear combinations of features. Autoencoders are neural networks that learn a low-dimensional projection of data that is optimized to produce a high-fidelity geometric reconstruction of a protein configuration (Fig. 1). However, like PCA, autoencoders still focus on capturing large geometric variations. Time-lagged independent component analysis (tICA)[20,21] is another common approach, which captures kinetically slow motions in a system by identifying linear combinations of features that decorrelate slowly. Many functionally relevant structural changes, like protein folding, are kinetically slow so tICA is a reasonable approach to capture them. However, there are many situations where the conformational changes of interest are fast relative to others (e.g. allostery within the native ensemble that is faster than folding and unfolding).

We hypothesize that dimensionality reduction algorithms can best identify relevant structural differences between protein variants if they are required to predict the variants' biochemical properties. The requirement to make such a prediction constrains the dimensionality reduction, forcing data to be sorted based on underlying biochemical properties. Experimentalists operate in a similar manner; they sort protein variants based on biochemical differences and then compare and contrast structural data to find features that explain these biochemical differences. In other words, they start with the assumption that there are differences between two classes of data and then search for features that separate the classes of data, regardless of if they are slow motions, large motions, or any other kind of motions. Designing a dimensionality reduction algorithm that does this systematically in a manner that accounts for the ensemble nature of proteins would be of great value to researchers.

Here we introduce DiffNets, an approach that uses neural **net**works to identify the ensemble features that explain biochemical **diff**erences between protein variants. DiffNets use MD simulation data to learn a low-dimensional projection (latent space) of protein structures that is explicitly organized to separate structural configurations based on the biochemical property a structure confers, rather than by slow motions or large geometric changes. For example, if a mutation gives rise to a protein variant that can metabolize a new substrate, DiffNets would learn differences between the wild-type and protein variant by learning a latent space that separates structural configurations based on the degree they contribute to metabolizing the new substrate. With the latent space organized in this way it becomes easy to identify structural features that explain biochemical differences between protein variants, and therefore, to understand the structural details that govern the biochemical property of interest. DiffNets achieve this by combining supervised autoencoders[22] with expectation maximization. In contrast to standard autoencoders (unsupervised) that are singly-focused on performing a geometric dimensionality reduction of structural configurations, DiffNets are multi-task networks that perform a dimensionality reduction where the low-dimensional projection of data must be able to both reconstruct the input structure and accurately classify structures with a label that reflects some biochemical property of interest (Fig. 1).

DiffNets rely on two key innovations. First, attaching a classification task to the latent space acts as a mechanism to organize the latent space (Fig. 1). It forces the dimensionality reduction to dedicate representational power to identifying degrees of freedom that distinguish protein variants with different behaviors instead of focusing exclusively on highly variable degrees of freedom. Second, we draw inspiration from multiple-instance learning[23] to curate training labels for the classification task. During multiple-instance learning, learners are given bags of training examples where each bag is labelled negative, indicating that the bag is full of negative examples, or positive, indicating that there are at least some positive examples in the bag. The learner then must figure out how to label all of the individual instances. This is similar to our situation where we know the biochemical property of each protein variant (i.e. negative bag or positive bag), but we do not know the biochemical property of each structural configuration sampled in a simulation. To solve this problem, we train DiffNets using an expectation maximization framework that iteratively updates the classification training labels from initial labels indicating which variant simulation a structure came from to labels that indicate the biochemical property of each structural configuration. Ultimately, DiffNets learn a latent representation of MD simulation data that encodes a faithful representation of a protein's structure and sorts structures based on their association with the biochemical property of interest. DiffNets can split up the input and focus the classification task on a designated region of the protein, like the region of a mutation (Fig. 1), or the active site of an enzyme. This provides a way to guide DiffNets to search for differences at

the root (e.g. the mutation), or to incorporate *a priori* information (e.g. the mutation changes the active site of an enzyme).

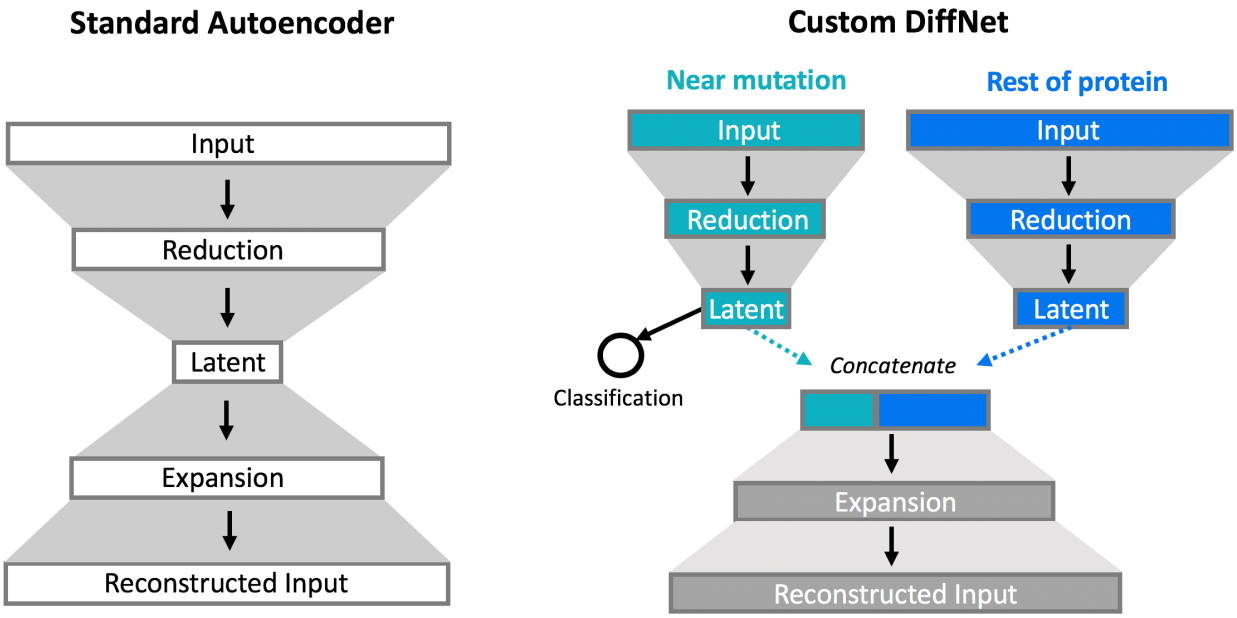

Figure 1. Standard autoencoder architecture and the DiffNet architecture used throughout the manuscript. Autoencoders have an encoder that compress the input data to a bottleneck, or latent, layer and a decoder that expands the latent representation to reconstruct the original input. The DiffNet used in this manuscript splits the input into two encoders where atoms near the mutation are reduced to a latent layer that performs a classification task, while other atoms are reduced to a latent layer without a classification task. The two latent layers are concatenated and trained to reconstruct the original input.

