Supplementary Information for:
Generative Capacity of Probabilistic Protein Sequence Models

Francisco McGee Quentin Novinger Ronald M. Levy Vincenzo Carnevale Allan Haldane

1 vVAE implementation

The vanilla variational autoencoder (vVAE) is a deep, symmetrical, and undercomplete autoencoder neural network composed of a separate encoder $q_\phi(Z|S)$ and decoder $p_\theta(S|Z)$, which map input sequences S to regions of a low-dimensional latent space Z and back.[1] It is a probabilistic model, and in our implementation we assume sequences will be distributed according to a unit normal distribution in latent space, $p(Z) = \mathcal{N}[0,1](Z)$, as demonstrated in Ref. 1. Training of a VAE can be understood as maximization of (the logarithm of) the dataset likelihood $\mathcal{L} = \sum_S p_\theta(S) = \sum_S \int p_\theta(S|Z)p(Z)dZ$ with the addition of a Kullback-Leibler regularization term $\mathrm{D}_{\mathrm{KL}}[q_\phi(Z|S), p_\theta(Z|S)]$, where $p_\theta(Z|S)$ is the posterior of the decoder, which allows use of the fitted encoder $q_\phi(Z|S)$ to perform efficient estimation of the likelihood and its gradient by Monte-Carlo sampling, for appropriate encoder models.

Our vVAE architecture is built on the same basic VAE architecture described in Ref. 2, which itself appears to be built on the VAE implementation provided with the Keras library.[3] It is composed of 3 symmetrical ELU-activated layers in both the encoder and decoder, each layer with 250 dense (fully-connected) nodes. The encoder and decoder are connected by a latent layer of l nodes, we use $l = 7$ in the main text. Our vVAE's input layer accepts one-hot encoded sequences, the output layer is sigmoid-activated, and its node output values can be interpreted as a Bernoulli distribution of the same dimensions as a one-hot encoded sequence. The first layer of the encoder and the middle layer of the decoder have dropout regularization applied with 30% dropout rate, and the middle layer of the encoder uses batch normalization with a batch size of 200.[2,4,5] In all inferences, we hold out 10% of the training sequences as a validation dataset, and perform maximum likelihood optimization using the Keras Adam stochastic gradient optimizer on the remaining 90%.[6] After each training epoch we evaluate the loss function for the training and validation data subsets separately. We have tested using early-stopping regularization to stop inference once the validation loss has not decreased for three epochs in a row as in previous implementations, but this led to some variability in the model depending on when the early stopping criterion was reached. To avoid this variability, and to make different models more directly comparable, we instead fixed the number of epochs to 32 for all models, since in the early stopping tests this led to near minimum training loss and validation loss, and did not lead to significant overfitting as would be apparent from an increase in the validation loss.

Our model was implemented using Keras building on the previous implementations of Refs. 2,3, however with a modification of the loss function relative to both of these, to remove a scaling factor of Lq on the reconstruction loss, which is sometimes used to avoid issues with local minima as described further below. This prefactor leads to a non-unit variance of the latent space distribution of the dataset sequences, violating our definition that the latent space distribution should be normal with unit variance, $p(Z) = \mathcal{N}[0,1](Z)$. In the next section we show that after removing the prefactor the latent space distribution is approximately a unit normal, which more closely follows the original VAE theory developed in Ref. 1. Our implementation is available at `https://github.com/ahaldane/MSA_VAE`.

To generate a sequence from the model we generate a random sample in latent space from the latent distribution $\mathcal{N}[0,1]$, pass this value to the decoder to obtain a Bernoulli distribution, from which we sample once. To evaluate the log-probability of a sequence, we use importance sampling, averaging over 1000 samples from the latent distribution $q_\phi(Z|S)$ following from the relations[7,8]

$$\begin{aligned} p_\theta(S) &= \int p_\theta(S|Z)p(Z)dZ = \int q_\phi(Z|S)\frac{p_\theta(S|Z)p(Z)}{q_\phi(Z|S)}dZ \\ &= \mathop{\mathbb{E}}_{Z\sim q_\phi(Z|S)}\left[\frac{p_\theta(S|Z)p(Z)}{q_\phi(Z|S)}\right] \approx \frac{1}{N}\sum_i^N \frac{p_\theta(S|Z^i)p(Z^i)}{q_\phi(Z^i|S)} \end{aligned} \tag{1}$$

where, Z^i are independent samples from $q_\phi(Z|S)$ and N a large number of samples. Here $q_\theta(Z|S)$ plays the role of a sampling bias function, biasing samples to regions of latent space which are likely to have generated the sequence, leading to an accurate Monte-Carlo estimate of $p_\theta(S)$. The value $p_\theta(S)$ can be converted to a unit-less statistical energy as $E(S) = -\log p_\theta(S)$ for direct comparison with Mi3 and Indep statistical energies.

