Design of metalloproteins and novel protein folds using variational autoencoders

Joe G Greener[1,+], **Lewis Moffat**[1,+], **and David T Jones**[1,*]

[1]Department of Computer Science, University College London, Gower Street, London WC1E 6BT; and Francis Crick Institute, 1 Midland Road, London NW1 1AT
[+]these authors contributed equally to this work
[*]d.t.jones@ucl.ac.uk

ABSTRACT

The design of novel proteins has many applications but remains an attritional process with success in isolated cases. Meanwhile, deep learning technologies have exploded in popularity in recent years and are increasingly applicable to biology due to the rise in available data. We attempt to link protein design and deep learning by using variational autoencoders to generate protein sequences conditioned on desired properties. Potential copper and calcium binding sites are added to non-metal binding proteins without human intervention and compared to a hidden Markov model. In another use case, a grammar of protein structures is developed and used to produce sequences for a novel protein topology. One candidate structure is found to be stable by molecular dynamics simulation. The ability of our model to confine the vast search space of protein sequences and to scale easily has the potential to assist in a variety of protein design tasks.

Introduction

The computational design and redesign of proteins provides a route to create new protein structures and functions[1,2]. The 'inverse folding problem' of finding a sequence that folds to a given structure or carries out a given function is challenging due to the vast number of possible sequences, the difficulty of assessing the suitability of a sequence, and the marginal stability of protein structures[3]. From early attempts to add functional motifs to existing structures[4], the field has progressed through design of a novel fold[5] to the use of high-throughput assays to test thousands of sequences simultaneously[6,7].

Metalloproteins have been shown to be incredibly abundant and important to cell function[8]. Designing metalloproteins offers the potential not only to improve our understanding of their function but also to develop new applications within research and industrial settings. Designing metal binding sites in proteins is a relatively mature research area[4,9] and a variety of methods have been succesfully applied ranging from the deterministic to probabilistic[10]. For example, random mutagenesis has recently shown to be succesful in developing metalloproteins[11]. A range of machine learning techniques have been applied to site prediction in the past decade[12–15] although they are not prevalent within the design task outside of computationally validating designed sequences before experimental characterization.

Another task within the realm of protein design is designing entirely new protein topologies. It is challenging but opens up new applications as well as exploring the limits of the available fold space[1,5,16,17]. Similar to metalloproteins, the design process also has not yet seen pervasive use of machine learning algorithms.

Typical workflows for the design of novel metalloproteins, and novel protein topologies, consist of designing a complex template followed by computational selection of the best designs before experimental validation. For metalloproteins this also includes choosing a metal and identifying relevant binding residues. The computational selection typically relies on computationally costly force-field based molecular dynamics (MD) simulations[1,12]. This makes improving the efficiency of the computional pipeline a valuable pursuit.

Outside of protein design, another field that has seen recent development is that of deep learning research. An aspect of this development has been the rise of powerful new generative methods[18–21] that leverage deep architectures in order to learn complex distributions[22,23]. One of the most widely used deep generative methods, and the one used in this work, is the variational autoencoder (VAE). VAEs use a combination of an inference mechanism and a generative mechanism in order to learn a compressed and compact latent representation of the input data[19,20,24]. The generative mechanism can be used to sample complex synthetic data, and the inference mechanism allows sampling of synthetic data that is similar to a specified real input. Furthermore, this two mechanism structure allows for auxiliary tasks, like semi-supervised learning.

Deep generative models are only recently being applied to protein sequences[25,26] and this leaves a variety of potential applications not yet explored. For example, some work has been done using deep learning to generate protein sequences with desired features, however they do not use generative models. Instead they use deterministic deep models that take in a variety of engineered features to produce a sequence[27,28]. One example of using deep generative models to produce proteins has been published[25]. This uses an autoregressive recurrent neural network trained on a database of antimicrobial peptides, however it does not take into account structure explicitly or learn a compact latent representation of the data.

In this work, a deep generative model is presented using conditional variational autoencoders (CVAE) for small single domain protein sequences. This model aims to reduce the initial search space by learning a distribution that narrows down the space of all possible proteins to a smaller space of proteins that are more likely to be natively ordered. This work explores two applications. Firstly, unsupervised design of metal binding sites in pre-existing proteins, and secondly producing proteins that fold into a novel topology.

For the metalloprotein design task the model is conditioned on metal binding ability for 8 different metals. Proteins are then designed by adding a new metal binding site where there had not been one. The second task takes inspiration from similar work which uses VAEs with data described by a grammar[29]. Instead of producing novel small molecules like that study[29], the second task produces protein sequences. In order to condition the model in this task on a specified protein fold, a discrete representation of protein tertiary and secondary structure was required. This requirement was met by developing a context-free grammar (CFG) for protein fold structure. The grammar uses a 'periodic table' of protein folds[30] from which a vectorized structural representation can be drawn. This task also uses both the generative and inference aspects of the VAE in an iterative method to explore the latent space, allowing sampling of sequences demonstrating a particular desired attribute.