To demonstrate the power of DiffNets, we apply it to a set of variants that modulate the stability of the enzyme TEM β-lactamase. TEM is a bacterial enzyme that confers antibiotic resistance by metabolizing β-lactam drugs like penicillin[24]. Bacteria are quick to evolve new variants of TEM that have activity against new drugs, but these mutations are often destabilizing, so compensatory mutations evolve to restore stability[25–27]. M182T is a clinically important stabilizing mutation in TEM that counterbalances destabilizing mutations that otherwise increase activity against antibiotics[28,29]. Until recently, the mechanism of stabilization by M182T was not known, since crystal structures of M182T and wild-type are nearly indistinguishable[5]. Further, even with simulation data of both variants, it is difficult to identify distinguishing structural preferences since the variants have highly overlapping conformational ensembles (Fig. 2). Recently, our group used MD simulations, NMR experiments, and X-ray crystallography to show that M182T and another stabilizing variant, M182S, increase stability relative to non-stabilizing variants (wild-type, M182N, and M182V) by stabilizing helix 9 and the domain interface between helix 9 and the adjacent β-sheet. Analysis of MD simulation data showed that stabilizing variants have an increased propensity to adopt a more compact helix 9 relative to non-stabilizing variants (Fig. 2). Additionally, NMR data showed that stabilizing variants have chemical shifts in the β-sheet adjacent to helix 9, which led to speculation that helix 9 has an increased propensity to be packed

more tightly to the adjacent β-sheet, though analysis of MD simulations failed to confirm that speculation. We evaluate if DiffNets can pinpoint these underlying features that explain stability differences between TEM variants. This is a difficult task for existing methods since helix 9 compaction is subtle (< 1 Ångstrom compaction) compared to loop motions elsewhere in the protein and required many months of analysis to identify in our past work[5].

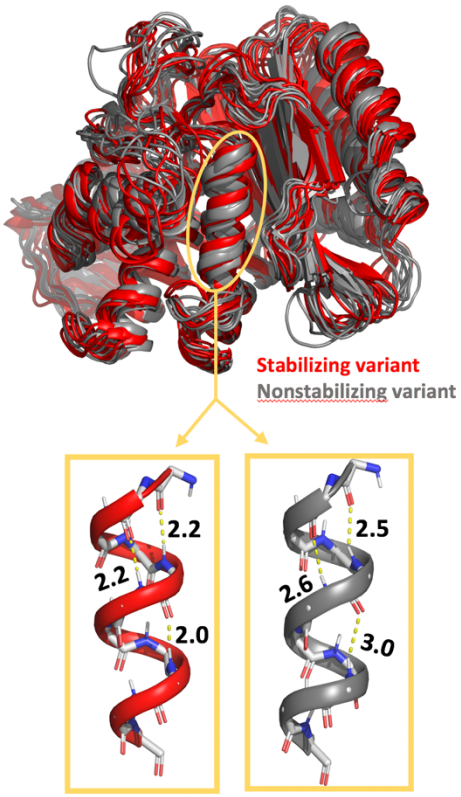

Figure 2. Structural configurations sampled from MD simulations of a wild-type TEM (grey – nonstabilizing variant) and M182T (red – stabilizing variant). Helix 9 is circled in yellow and shown below in a compact configuration (left, red) and a noncompact configuration (right, grey). Hydrogen bond distances are shown in Ångstroms.

Results and Discussion

The DiffNet architecture is inspired by supervised autoencoders.

The DiffNet architecture is based on an autoencoder, which is a deep learning framework commonly used for dimensionality reduction. Standard autoencoders have been used in many applications to MD simulation data.[4,30–41] They connect two separate neural networks, an encoder network and a decoder network, to perform an unsupervised dimensionality reduction on input data (e.g. a protein structure from a frame of an MD simulation). First, the encoder network transforms the input with a series of matrix multiplications and non-linear activation functions to progressively reduce the dimensionality of the input to a bottleneck layer, called the latent space vector. Then, the latent space vector is used as input to the decoder network, whose architecture is often a mirror image of the encoder network, that attempts to reconstruct the original input. If the autoencoder can compress and then reconstruct the original input with high accuracy, this implies that the low-dimensional latent space vector retains the salient features that describe the

input. Autoencoders are initialized with random matrix multiplications and the matrix values (weights) are tuned by training across many examples. Concretely, the weights in an unsupervised autoencoder are trained to minimize a loss function that measures the difference between the input and output of the model, called the input reconstruction error.

Supervised autoencoders[22] augment autoencoders with a loss function that measures how accurately the latent space vector performs a user-defined classification task (e.g. did the protein structure come from a wild-type or variant simulation?). Therefore, supervised autoencoders must learn weights that simultaneously minimize protein reconstruction error and classification error. Previously, supervised autoencoders have been used as a way to obtain better performance on a classification task[22]. In contrast, we use the classification task to learn a more interpretable low-dimensional projection of data.

DiffNets are multi-task neural networks that leverage supervised autoencoders to learn a low-dimensional representation of MD simulation data that organizes protein structural configurations based on the degree to which they are associated with a biochemical property. While DiffNets encompass any number of architectures that use a supervised autoencoder, this section describes the specific architecture that we use throughout the manuscript (Fig. 1). The input to this architecture is protein XYZ coordinates (C,CA,N,CB) from a simulation frame, which are whitened for normalization (see methods). The input gets split into two encoder networks, based on whether or not an atom is within 1nm of the mutated residue under consideration. Each encoder network has a single hidden layer that reduces the number of features by 4-fold, which is then connected to a bottleneck layer with a smaller number of variables. Only the encoder with inputs from near the mutated residue performs a classification task, which has a sigmoid activation function. This split architecture guides DiffNets to search in the region near the mutation to find differences between variants. In principle, the classification task can be performed on any portion of the protein, or the whole protein, but performing the classification based on the region near the mutation is a reasonable default since it is the root of any differences between variants. Moreover, classifying based on the region of the mutation does not preclude the identification of relevant distal structural differences between variants. If the mutation causes biochemically relevant differences at distal regions then these regions are inherently linked to the state of the region of the mutation and, thus, are implicitly linked to the classification task. After encoding and classification, latent vectors from both encoders are concatenated and passed through a single decoder network that reconstructs the input protein structure. All hidden layers employ rectified linear unit activation functions (ReLU).

Autoencoders and DiffNets successfully reconstruct protein structures.

To gain insight about structural features that distinguish protein variants, DiffNets, or autoencoders in general, must learn a low-dimensional representation that preserves important structural degrees of freedom. One way to test if the dimensionality reduction preserves important structural degrees of freedom is to reconstruct the protein structure from the latent representation and measure the reconstruction error. A low reconstruction error indicates that the latent space retains important structural information about the protein. Since autoencoders can be tuned to have any number of latent variables, it is possible to determine the relationship between number of latent variables and the extent that structural degrees of freedom are preserved. We sought to ascertain how many latent variables are needed to accurately reconstruct protein structures from protein simulations of TEM β-lactamase. Additionally, given that DiffNets commit representational

power to a classification task, we wanted to determine if the multi-task nature of DiffNets results in degraded ability to reconstruct protein structures relative to a standard autoencoder.