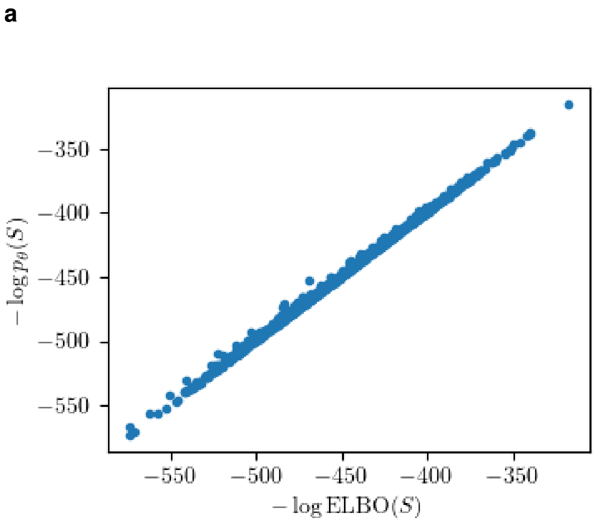

Figure 1. Comparison of $E(S) = -\log p_\theta(S)$ with the ELBO estimate for the vVAE with $l = 7$ fit to 1M sequences, evaluated for 1000 sequences S from the validation dataset, with $N = 1000$ samples for both the ELBO estimate and the $E(S)$ estimate.

Other publications have used the Evidence Lower Bound (ELBO) estimate as an approximation of $\log p_\theta(S)$,[9] and we have tested (see Fig. 1) that the ELBO and the log-probability are nearly identical, and $N = 1000$ samples is sufficient for an accurate estimate. The fact that the ELBO and log-probability are nearly identical is a sign that our encoder is well fit, as the difference between these values should equal the KL divergence $\mathrm{D}_{\mathrm{KL}}[q_\phi(Z|S), p_\theta(Z|S)]$ between the “true" posterior of the decoder $p_\theta(Z|S)$ and the approximate posterior $q_\phi(Z|S)$, which should be 0 if the encoder $q_\phi(Z|S)$ has accurately modelled the posterior.[1]

2 VAE model validation and generalization

To validate our choice of latent space size of $l = 7$ used in the main text, we tried fitting vVAEs with different latent space sizes from 2 to 10. In Fig. 2 we illustrate the latent space projections of the sequences in the training dataset. According to our specification underlying the VAE theoretically, we expect the latent space distribution to be a multidimensional normal distribution with mean 0 and unit variance. Indeed, as can be seen in the plot, and measured numerically, we generally find the latent space distribution of the dataset has close to unit variance and is approximately normal, although there is some non-normal structure in the distribution.

For latent spaces of $l = 8$ and $l = 10$ we observe that some latent dimensions appear to have "collapsed", in particular z_0 for $l = 8$, and z_1 and z_6 for $l = 10$. From repeated runs (not shown) we observe that the number of collapsed dimensions varies somewhat depending on the random seed used to initialize the stochastic optimizer, and also depends on the size of the training dataset as more dimensions collapse when fitting 1M sequences than fitting 10K sequences (not shown). For these "collapsed" dimensions, we see that the projected variance of the illustrated sequence in red in Fig. 2 is very close to 1.0, unlike in other dimensions where the projected variance is much smaller. These behaviors are consistent with a well known phenomenon of "posterior collapse" discussed in VAE literature.[10] It has been suggested that VAE posterior collapse can occur due to local minima in the likelihood function which are not global minima,[10] but in some situations can be a sign that additional latent dimensions are uninformative, and that fewer latent dimensions better represent the data.[11] We find that choosing $l = 7$ gives the best performing model which avoids posterior collapse. Interestingly, the number of “informative” latent variables, i.e. those that do not undergo posterior collapse, turns out to coincide with the intrinsic dimension (ID) of the dataset of training sequences, estimated from the set of pairwise distances using a completely independent approach.[12] In brief, it has been shown that graph distances calculated on k-neighbor graphs can be used to approximate geodesics and thus to generate the distribution of “intrinsic” distances. Close to the maximum, the latter depends exclusively on the dimension of the distance distribution’s support. This observation is used to devise a family of estimators for the ID. Using these tools, we estimated an ID of 7 or 8 for the synthetic dataset used in the main text. These numbers are consistent with what was observed in terms of collapse of the posterior distribution: the ID is seemingly related to the number of informative latent variables so that if the number of nodes in the embedding layer is increased past this number, then posterior collapse occurs, indicating that the additional variables are not