Code and documentation required to reproduce the results of the paper and generate further sequences, along with a copy of the trained model, is freely available at https://github.com/psipred/protein-vae.

Methods

Data collection

Proteins are extracted from the Protein Data Bank (PDB)[31] with the following criteria: monomers only, chain length no greater than 120 residues and no DNA/RNA present. Only chains with single domains are retained by comparison with CATH[32]. This gives 3,785 protein chains. Homologues were found for each chain by running blastp[33] against the UniRef90 dataset of available sequences clustered at the 90% similarity level[34]. An E-value threshold of 10^{-3} was used and up to 50 hits were added to the dataset for each PDB structure. The length of blastp hits was limited to 80%-120% of the query sequence with an upper limit of 140 residues.

Information on metal binding is collected from MetalPDB[8] for 8 metals - Fe (bound to 10% of proteins in dataset), Zn (6%), Ca (4%), Na (2%), Cu (2%), Mg (1%), Cd (0.9%) and Ni (0.5%). For the model conditioned on metal binding sites, another sequence is added to the dataset to act as a negative training example for each sequence in the dataset that binds a metal. This sequence is identical except that metal binding residues identified in MetalPDB are mutated to a random amino acid drawn from a distribution where each amino acid has the same frequency as in the whole training set. For homologous sequences the residues in the corresponding places in the multiple sequence alignment are mutated if they are the same amino acid as in the PDB structure. The dataset used for the metal binding model has 148,000 entries including the homologous sequences and negative training examples.

To construct the hidden Markov model (HMM) used for comparison to the VAE, hmmbuild from HMMER[35] was run with default parameters on a multiple sequence alignment (MSA) generated from the results of a blastp query of the sequence.

Context free grammar

A grammar for protein structures allows conditioning of the output of a VAE. We base our grammar on Taylor's 'periodic table' of protein structures[30]. This considers protein structures to belong to one of three basic forms (αβα layer, αββα layer and αβ barrel) with secondary structural elements added or removed to form individual topologies. The formal context-free grammar that describes the topology strings is expressed as $G = \{N, \Sigma, P, S\}$ where:

- Non-terminal symbols, $N = \{$
  '<topology>',
  '<element>',
  '<orientation>',

‘<layer>’,
‘<position>’
}

- Terminal symbols, $\Sigma = \{$
  ‘+’, ‘-’,
  ‘A’, ‘B’, ‘C’, ‘D’,
  ‘+0’, ‘+1’, ‘+2’, ‘+3’, ‘+4’, ‘+5’, ‘+6’,
  ‘-1’, ‘-2’, ‘-3’, ‘-4’, ‘-5’, ‘-6’
  }
- Production rules, $P = \{$
  ‘<topology>’ → ‘<element><topology>’,
  ‘<topology>’ → ‘<element>’,
  ‘<element>’ → ‘<orientation><layer><position>’,
  ‘<orientation>’ → [‘+’, ‘-’] (2 rules),
  ‘<layer>’ → [‘A’, ‘B’, ‘C’, ‘D’] (4 rules),
  ‘<position>’ → [‘+0’, ‘+1’, ‘+2’, ‘+3’, ‘+4’, ‘+5’, ‘+6’, ‘-1’, ‘-2’, ‘-3’, ‘-4’, ‘-5’, ‘-6’] (13 rules)
  }
- Start symbol, $S =$ ‘<topology>’

For example, the reductase-related bacterial protein with PDB ID 2CU6 fits form 0-3-2 (αβα) with topology string ‘-C+0+B+0-B-1+C-1-B-2’, where B and C are the layers prefixed by their relative orientation to the first strand in the sheet and suffixed by their position relative to the first element in each layer[30]. This example is shown in Figure 1.

Proteins in our dataset were assigned to topology strings using an assignment of topology strings to SCOP folds[36,37]. Each protein was compared to all assigned representative SCOP proteins and the highest TMAlign score above 0.6, indicating a high chance of the same fold, was used to select the topology string to assign. This allowed assignment for 65% of proteins in the dataset described above to 325 unique topology strings. This 65%, equating to 105,000 sequences including homologues, was used to train the CVAE for the fold generation model.