To evaluate how latent space size affects protein reconstruction error for autoencoders and DiffNets, we trained them on TEM simulation data using a variable number of latent variables. For this experiment, we trained DiffNets in a supervised setting that we designed explicitly to understand how the classification task alters the dimensionality reduction. Specifically, we curated a dataset of ~18,000 simulation frames from wild-type and M182T simulations where half of the frames have a compact helix 9 (helix compaction criteria described in Methods). The DiffNets are trained to perform a dimensionality reduction and classify whether or not a given frame has a compact helix. The classification labels are not updated with expectation maximization in this case. For comparison, we trained an unsupervised autoencoder with the same exact architecture and data, only without the classification task. To determine how protein reconstruction error changes as a function of the number of latent space variables, we trained with a latent space of 3, 5, 10, 25, or 50 latent variables. We calculated protein reconstruction error as the root-mean-square deviation (RMSD) of the reconstructed structure from the DiffNet, or autoencoder, compared to the original simulation frame.

We find that standard autoencoders and DiffNets reconstruct protein structures from MD simulation data with similar performance. Autoencoders and DiffNets can reduce a protein structural configuration down to 3 variables and then reconstruct the original protein conformation with ~1 Ångstrom error (Fig. 3a,b). We are only aware of one other study that has reported RMSD error for an autoencoder applied to MD simulation data. They calculate reconstruction error for 8 proteins with varying degrees of freedom, fewer and greater than TEM, and report RMSDs ranging from 0.89-1.67 Ångstroms when using 30 latent variables[32]. Generally, adding more dimensions can decrease interpretability so it is encouraging that DiffNets produce a high-fidelity dimensionality reduction with a small number of latent variables. Moreover, the small difference in reconstruction error between DiffNets and unsupervised autoencoders highlights that dedicating representational power to a feature classification task results in a negligible degradation of the geometric dimensionality reduction.

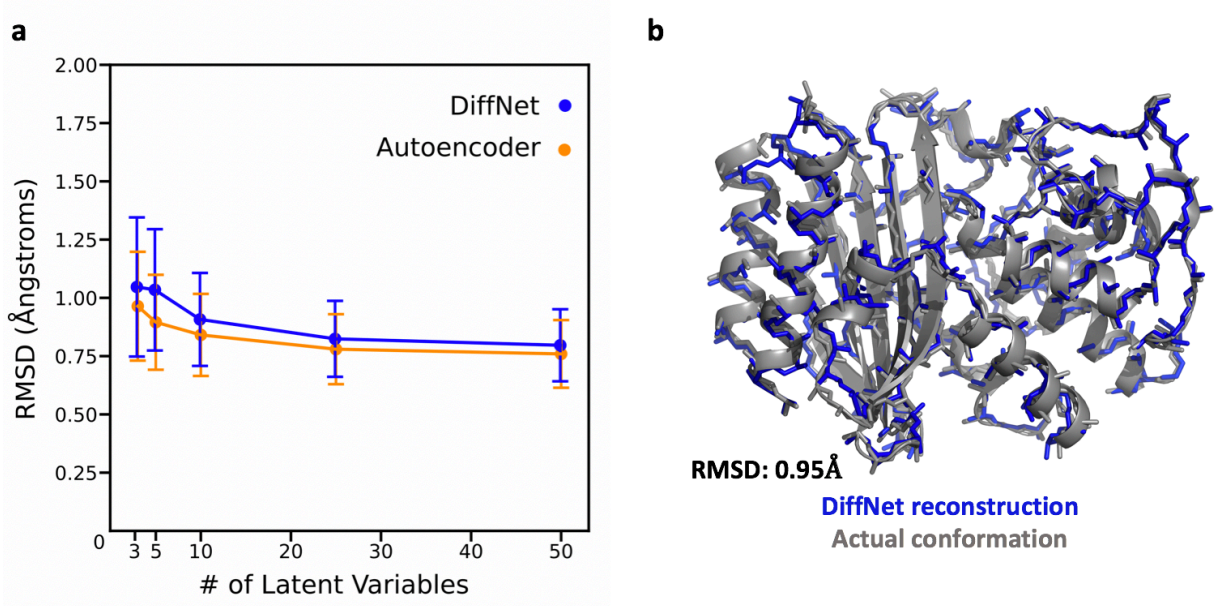

Figure 3. Ability of autoencoders and DiffNets to reconstruct protein structures. **a** Reconstruction error plots showing the RMSD between a protein structure from simulation and the corresponding protein structure generated by an unsupervised autoencoder (yellow) or DiffNets (blue). **b** structure representing the difference between a structure generated by the DiffNet (blue) vs. the actual conformation from simulation (grey) when training with 3-5 latent variables.

DiffNets classification task organizes the latent space to emphasize structural features important to biochemical properties.

Dimensionality reduction algorithms are only helpful for identifying differences between two classes of data if the two classes of data are separated in the latent space. We propose that unsupervised autoencoders learn a latent representation of data that focuses on large geometric variations, so structures with large geometric differences are separated, while structures with subtle differences are close together. As a result, if biochemical differences between protein variants are caused by subtle geometric changes, then the variants will be highly overlapping in the latent space and thus, the autoencoder fails to provide a useful way to distinguish variants. We hypothesized that augmenting a standard autoencoder with a classification task, as with DiffNets, would reorganize the latent space to highlight relevant differences between datasets, even if they are subtle.

To evaluate if the DiffNet classification layer alters the latent space in a way that helps identify differences between two classes of data, we compared the latent space of DiffNets to the latent space of unsupervised autoencoders after training on a dataset that includes two classes of data distinguishable by a subtle, but important difference. Helix 9 compaction is a subtle structural change, but an important indicator of the relative stability of TEM protein variants. As in the previous section, we trained DiffNets and autoencoders on a dataset of ~18,000 simulation frames from wild-type and M182T simulations where half of the frames have a compact helix 9. The DiffNets and autoencoders trained were identical, except the DiffNet latent space has to classify helix 9 as compact or non-compact. This dataset was selected specifically to evaluate how the classification task of the DiffNet alters the dimensionality reduction compared to a standard autoencoder. In a normal setting we would not have *a priori* knowledge about the importance of helix 9 compaction. However, this is an important test to determine if adding a classification task can reorganize the latent space to highlight differences between datasets, which is a property that DiffNets will ultimately need to identify differences between variants.

Requiring DiffNets to perform a classification task in tandem with dimensionality reduction successfully reconfigures the latent space to disentangle compact helix configurations from non-compact helix configurations. In the unsupervised autoencoder, simulation frames of compact and non-compact helices are overlapping in the latent space (Fig. 4a). This demonstrates that training an unsupervised autoencoder on two classes of data does not necessarily yield a latent representation that provides any insight on how the two classes of data are different. Further, we fixed the latent space and then trained a classification layer (i.e. performed logistic regression) to classify frames as having a compact vs. noncompact helix 9. The resulting Receiver Operating Characteristic (ROC) curve, which measures classification performance, shows a classification performance similar to random guessing (AUC=0.54) providing further evidence that the latent representation does not help distinguish the two classes of data. In contrast, when autoencoders are trained with a classification task (as with DiffNets), there is clear separation between the classes of data (Fig. 4a). The latent space is organized explicitly to highlight the differences

between the two classes of data and has excellent performance at classifying compact vs. non-compact helix states (AUC=0.91) (Fig. 4b). From this result, we reasoned that DiffNets should be able to learn a latent space that highlights biochemically-relevant differences between protein variants if given a classification task that labels structural configurations with an indication of their biochemical properties.