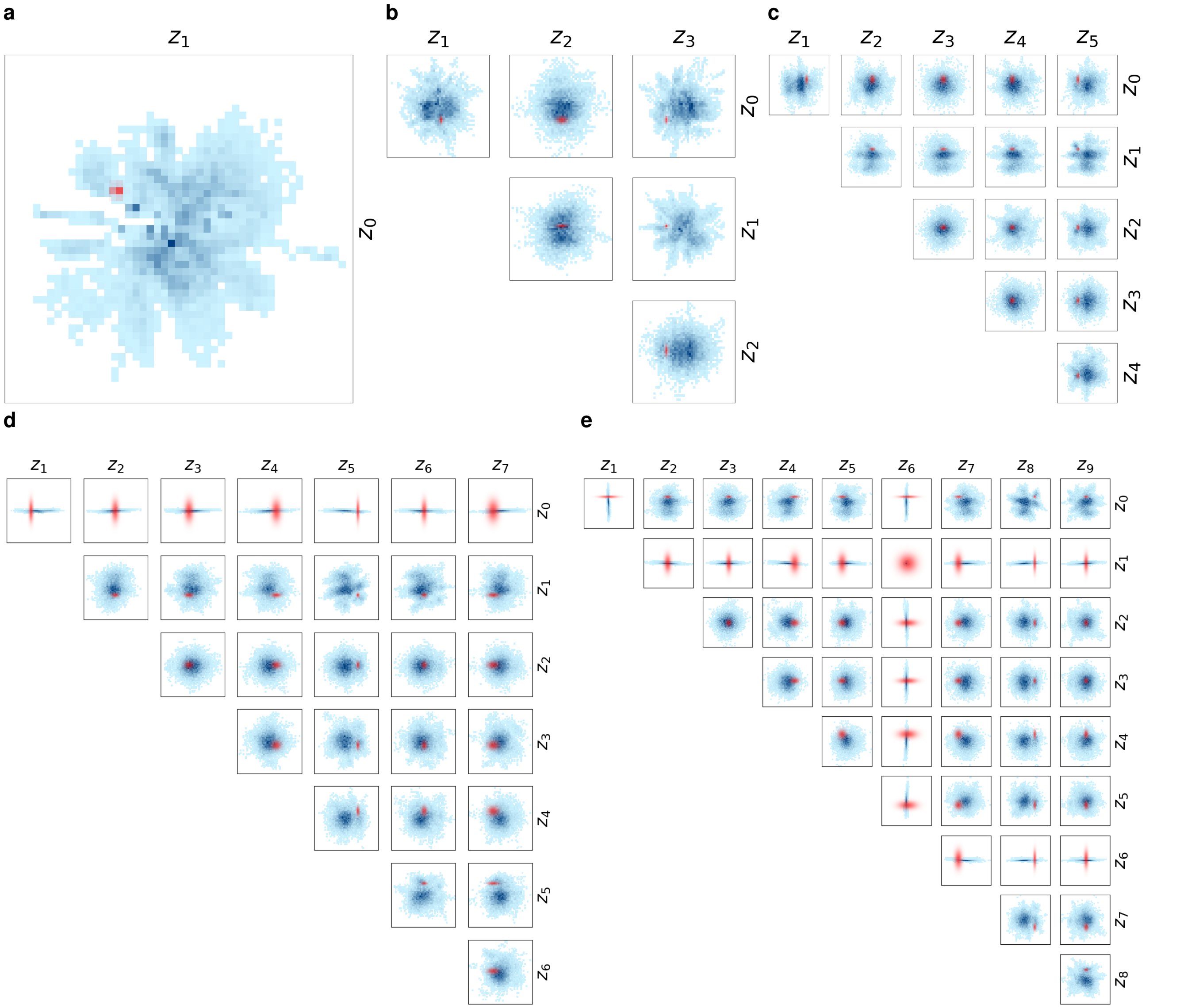

Figure 2. Plots of latent space distribution of the training dataset for vVAE models fit with different latent space sizes of 2, 4, 6, 8, and 10 (**a**,**b**,**c**,**d**,**e** respectively), fit to 1M synthetic training sequences as in the synthetic test in the main text. For each latent space size we show, for each pair of latent variables, a 2d histogram of the projected means of 10K training dataset sequences in latent space in blue. There is one subplot for $l = 2$, six subplots for $l = 4$, etc. Each plot ranges from -4 to 4 on both axes. The latent distribution $q_\phi(Z|S)$ for single random sequence from the training dataset is shown as a red shading in proportion to probability.

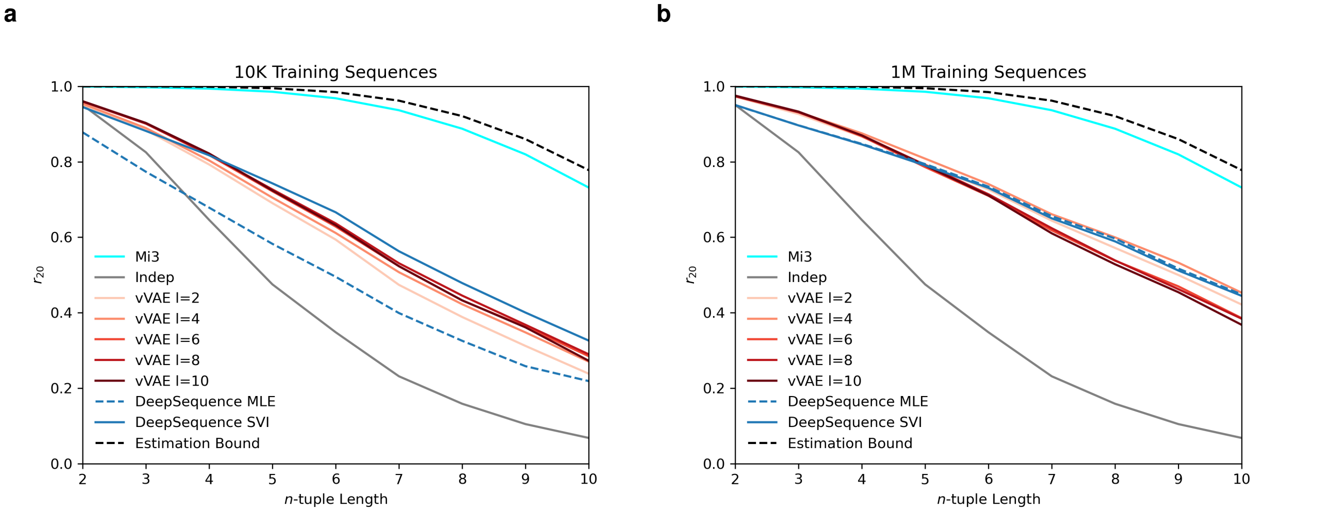

Figure 3. Performance comparison of vVAEs for different l compared to DeepSequence VAEs and Mi3 using the r_{20} metric on 10K synthetic (**a**) and 1M synthetic (**b**) training sequences.

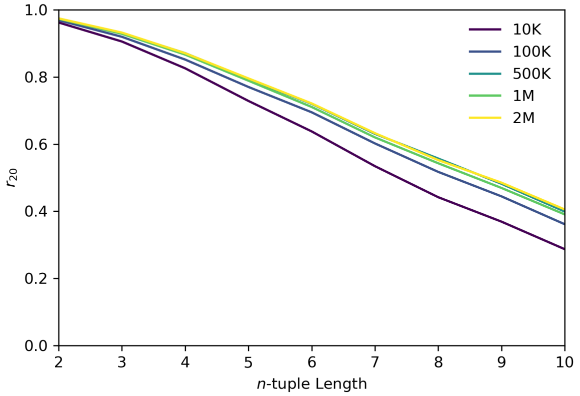

Figure 4. vVAE performance for $l = 7$ for varying synthetic training dataset sizes. For each training dataset size, two inferences are run with different random seeds, shown in solid and dashed lines for each training size.

needed to explain the data.

To compare the generative capacity of different GPSMs to determine how general our results are, we computed our MSA statistics for other VAEs besides the $l = 7$ vVAE shown in the main text. In Fig. 3 we show the r_{20} scores for different models when fit to either 10K or 1M synthetic sequences, as in the synthetic tests in the main text. We include the Mi3 and Indep models, as well as vVAEs for different latent space sizes, and also models produced using the DeepSequence VAE software which comes in two variations, the "MLE" and the "SVI" algorithms,[9] for which we use the default or example parameters. All the VAEs perform fairly similarly in this metric, including the DeepSequence VAEs. For the smaller training dataset of 10K sequence the DeepSequence SVI algorithm outperforms the other VAEs, suggesting it is less susceptible to out-of-sample error. These results suggest that our results for the vVAE shown in the main text generalize to other VAEs, including the significantly more complex DeepSequence VAE, and are not strongly dependent on implementation or number of latent variables l. The models with $l \sim 7$ perform among the best of the vVAE models for both the 10K and the 1M training datasets, though the difference between the models is small, and this further justifies our choice of $l = 7$ in the main text.

3 Minimizing VAE out-of-sample error

The goal of the synthetic test with 1M training sequences in the main text is to eliminate out-of-sample error (overfitting) by using an extremely large training dataset. How large must the training dataset be to mostly eliminate out-of-sample error for the vVAE? In Fig. 4 we show tests for the $l = 7$ vVAE for increasing training dataset sizes, finding that after 500K sequences the improvement in performance becomes small. This justifies our choice of using 1M synthetic training sequences, as there is little additional improvement to be gained by fitting to 2M sequences at the cost of increasingly prohibitive fitting time.

4 Using vVAE as the synthetic target distribution

In the main text, our synthetic GPSM tests are performed using a Potts model as the synthetic target distribution. This means that the target distribution is constructed without higher-order interaction terms, and a Potts model is by definition well specified to fit data generated from this target distribution. Here, we show GPSM performance when the synthetic target distribution instead corresponds to a vVAE, which potentially generates data which cannot be fit by a model with only pairwise interaction terms.

In this section we take the synthetic target distribution to be described by the vVAE fit in the main text from 10K natural sequences. As described in the main text, this model potentially generates patterns of higher-order mutational covariation which cannot be fit by the Mi3 model. We then follow the same procedure as for our synthetic 1M test of the main text, but using this target distribution. We generate 1M sequences from the target vVAE distribution which we use as training data for each GPSM, that is for Mi3, a vVAE, and an Indep model. We generate evaluation MSAs from each inferred model and compare it to evaluation MSAs generated from the target distribution, using our test statistics.

In Fig. 5 we show MSA test statistics for the models fit to the vVAE target. We find that the performance of Mi3 fit to this target performs at least as well as the vVAE model fit to the same target. As in the main text 1M synthetic test, the correlation scores are estimated from 500K evaluation sequences from the target and each GPSM, the r_{20} scores using 6M evaluation sequences, the Hamming distributions from 30K sequences, and the energies are evaluated for 1K sequences using 1000 Monte Carlo samples. For the r_{20} test we measure the estimation limit due to the finite size of the evaluation MSAs by computing r_{20} between two MSAs of size 6M generated from the target distribution. There is a small difference between the estimation error limit and the Mi3 result, which may be due to out-of-sample error due to the finite 1M training data, or due to specification error, and this difference is smaller than the difference of the vVAE fit to the same target distribution (red). In sum, we interpret these tests to show that the Mi3 model is able accurately fit the vVAE target distribution.