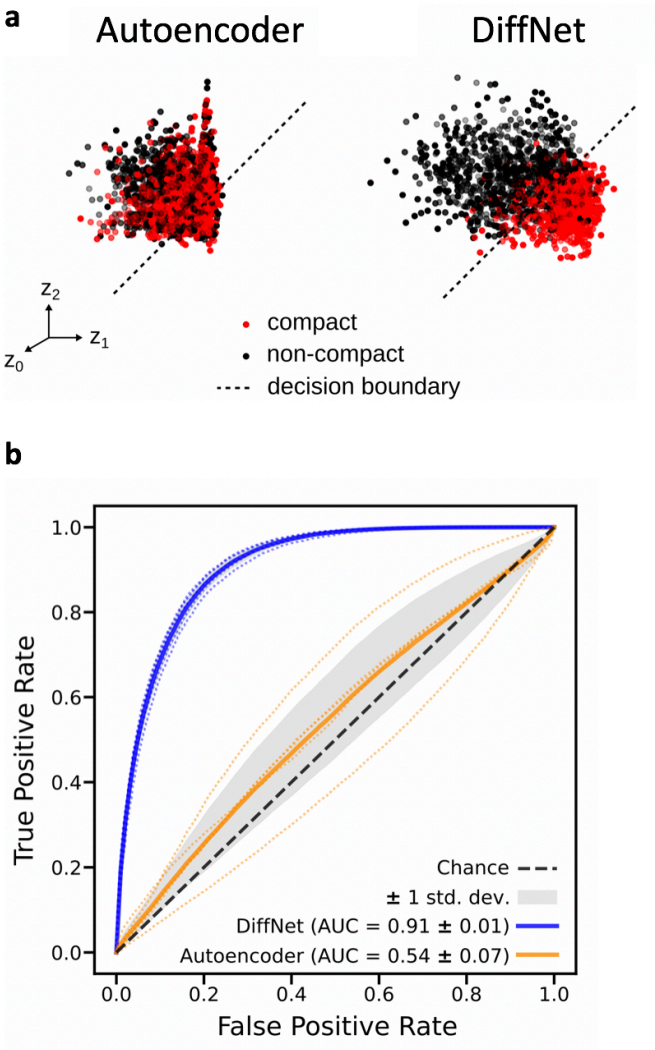

Figure 4. Evaluation of how the classification layer of a supervised autoencoder alters the learned latent representation. **a** Simulation frames that have a compact helix (red) and a noncompact helix (black) are projected onto the three-dimensional latent space learned by an unsupervised autoencoder (left) and a DiffNet (right). The decision boundary (black dotted line) indicates the plane that each neural network uses to separate compact helix states from non-compact helix states. **b** ROC curve showing the average classification performance of the unsupervised autoencoder (dark yellow) and the DiffNet (dark blue) as well as the performance for each of the 5 folds of cross validation (faded dotted lines).

Self-supervised DiffNets learn the important structural preferences that govern stability in TEM variants.

We propose that DiffNets trained to classify structural configurations based on their associated biochemical property would result in a latent space that separates structural configurations based on their biochemical properties. This would make it easy to identify structural features that explain biochemical differences between protein variants. However, providing a label indicating the biochemical property of each structural configuration is a difficult challenge since we typically only know the biochemical property associated with each protein variant, rather than each structural configuration. Requiring DiffNets to classify simulation frames from different protein variants as different classes could be a reasonable starting point but comes with challenges. First, many labels are likely to be incorrect. For example, while M182T is a stabilizing variant, M182T simulations are likely to sample some conformations that are not stabilizing, and vice-versa for wild-type. Additionally, it is unlikely that binary labels (e.g. stabilizing vs. not stabilizing) are optimal. Structural configurations are often on a spectrum where a specific configuration may have some features associated with a biochemical property, but not all, so labels are not easily discretized. Without addressing these challenges, DiffNets would learn a latent space that places structural configurations on a spectrum that indicates the degree that a single configuration is "wild-type-like" or "variant-like", but does not explicitly separate configurations based on the biochemical property they confer, which is our ultimate goal. As a result, DiffNets would separate structural configurations based on any structural differences that distinguish protein variants, even if they are unrelated to their biochemical differences. This classic supervised approach might have utility for developing therapeutics, but would be bolstered by a mechanism that addresses the above challenges to shift the classification labels from indicating which variant simulation a structure came from to labels that indicate the biochemical property that a structure is associated with.

We devised an expectation maximization scheme that iteratively updates the classification labels of simulation frames to indicate the degree to which each structural configuration is associated with a particular biochemical property. Expectation maximization is a statistical method that allows the parameters of a model to be fit, even when the outputs of the model cannot directly be observed in the training data[42] (i.e. when they are hidden). In our case, the hidden variables are the elements of a vector of numbers (y) that label each individual frame, or conformation, of a molecular dynamics simulation with a classification label indicating the degree that a conformation is associated with a biochemical property, or not. For example, if stability is the biochemical property that distinguishes protein variants, each frame, (y_i), should have a label between 0 and 1 that ranges from nonstabilizing to stabilizing. However, there is no *a priori* knowledge detailing if a given conformation increases stability, thus, these variables are hidden. To estimate these values, the expectation maximization algorithm iteratively alternates between an expectation step and a maximization step. We adapt an expectation maximization scheme from Zaretski et. al.[43] to improve DiffNets training labels. During the maximization step, DiffNets are trained to estimate a value for each state indicating the likelihood that a state is stabilizing, which is initially learned using noisy training labels (e.g. all 1s for stabilizing variants and all 0s for non-stabilizing variants). Then, the expectation step refines the training labels by computing the expected values of the hidden variables, y, using the output from the DiffNet, $\hat{y}$, conditioned on constraints about how many frames from each type of simulation we expect to be in stable conformations. The expectation is the probability-weighted average of all binary realizations of binomial distributions parameterized by $\hat{y}$ that assign the right number of stable conformations to a given batch of conformations.

$$y_i = E[\hat{y}_i \mid S_L \leq \hat{y}_b \leq S_U]$$

$$= P(\hat{y}_i \text{ is } 1) * \left(\frac{P(S_L - 1 \leq \hat{y}_r - \hat{y}_i \leq S_U - 1)}{P(S_L \leq \hat{y}_r \leq S_U)}\right)$$

where y_i is the true label of each individual frame, $\hat{y}_i$ is the DiffNet output, S_L and S_U are the lower and upper bounds on how many conformations in a batch are stabilizing based on user-defined constraints, $\hat{y}_b$ is the sum of the binary outcomes of a batch which contains conformation i, $P(S_L - 1 \leq \hat{y}_r - \hat{y}_i \leq S_U - 1)$ is the probability that the number of conformations in a batch is within the limits if conformation i is ignored, and $P(S_L \leq \hat{y}_r \leq S_U)$ is the probability that the number of conformations in a batch is within the limits, including conformation i. Ultimately, the desired outcome is that the expectation maximization algorithm redistributes training labels from all 0s and 1s for simulation frames of non-stabilizing and stabilizing variants, respectively, to values between 0 and 1 that reflect the degree that a structural configuration is stabilizing. This mechanism is self-supervised since the training labels are learned by the algorithm, rather than explicitly curated.