5 How higher-order covariation is represented by pairwise models

One of the questions we address in the main text is whether different GPSMs are wellmspecified to describe protein sequence variation, especially in the case of covariation of many positions in the sequence at once. Of particular interest is whether a model which explicitly includes only pairwise interactions, such as the Potts model, is sufficient to model higher order epistasis, or whether GPSMs with more complex functional forms, such as the vVAE, are necessary.

For clarity, we give a brief example describing how Potts models can predict many patterns of higher-order covariation, meaning triplet and higher patterns of residue covariation, despite only modelling pairwise interactions. We illustrate this using a toy model describing sequences of length $L = 3$ with two residue types A and B, with $2^3 = 8$ possible sequences, and show different forms of higher-order covariation which a pairwise model can and cannot fit. We refer to Refs 13–15 for detailed discussion of these issues and theoretical results suggesting why pairwise models are often sufficient to model many datasets.

First, we show how such a Potts model generates triplet covariation. Consider a Potts model with parameters given by $J^{12}_{AA} = J^{23}_{AA} = -s$ for some interaction strength s and all other field and coupling parameters are 0. This directly couples the character "A" between positions (1,2) and also positions (2,3). These interactions cause pairwise covariation between the directly coupled residues, and in the limit of large s we find $C^{12}_{AA} = C^{23}_{AA} = 0.08$, or 8%, but they also cause covariation between the indirectly coupled pair, as $C^{13}_{AA} = 0.04$, or 4%. Furthermore, this Potts model predicts three-body covariation, as can be seen by computing the three-body covariation terms found in cluster expansions in statistical physics given by

$$C^{123}_{\alpha\beta\gamma} = f^{123}_{\alpha\beta\gamma} - f^{1}_{\alpha}C^{23}_{\beta\gamma} - f^{2}_{\beta}C^{13}_{\alpha\gamma} - f^{3}_{\gamma}C^{12}_{\alpha\beta} - f^{1}_{\alpha}f^{2}_{\beta}f^{3}_{\gamma} \tag{2}$$

and we find that $C^{123}_{AAA} = 0.024$, or 2.4%, which is nonzero. This shows that a Potts model generates and can fit higher-order covariation between sets of residues even though the interactions are only pairwise, as a result of indirect covariation through chains and loops of pairwise interactions.

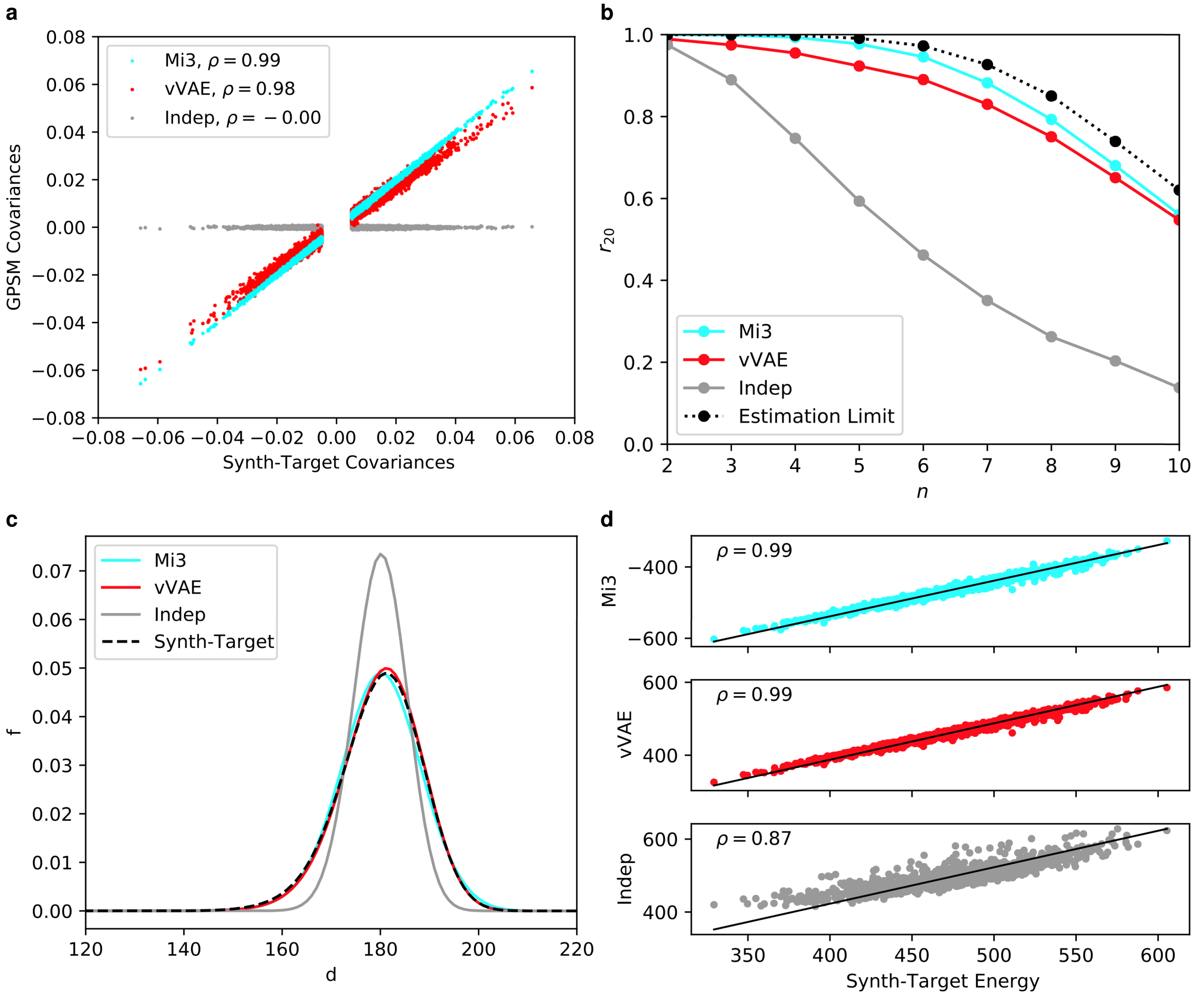

Figure 5. Synthetic test of the performance of different GPSMs when the synthetic target distribution is specified by a vVAE. **a** Pairwise covariance correlation scores, as in main text Figure 2a. **b** r_20 scores, as in main text Figure 2d. **c** Hamming distance distributions, as in main text Figure 3a. **d** Statistical energy scores, as in main text Figure 4 panels a, c, e.

AAA
ABB
BAB
BBA

Table 1. Example MSA following the XOR pattern.

An example of MSA triplet statistics which a Potts model is mis-specified to describe is the XOR pattern in which the dataset is composed in equal proportions of copies of the four sequences shown in Table 1. These sequences follow the XOR function in boolean logic, so that the 3rd position is the XOR function applied to the first two positions. One can see that both the A and B residues have a 50% probability at each position, and that for each pair of positions the probability of each of the four combinations AA, AB, BA, BB is 1/4. This means that the pairwise covariances $C^{ij}_{\alpha\beta} = 0.25 - 0.5 \times 0.5$ are all 0. Because there are no pairwise covariances, fitting a Potts model to this data will yield a model with no coupling terms, equivalent to an Indep model. Sequences generated from this (or any) Indep model have all three-body covariation terms equal to 0. However, the three-body covariations of the dataset are non-zero and $C^{123}_{AAA} = 0.125$. This shows how a Potts model fit to XOR data will fail to reproduce the correct three-body covariations. More generally, it will fail to model data which follows a boolean parity function, which generalizes the XOR function to longer strings, and is defined so that the last character is set to "B" if there are an odd number of "B" characters in the preceding sequence.

A motivation for the VAE is that it may potentially be able to model patterns of covariation such as the XOR pattern which a Potts model cannot. Whether a VAE is able to outperform the Potts model when fit to protein sequence data will depend on the prevalence of patterns such as XOR in the data which cannot be fit by a Potts model. If they are undetectable, the Potts model will be well specified and third order parameters are unnecessary. Our results with the natural dataset in the main text suggest no evidence that the Potts model is mis-specified to our dataset, as it is able to reproduce all the MSA statistics we tested up to the limits imposed by estimation and out-of-sample error.

6 Analysis of r_{20} estimation error

When computing the r_{20} scores we are able to quantify estimation error, as can be seen by the r_{20} upper limit illustrated in Fig. 5b (black dotted line). Here we provide quantitative intuition for the behavior of the r_{20} score as a function of the evaluation MSA size N, which explains the difficulty in eliminating estimation error entirely.

Consider a particular set of positions for which we estimate the frequency f of each subsequence at those positions in the target distribution, based on a finite MSA of size N generated from the target distribution, giving estimated marginals $\hat{f}$. We retain only the top twenty observed subsequences for use in the r_{20} computation. The statistical variance in $\hat{f}$ caused by finite-sampling error will be $f(1-f)/N$, following a multinomial sampling process, and we will approximate that all top 20 marginals have similar magnitude and we approximate this error as $\langle f \rangle (1 - \langle f \rangle)/N$ for all twenty values, where $\langle f \rangle$ is the mean value of the top 20 marginals.

We can then approximate that the expected Pearson correlation ρ^2 between values estimated from two such MSAs will be $\rho^2 \approx \chi^2/(\chi^2 + \sigma^2)$ where χ^2 is the variance in the values of the top 20 marginals (reflecting the variance of the "signal"), and $\sigma^2 \approx \langle f \rangle (1 - \langle f \rangle)/N$ is the statistical error in each value (representing the variance of the "noise").