We trained self-supervised DiffNets to identify structural preferences that distinguish stabilizing and non-stabilizing TEM variants using data from four β-lactamase variants. In this case, the DiffNets receive no *a priori* information about features, like helix 9 compaction, that are responsible for biochemical differences between variants. If self-supervision of DiffNets works as expected, then training should produce a latent space where it is easy to identify the structural features that confer stability to M182T and M182S, relative to WT and M182V. Specifically, we expect to see structural configurations with a compact helix 9 in one region of latent space and scanning across to the other side of latent space should correlate with structural configurations becoming less compact. Additionally, based on previous NMR data, we expect to see changes at the interface between helix 9 and the adjacent β-sheet. To evaluate if DiffNets learn these functionally-relevant structural differences between variants, we trained DiffNets on 6.5μs of simulation data for each variant: M182T, M182S, WT, and M182V. All frames from M182T and M182S (stabilizing variants) were initially assigned classification labels of 1, and simulation frames from M182V and wild-type were initially assigned 0s. During the EM procedure, we calculate the expected values (updated labels) conditioned on the constraint that 0-30% of non-stabilizing variants frame are likely to be stabilizing, and 60-90% of frames for stabilizing variants.

Using expectation maximization, DiffNets learn a low-dimensional representation that accurately highlights helix 9 compaction as an important difference between protein variants. If self-supervision helps DiffNets learn structural differences between variants that explain their biochemical differences, we expect to see a gradient in the latent space that correlates with changes in structural configurations from non-stabilizing to stabilizing. We can use the DiffNets classification score as a proxy for the latent space and measure how it changes in response to helix 9 compaction. Encouragingly, the DiffNets classification score correlates well with helix compaction. When we bin DiffNet classification scores from 0 to 1 on all simulation frames, we find the DiffNet classification score increases as the distance between three important residue pairs involved in helix 9 compaction decreases (Fig. 5a). This shows that the DiffNet accurately learned that helix 9 compaction is a structural preference that confers stability to M182T and M182S even though it received no *a priori* knowledge about the importance of this helix. While capturing helix compaction is necessary to explain the mechanism of increased stability, it is not sufficient, since packing against nearby β-sheets is also important. Therefore, if the DiffNet

classification score is comprehensively capturing stability it should not be a perfect proxy for helix compaction. To evaluate if the DiffNet classification score is overfit to helix compaction, we measured how well the DiffNet classification score classifies simulation frames that have compact vs. non-compact helices. Appropriately, the DiffNet classification score classifies compact vs non-compact helix with reasonable, but not perfect, accuracy (AUC=0.73) (Fig. 5B). We are encouraged by this AUC score because it indicates that the DiffNets classification score is strongly influenced by helix compaction, but must also be a function of other structural features.

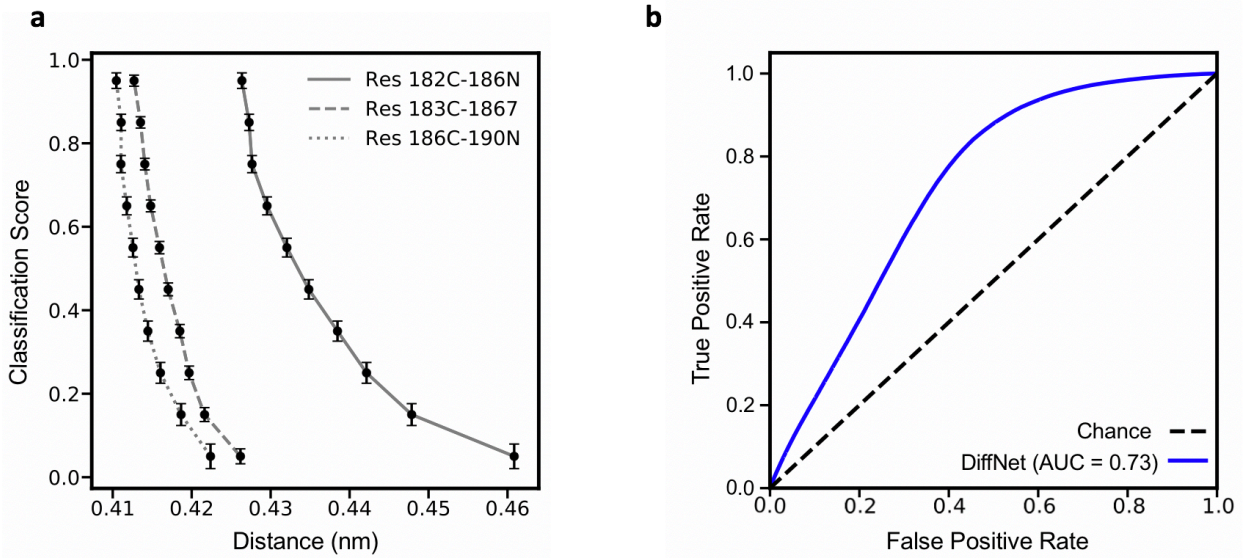

Figure 5. Ability of self-supervised DiffNets to identify the importance of helix 9 compaction. **a** DiffNet classification score as a function of the distance of 3 hydrogen bond residues in helix binned using all simulation data from WT, M182V, M182T, and M182S. **b** ROC curve showing the performance of DiffNets when classifying compact vs. noncompact helix states, which DiffNets were not explicitly trained on.

While many deep learning approaches are criticized for their lack of interpretability, the DiffNet architecture provides a straightforward means to understand what the network learned, which makes it easy to gain biophysical insight. To automate DiffNet interpretation, we measured all inter-atom distances within 1nm of the mutation for 2000 cluster centers calculated from all simulations and then measured the linear correlation between each distance and the DiffNet classification score. We plot the top 1% of distances correlated with the DiffNet classification score to visualize the conformational changes that the DiffNet views as important for distinguishing stabilizing and non-stabilizing variants. Encouragingly, the distance correlations strongly point to helix 9 compaction as an important feature of stability (Fig. 6). While the helix compaction is striking, there are other important distance changes that match well with previous NMR data[5]. The previous study proposed that the interface between helix 9 and the adjacent β-sheet become more tightly packed, but months of analysis of simulation data failed to confirm this hypothesis. On the same simulation dataset, DiffNets clearly learn that this interface becomes more tightly packed for stabilizing variants (Fig. 6). Moreover, the distance correlations line up with NMR data perfectly in some instances. For example, the distance correlations show a compaction between residue 189, in the middle of the helix, and residue 263 on the β-sheet, which both have chemical shift perturbations for stabilizing variants relative to the nonstabilizing wild-type variants. Often times the important order parameter that distinguishes protein variants can be complicated and, therefore, easily missed even with months of analysis. DiffNets can learn complicated order

parameters and help automate the process of identifying structural features that explain biochemical differences between protein variants.

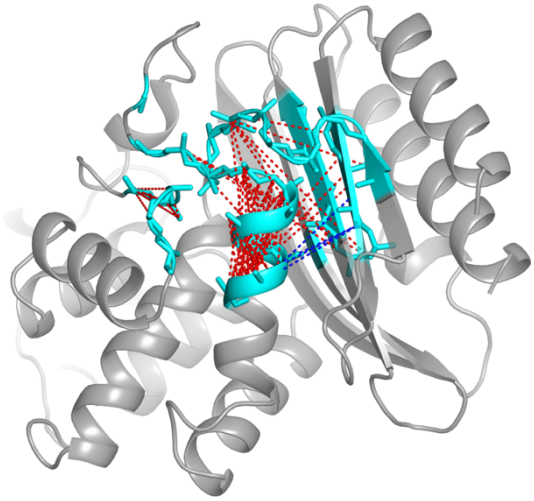

Figure 6. Visualization of the features that DiffNets find important for increased stability of M182T and M182S variants. Protein atoms are colored cyan if they are near the mutation, which indicates that they were included in the classification task and considered for the distance correlation calculation. Dotted lines indicate distances between two atoms that change in a way that is strongly correlated with an increased DiffNet classification score. Red indicates the atoms move closer together as the classification score increases, blue indicates atoms moving away from each other.

DiffNets predictive power extends to variants not seen during training.

We challenged DiffNets by testing if they correctly classify M182N, a variant outside of the training set that does not confer stability to β-lactamase. The M182N mutation is not stabilizing, but does have a helix-capping residue, like the stabilizing variants, whereas the non-stabilizing variants do not. Ultimately, M182N is not stabilizing because of competition between capping of helix 9 and other interactions. Accurately classifying M182N with the non-stabilizing variants is a particularly difficult challenge for DiffNets since M182N may have helix dynamics that are more similar to the stabilizing variants. We used 6.5μs of simulation data from the M182N variant as input for the self-supervised DiffNet described above.

DiffNets accurately groups M182N with the non-stabilizing variants. The DiffNet classification score across variants that were in the training is as expected; stabilizing variants are grouped together at high values, and non-stabilizing variants group together at lower values (Fig. 7). Interestingly, there is some overlap between stabilizing and non-stabilizing variants, which suggests that self-supervision helped DiffNets learn that some structural configurations from stabilizing variants can be non-stabilizing, and vice-versa. Importantly, the untrained variant, M182N, has a major peak with the non-stabilizing variants and a minor peak with the stabilizing variants (Fig. 7). We find it interesting, but not surprising, that M182N has a minor peak with the stabilizing variants even though M182N is not a stabilizing mutation. We reason that DiffNets can recognize that the helix capping residue is important for stability, but still recognizes other nuances of stability that help it accurately group M182N with the non-stabilizing variants. Altogether, these results suggest that self-supervised DiffNets learn how to separate protein

variants based on biochemical differences and will be a useful tool to identify the structural basis of biochemical properties. In future scenarios, it is possible that MD simulations of novel variants could be combined with DiffNets to predict their biochemical properties. It is likely that this will only work if the conformational space of the novel variant has high overlap with variants that were used to train DiffNets.

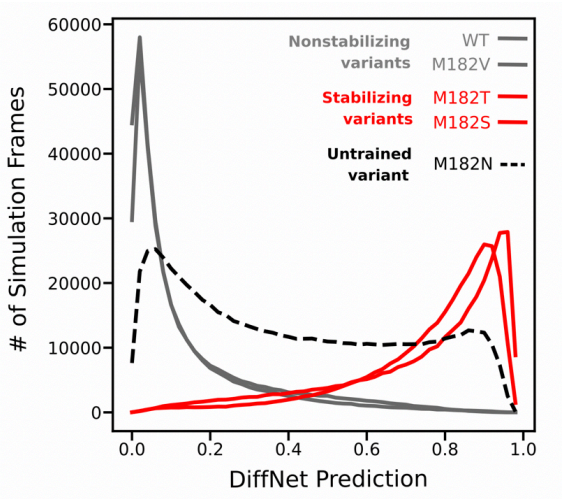

Figure 7. DiffNet groups M182N with nonstabilizing variants. Histogram shows the DiffNet classification score across all simulation frames from M182Tand M182S (red - stabilizing variants), WT and M182V (grey – nonstabilizing variants), and M182N (dotted black – untrained variant).

Discussion

We have introduced DiffNets, a deep learning framework to identify the mechanistic basis for biochemical differences between protein variants. MD simulations provide information that can be used to uncover structural mechanisms that explain how an amino acid substitution causes a protein to lose its normal function to become pathogenic or gain a new function to adapt to new environments. However, comparing MD simulations of protein variants is difficult since it typically requires several steps including data featurization, dimensionality reduction, clustering, building conformational maps of variants, and then searching in a high-dimensional space to identify structural trends that differentiate variants. Previous work, especially VAMPnets[44], has demonstrated that deep learning algorithms can be used to automate many parts of MD simulation analysis for understanding the important states and dynamics of a single protein. In contrast, DiffNets use deep learning to simplify the process of identifying important structural differences *between* systems. The key to performing this task is to perform a dimensionality reduction simultaneously with a classification task to learn a latent representation that organizes protein structural configurations based on their biochemical properties. In this work we applied DiffNets to identify structural differences that explain stability differences in TEM protein variants, but we imagine DiffNets will be used to identify biochemically relevant structural differences between systems with any type of perturbation. For example, DiffNets applied to a protein receptor in the apo- and holo- states would identify differences in these systems that could provide information on how signaling proteins are activated.

Methods

MD Simulations

All MD simulation data were generated for a previous manuscript by Zimmerman et. al[5]. Briefly, all simulations were run with Gromacs 5.1.1[45]. Simulations were initialized from the TEM-1 β-lactamase crystallographic structure (PDB ID: 1JWP)[26] and ran at 300K using the AMBER03 force field with explicit TIP3P solvent[46,47]. Each variant, wild-type, M182V, M182T, M182S, and M182N was simulated for 6.5 µs including 4 µs of FAST-RMSD adaptive sampling[48] and 2.5 µs of conventional sampling. Conformations were stored every 20 ps.

Data Processing

Simulation data was preprocessed before becoming input to the DiffNets. Simulation trajectories and the original crystallographic structure (PDB ID: 1JWP) are stripped down to the XYZ coordinates of the protein backbone without carbonyl oxygens (C, CA, CB, and N). Then, the trajectories are centered at the origin and aligned to the crystallographic structure. Next, we follow a procedure similar to Wehmeyer and Noe[36] to mean-shift the XYZ coordinates to zero, followed by whitening. First, we mean shift,

$$x^{mean-free} = \sum_{i=1}^{N_t} x_i - \bar{x}$$

where $x^{mean-free}$ is the mean-shifted trajectory of XYZ coordinates, x_i is a single frame with XYZ coordinates, $\bar{x}$ is the mean of the XYZ coordinates across all trajectories, and N_t is the number of frames in all trajectories.