$\langle f \rangle$ and χ are properties of the protein family being modelled, at each position-set, and do not depend on N. This invariant allows us to extrapolate, since if we solve for $\langle f \rangle (1 - \langle f \rangle)/\chi^2 = N(1/\rho^2 - 1)$, the rhs should be invariant when we change the size of the dataset MSA from N to N_0 or vice versa. If we estimate the rhs for a particular N_0 and measured ρ_0 numerically, we can solve for ρ at higher N since $N(1/\rho^2 - 1) = N_0(1/\rho_0^2 - 1)$, or

$$N = N_0 \frac{\rho^2/(1-\rho^2)}{\rho_0^2/(1-\rho_0^2)}. \qquad (3)$$

The approximations we used to derive this formula will become more accurate for larger N_0. We have tested this formula by predicting the expected r_{20} for MSAs of size N by extrapolating based on the measured r_{20} for MSAs of smaller size N_0, and find it is quite accurate.

This equation shows how extremely large MSAs can be required to reduce estimation errors when evaluating r_{20}, as the extrapolated N diverges as $\propto 1/x$ as $x = 1 - \rho^2$ approaches 0. For instance, if with an MSA of 6 million sequences we obtain $r_{20} = 0.8$, then we would require 28.5 million sequence to obtain $r_{20} = 0.95$ and 148 million to reach $r_{20} = 0.99$.

7 Typical natural sequence dataset MSA size

The 10K sequence training datasets we use in the main text are meant to illustrate performance for typical protein family dataset sizes. The size of 10K sequences is the number of estimated effective sequences N_{eff} remaining after curation and phylogenetic filtering for the 20th most frequent protein (Cadherin) in Pfam (Fig.6, right).[16] Some of our measurements show significant out-of-sample error for Mi3 and vVAE based on training sample size alone, suggesting that the vast majority of GPSMs training on natural data could be subject to the level out-of-sample error reported in our results.

In Pfams's Top 20 most frequent protein domains, ranked by total number of sequences, there are between 10^5 and 10^6 total sequences each (Fig.6, right). In this work, we use the 4th most frequent protein out of this ranking, Pkinase. After curation and phylogenetic filtering of the kinase, we retained only N_{eff} ~22K, or ~5% of the original ~424K kinase sequences (Fig.6, left). Extending this fraction of ~5% to the other Top 20 proteins, we estimate that N_{eff} is capped at $\sim 10^5$ (100K) for GPSMs trained on single domains, and that proteins outside the Top 20 can generally expect $N_{\text{eff}} < 10^4$ (10K). This tabulation of Pfam data demonstrates that, for the vast majority of proteins with publicly available natural sequence data, contemporaneous GPSMs must have approximately $N_{\text{eff}} < 10K$ for training, validation, and testing.

References

1. Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]* (2014). URL http://arxiv.org/abs/1312.6114. ArXiv: 1312.6114.
2. Sinai, S., Kelsic, E., Church, G. M. & Nowak, M. A. Variational auto-encoding of protein sequences. *arXiv:1712.03346 [cs, q-bio]* (2018). ArXiv: 1712.03346.
3. Chollet, F. *et al.* Keras (2015). URL https://github.com/fchollet/keras.
4. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15**, 1929–1958 (2014).
5. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]* (2015). ArXiv: 1502.03167.
6. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (2017). URL http://arxiv.org/abs/1412.6980. ArXiv: 1412.6980.
7. Kingma, D. P. & Welling, M. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning* **12**, 307–392 (2019). URL http://dx.doi.org/10.1561/2200000056.
8. Ding, X., Zou, Z. & Brooks Iii, C. L. Deciphering protein evolution and fitness landscapes with latent space models. *Nature Communications* **10**, 5644 (2019). Number: 1 Publisher: Nature Publishing Group.
9. 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).
10. Lucas, J., Tucker, G., Grosse, R. & Norouzi, M. Understanding posterior collapse in generative latent variable models. *ICLR 2019 Workshop DeepGenStruct* (2019).
11. Dai, B., Wang, Z. & Wipf, D. The usual suspects? reassessing blame for vae posterior collapse. In *ICML 2020* (2020). URL https://www.microsoft.com/en-us/research/publication/the-usual-suspects-reassessing-blame-for-vae-posterior-collapse/.
12. Granata, D. & Carnevale, V. Accurate Estimation of the Intrinsic Dimension Using Graph Distances: Unraveling the Geometric Complexity of Datasets. *Scientific Reports* **6**, 31377 (2016).
13. Schneidman, E., Still, S., Berry, M. J. & Bialek, W. Network information and connected correlations. *Phys. Rev. Lett.* **91**, 238701 (2003). URL https://link.aps.org/doi/10.1103/PhysRevLett.91.238701.
14. Schneidman, E., Berry, M. J., Segev, R. & Bialek, W. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature* **440**, 1007–1012 (2006).
15. Merchan, L. & Nemenman, I. On the sufficiency of pairwise interactions in maximum entropy models of networks. *Journal of Statistical Physics* **162**, 1294–1308 (2016).