Next, we whiten the data,

$$\tilde{x} = C_{00}^{-\frac{1}{2}} x^{mean-free}$$

where $\tilde{x}$ is the whitened trajectory of XYZ coordinates and C_{00} is the covariance matrix for the XYZ coordinates. Whitening decorrelates the inputs and adjusts their variance to be unity. After whitening, we use one out of every ten simulation frames for each epoch of DiffNet training. In practice, whitening and unwhitening of the data is performed on the input XYZ coordinates directly in the DiffNet with frozen (untrainable) weights.

Neural network training

We trained DiffNets with three loss functions to minimize protein reconstruction error (ℓ_{Recon}), minimize feature classification error (ℓ_{Class}), and minimize the correlation of latent space variables (ℓ_{Corr}).

$$\mathcal{L}_{DiffNet} = \ell_{Recon} + \ell_{Class} + \ell_{Corr}$$

The reconstruction loss term attempts to tune the network weights to properly reconstruct the original XYZ coordinates of the protein. This loss combines an absolute error (L1), which

funnels reconstructions to the proper XYZ coordinates, and a mean-squared error (L2) to strongly discourage outliers. Explicitly,

$$\ell_{Recon} = \frac{1}{N_b}\sum_{i=1}^{N_b}\frac{1}{N_n}\sum_{j=1}^{N_n}\left[\left|x_{ij} - \hat{x}_{ij}\right| + \left(x_{ij} - \hat{x}_{ij}\right)^2\right]$$

where N_n is the number of output nodes (all XYZ coordinates), N_b is the number of examples in a training batch, x_{ij} is a target value (actual XYZ coordinate), and $\hat{x}_{ij}$ is the output value from the DiffNet.

The classification error is a binary cross entropy error that penalizes misclassifications by the latent space. This classification loss attempts to constrain the latent space to learn a dimensionality reduction that can also classify a biophysical feature. Explicitly,

$$\ell_{Class} = \frac{1}{N_b}\sum_{i=1}^{N_b} y_i * \log(\hat{y}_i) + (1 - y_i) * \log\left(1 - \hat{y}_i\right)$$

where N_b is the number of examples in a training batch, y_i is the target value, a binary value indicating if a simulation frame has a specific feature or not, and $\hat{y}_i$ is the output of the classification layer by the DiffNet. For the section of the paper that demonstrates how the DiffNets classification layer alters the latent space, we provided classification labels as “compact helix” or “non-compact helix”. This designation was based on a previous manuscript that identified three key hydrogen bond distances in Helix 9 that distinguish stabilizing variants from nonstabilizing variants (Res 182-186, Res 183-187, and Res 186-190)[5].

Finally, we include a loss function to minimize the covariance between latent space variables. This loss takes the form of

$$\ell_{Corr} = \sum_{i \neq j} Cov\left(z_i, z_j\right)^2$$

where $Cov(z_i, z_j)$ is the covariance matrix of the latent vector, Z, across all N_b samples in a training batch. We reason that preventing redundancy in latent variables should maximize the amount of information one can gain in a small number of variables. Ideally, this sets us up to use just a few latent variables and still have a rich amount of information. With fewer latent variables, models are generally more interpretable.

Our training procedure uses several training iterations to progressively build in hidden layers of the DiffNet. First, we train a minimal version of a DiffNet. Explicitly, the encoders have an input layer and a reduction layer with a four-fold reduction in variables. There is no further reduction to a bottleneck layer. Instead, the decoder takes the reduction layer as input and passes it to an output layer. Training this simplified autoencoder is an easier task than training a full DiffNet because the dimensionality reduction it performs is modest. It has ~an order of magnitude more dimensions to explain the original data compared with a true bottleneck layer. We reason that this can generate useful priors for what the reduction layer should capture. In our second pretraining procedure, we freeze those priors and add the bottleneck layer in to train the full

DiffNet. Therefore, this second pretraining step concentrates its representational power on tuning how to properly reduce from the reduction layer to the bottleneck layer. Finally, we unfreeze the priors and train the full DiffNet to polish all weights. Each of these three procedures undergoes 20 training epochs. In the self-supervised setting, classification labels are updated using expectation maximization after each training epoch.

All training was performed in PyTorch 1.1[49] on an NVIDIA Tesla P100. Training on ~120,000 simulation frames of TEM takes under 30 minutes. Training with expectation maximization approximately doubled the training time for DiffNets trained on TEM. We used the Adam optimizer with a learning rate of 0.0001 and a batch size of 32. We did not perform any hyperparameter search.

Code Availability

Data normalization and DiffNet training with, or without, expectation maximization is freely available on GitHub at https://github.com/bowman-lab/diffnets.

Acknowledgements

This work was funded by NSF CAREER Award MCB-1552471 and NIH grant R01 GM124007 (GRB). GRB holds a Career Award at the Scientific Interface from the Burroughs Wellcome Fund and a Packard Fellowship for Science and Engineering from The David & Lucile Packard Foundation. MDW was supported by a MolSSI COVID-19 seed software fellowship.

References

1. Erickson RP. Somatic gene mutation and human disease other than cancer: An update. *Mutat Res - Rev Mutat Res*. 2010. doi:10.1016/j.mrrev.2010.04.002
2. Krawczak M, Ball E V., Fenton I, et al. Human Gene Mutation Database - A biomedical information and research resource. *Hum Mutat*. 2000. doi:10.1002/(SICI)1098-1004(200001)15:1<45::AID-HUMU10>3.0.CO;2-T
3. Davies J. Origins and evolution of antibiotic resistance. *Microbiologia*. 1996. doi:10.1128/mmbr.00016-10
4. Sultan MM, Wayment-Steele HK, Pande VS. Transferable Neural Networks for Enhanced Sampling of Protein Dynamics. *J Chem Theory Comput*. 2018. doi:10.1021/acs.jctc.8b00025
5. Zimmerman MI, Hart KM, Sibbald CA, et al. Prediction of New Stabilizing Mutations Based on Mechanistic Insights from Markov State Models. *ACS Cent Sci*. 2017. doi:10.1021/acscentsci.7b00465
6. Perryman AL, Lin J-H, McCammon JA. HIV-1 protease molecular dynamics of a wild-type and of the V82F/I84V mutant: Possible contributions to drug resistance and a potential new target site for drugs. *Protein Sci*. 2004. doi:10.1110/ps.03468904
7. Schwantes CR, Shukla D, Pande VS. Markov state models and tICA reveal a nonnative folding nucleus in simulations of NuG2. *Biophys J*. 2016. doi:10.1016/j.bpj.2016.03.026
8. Sang D, Pinglay S, Wiewiora RP, et al. Ancestral reconstruction reveals mechanisms of erk regulatory evolution. *Elife*. 2019. doi:10.7554/eLife.38805
9. Razavi AM, Voelz VA. Kinetic Network Models of Tryptophan Mutations in β-Hairpins Reveal the Importance of Non-Native Interaction. *J Chem Theory Comput*. 2015.