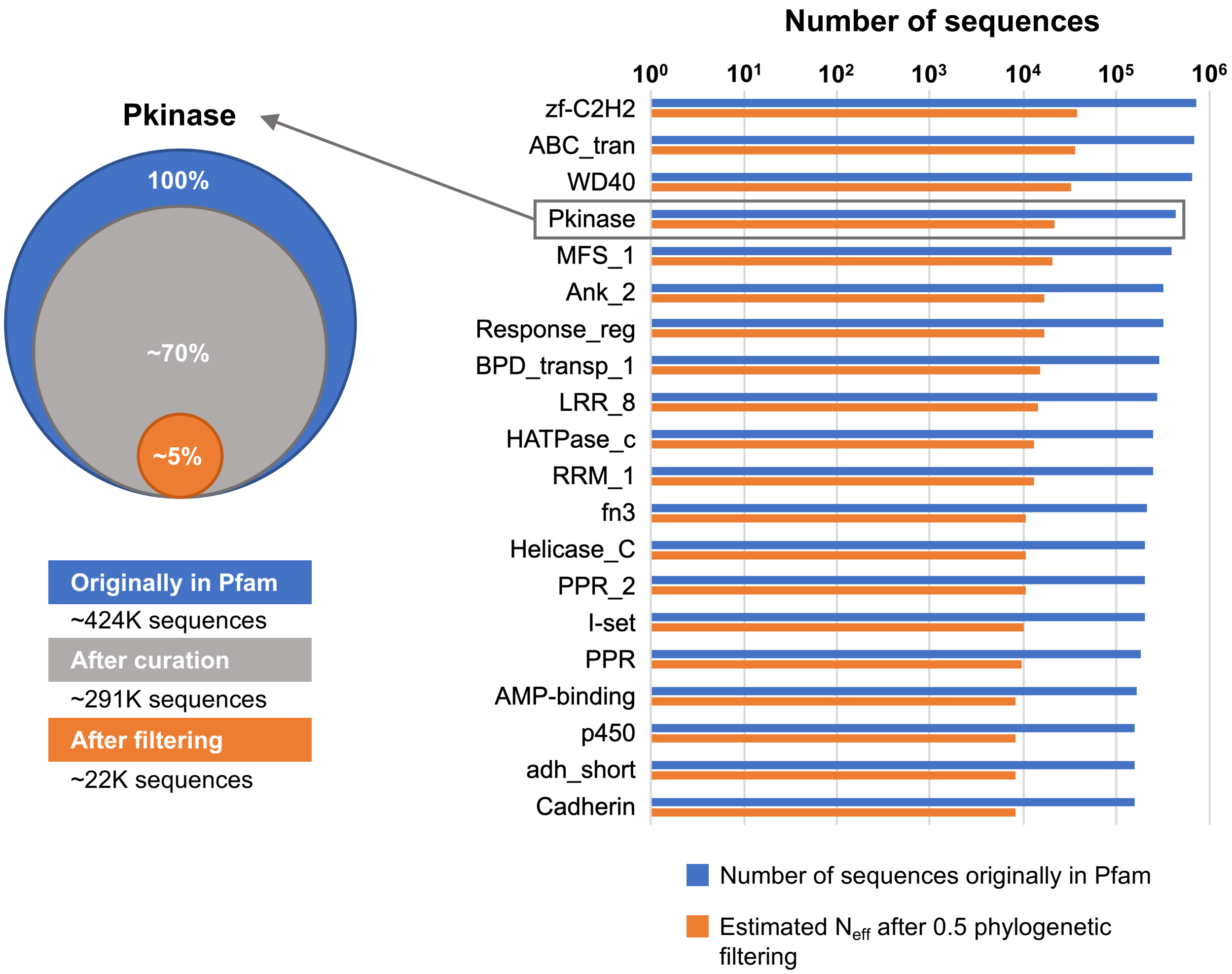

Figure 6. Pfam Top 20. GPSMs trained on publicly available natural sequence data could be inherently data-starved. **Right**, Log-scaled histogram of Pfam sequence frequencies. Sorted by the log-scaled number of sequences originally in Pfam (blue), the histogram shows estimated number of effective sequences N_{eff} after phylogenetic filtering at the 0.5 similarity cutoff (orange). All estimates are based on the actual N_{eff} for Pkinase, the fourth most frequent protein family and the one used in this work, which is ~22K sequences, or ~5% of the total ~424K Pkinase sequences in Pfam (left). Cadherin, the last entry (bottom), has $N_{\mathrm{eff}} < 10^4$ (10K sequences), meaning that this must be the approximate upper-bound of N_{eff} for GPSMs training on natural data outside the Pfam Top 20. Since all proteins outside the Pfam Top 20 must $N_{\mathrm{eff}} < 10^4$, we chose 10K sequences as the lower limit of total training sequences for our synthetic analysis. **Left**, Curation and phylogenetic filtering breakdown for Pfam Pkinase dataset. Of ~424K Pkinase sequences in Pfam (blue), only ~291K (~70%) remained after curation (grey). This curated set was phylogenetically filtered at 0.5 similarity, resulting in N_{eff} ~22K (orange), or 5% of the original ~424K.

16. El-Gebali, S. *et al.* The Pfam protein families database in 2019. *Nucleic Acids Research* **47**, D427–D432 (2019). Publisher: Oxford Academic.