doi:10.1021/acs.jctc.5b00088
10. Romero PA, Arnold FH. Exploring protein fitness landscapes by directed evolution. *Nat Rev Mol Cell Biol*. 2009. doi:10.1038/nrm2805
11. James LC, Tawfik DS. Conformational diversity and protein evolution - A 60-year-old hypothesis revisited. *Trends Biochem Sci*. 2003. doi:10.1016/S0968-0004(03)00135-X
12. Hart KM, Ho CMW, Dutta S, Gross ML, Bowman GR. Modelling proteins' hidden conformations to predict antibiotic resistance. *Nat Commun*. 2016. doi:10.1038/ncomms12965
13. Knoverek CR, Amarasinghe GK, Bowman GR. Advanced Methods for Accessing Protein Shape-Shifting Present New Therapeutic Opportunities. *Trends Biochem Sci*. 2019. doi:10.1016/j.tibs.2018.11.007
14. Karplus M, McCammon JA. Molecular dynamics simulations of biomolecules. *Nat Struct Biol*. 2002. doi:10.1038/nsb0902-646
15. Bowman GR, Pande VS, Noé F. An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation. *Springer*. 2014. doi:10.1007/978-94-007-7606-7
16. Husic BE, McKiernan KA, Wayment-Steele HK, Sultan MM, Pande VS. A Minimum Variance Clustering Approach Produces Robust and Interpretable Coarse-Grained Models. *J Chem Theory Comput*. 2018. doi:10.1021/acs.jctc.7b01004
17. David CC, Jacobs DJ. Principal component analysis: A method for determining the essential dynamics of proteins. *Methods Mol Biol*. 2014. doi:10.1007/978-1-62703-658-0_11
18. Teodoro ML, Phillips GN, Kavraki LE. Understanding Protein Flexibility through Dimensionality Reduction. In: *Journal of Computational Biology*. ; 2003. doi:10.1089/10665270360688228
19. Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science (80-)*. 2006. doi:10.1126/science.1127647
20. Naritomi Y, Fuchigami S. Slow dynamics in protein fluctuations revealed by time-structure based independent component analysis: The case of domain motions. *J Chem Phys*. 2011. doi:10.1063/1.3554380
21. Pérez-Hernández G, Paul F, Giorgino T, De Fabritiis G, Noé F. Identification of slow molecular order parameters for Markov model construction. *J Chem Phys*. 2013. doi:10.1063/1.4811489
22. Le L, Patterson A, White M. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In: *Advances in Neural Information Processing Systems*. ; 2018.
23. Babenko B. Multiple instance learning: algorithms and applications. *View Artic PubMed/NCBI Google Sch*. 2008.
24. Jacquier H, Birgy A, Le Nagard H, et al. Capturing the mutational landscape of the beta-lactamase TEM-1. *Proc Natl Acad Sci U S A*. 2013. doi:10.1073/pnas.1215206110
25. Orencia MC, Yoon JS, Ness JE, Stemmer WPC, Stevens RC. Predicting the emergence of antibiotic resistance by directed evolution and structural analysis. *Nat Struct Biol*. 2001. doi:10.1038/84981
26. Wang X, Minasov G, Shoichet BK. Evolution of an antibiotic resistance enzyme constrained by stability and activity trade-offs. *J Mol Biol*. 2002. doi:10.1016/S0022-2836(02)00400-X
27. Thomas VL, McReynolds AC, Shoichet BK. Structural Bases for Stability-Function Tradeoffs in Antibiotic Resistance. *J Mol Biol*. 2010. doi:10.1016/j.jmb.2009.11.005
28. Woodford N, Ellington MJ. The emergence of antibiotic resistance by mutation. *Clin*

Microbiol Infect. 2007. doi:10.1111/j.1469-0691.2006.01492.x
29. Salverda MLM, de Visser JAGM, Barlow M. Natural evolution of TEM-1 β-lactamase: Experimental reconstruction and clinical relevance. *FEMS Microbiol Rev*. 2010. doi:10.1111/j.1574-6976.2010.00222.x
30. Lemke T, Peter C. EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations. *J Chem Theory Comput*. 2019. doi:10.1021/acs.jctc.8b00975
31. Greener JG, Moffat L, Jones DT. Design of metalloproteins and novel protein folds using variational autoencoders. *Sci Rep*. 2018. doi:10.1038/s41598-018-34533-1
32. Degiacomi MT. Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space. *Structure*. 2019. doi:10.1016/j.str.2019.03.018
33. Noé F, De Fabritiis G, Clementi C. Machine learning for protein folding and dynamics. *Curr Opin Struct Biol*. 2020. doi:10.1016/j.sbi.2019.12.005
34. Hernández CX, Wayment-Steele HK, Sultan MM, Husic BE, Pande VS. Variational encoding of complex dynamics. *Phys Rev E*. 2018. doi:10.1103/PhysRevE.97.062412
35. Tsuchiya Y, Taneishi K, Yonezawa Y. Autoencoder-Based Detection of Dynamic Allostery Triggered by Ligand Binding Based on Molecular Dynamics. *J Chem Inf Model*. 2019. doi:10.1021/acs.jcim.9b00426
36. Wehmeyer C, Noé F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J Chem Phys*. 2018. doi:10.1063/1.5011399
37. Chen W, Ferguson AL. Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration. *J Comput Chem*. 2018. doi:10.1002/jcc.25520
38. Teletin M, Czibula G, Bocicor MI, Albert S, Pandini A. Deep autoencoders for additional insight into protein dynamics. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. ; 2018. doi:10.1007/978-3-030-01421-6_8
39. Wang Y, Ribeiro JML, Tiwary P. Past–future information bottleneck for sampling molecular reaction coordinate simultaneously with thermodynamics and kinetics. *Nat Commun*. 2019. doi:10.1038/s41467-019-11405-4
40. Wang Y, Lamim Ribeiro JM, Tiwary P. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Curr Opin Struct Biol*. 2020. doi:10.1016/j.sbi.2019.12.016
41. Lusch B, Kutz JN, Brunton SL. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat Commun*. 2018. doi:10.1038/s41467-018-07210-0
42. Moon TK. The expectation-maximization algorithm. *IEEE Signal Process Mag*. 1996. doi:10.1109/79.543975
43. Zaretzki JM, Browning MR, Hughes TB, Swamidass SJ. Extending P450 site-of-metabolism models with region-resolution data. *Bioinformatics*. 2015. doi:10.1093/bioinformatics/btv100
44. Mardt A, Pasquali L, Wu H, Noé F. VAMPnets for deep learning of molecular kinetics. *Nat Commun*. 2018. doi:10.1038/s41467-017-02388-1
45. Abraham MJ, Murtola T, Schulz R, et al. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*. 2015. doi:10.1016/j.softx.2015.06.001
46. Duan Y, Wu C, Chowdhury S, et al. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins Based on Condensed-Phase Quantum Mechanical Calculations. *J Comput Chem*. 2003. doi:10.1002/jcc.10349
47. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, Klein ML. Comparison of simple potential functions for simulating liquid water. *J Chem Phys*. 1983.

doi:10.1063/1.445869
48. Zimmerman MI, Bowman GR. FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs. *J Chem Theory Comput*. 2015. doi:10.1021/acs.jctc.5b00737
49. Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. In: *Advances in Neural Information Processing Systems 32*. ; 2019.