Model

In this section we describe the how a variational autoencoder can be used to produce novel protein sequences when conditioned on an attribute like a grammar or a metal code. VAEs simultaneously train two models, an inference model $q_\phi(z|x)$ and a generative model $p_\theta(x|z)p_\theta(z)$ for data x and the latent variable z. For this application both models are also conditioned on a chosen attribute of the sequences, a - see Figure 2. Both models are jointly optimized using the tractable variational Bayes approach which maximizes the evidence lower bound (ELBO).

$$\log p(x|a) \geq E_{q_\phi(z|x,a)}\left[\log \frac{p_\theta(x|z,a)}{q_\phi(z|x,a)}\right] = \mathcal{L}(\theta,\phi;x) \tag{1}$$

$$= E_{q_\phi(z|x,a)}\left[\log p_\theta(x|z,a)\right] - KL(q_\phi(z|x,a)||p_\theta(z)) \tag{2}$$

This equates to minimizing the reconstruction loss on x and the Kullback-Leibler (KL) divergence between the inference model and a prior $p(z)$ usually characterized by an exponential family distribution, most typically a standard Gaussian.

$$q_\phi(z|x,a) = \mathcal{N}(z|\mu_q(x,a),\sigma_q^2(x,a)) \tag{3}$$

$$p_\theta(z) = \mathcal{N}(z|0,\mathbb{I}) \tag{4}$$

Using the reparameterization trick[19,20], which incorporates the prior using a linear transform of noise, ensures that both models are differentiable and can be optimized using stochastic gradient descent (SGD).

In terms of layerwise construction of the model we define a linear block (LB) as the following

$$\mathtt{LB}(x) = \mathtt{ReLU}(\mathtt{BN}(Wx+b))$$

Where ReLU is a rectified linear unit and BN is batch normalization[38]. It was found that inclusion of batch normalization improved reconstruction accuracy, which has been documented particularly in the case of VAEs[39].

We further define a multi-layer perceptron (MLP) as three sequential LBs. Both the decoder and the encoder contain one MLP. In the case of the decoder, the MLP is followed by a linear layer with a sigmoid activation function that produces the output sequence. The MLP in the encoder produces the Gaussian parameters as follows, where it is assumed a and x are concatenated as y:

$$\mu = \texttt{Lin}(\texttt{MLP}(y)) \tag{5}$$

$$\sigma^2 = \texttt{Softplus}(\texttt{Lin}(\texttt{MLP}(y))) \tag{6}$$

Where `Lin` is a linear layer. Different sizes of latent space were tried and the final chosen for sequence generation is 16 dimensions. The sizes of the layers within the MLP are 512, 256, & 128 units. The order as presented is used for the encoder but reversed for the decoder.

Each sequence is represented as a series of one-hot encoded tensors with 22 possible positions. This represents the 20 amino acids, a padding symbol to reach max length, and a symbol for any non-standard amino acid such as selenocysteine. Given $L_{max} = 140$ the sequences are sized as 140×22 before being flattened to 3080 at which point they are inserted into the network while being concatenated with a tensor representing the conditioning attribute. In the case of metal binding sites this is a 1D tensor containing 8 values. Each is set to either zero or one to denote the absence or presence respectively of one of the eight metal binding sites. For the structure grammar this is a 1265 long flattened tensor of the one-hot encoded rules. Each rule is described by a one-hot tensor with 23 options, corresponding to the 22 rules P and a blank rule for padding, and up to 55 rules are able to produce any given topology in the dataset. When converting produced sequences to their amino acid representations tensors are resized from 3080 to 140×22 and then the $\arg\max$ is taken across the second dimension to choose the most likely amino acid for each position, leaving a sequence 140 long.

The model was optimized using an Adam optimizer[40] with a learning rate of 5×10^{-4}, a batch size of 512, and β values of 0.9 and 0.9. These values were optimized by grid search. The KL divergence was set to zero at the beginning of training before being multiplied by a linearly increasing "burn-in" factor until the full KL loss was incorporated. This is to avoid the documented behavior of the KL loss turning off layer units early in training[39].

Models were trained until convergence. Furthermore all models were built using the Pytorch[41] framework for the Python programming language. Training time for the final model took roughly 6 hours to converge using a NVIDIA GTX 1080 TI GPU.

Sampling procedure - structure task

With a maximum sequence length of 140 residues there are 20^{140} possible proteins implying that even a reduced sequence space for a particular structure will likely contain a very large number of potential sequences[42]. It has also been shown that within the protein space the optimal set of sequences for a particular structure is surprisingly small[43]. This being the case a sequential sampling-analysis-sampling method was used to traverse the space towards finding the region containing sequences that belonged to this optimal set.

For designing a new topology, the generator was sampled to produce 1,000 different sequences from across the space. From there sequences were analyzed, as described below, to find the best sequence of those generated. This sequence is then fed into the inference mechanism to find the mean and standard deviation tensors that define its latent code. From these are then used to sample 1,000 more samples using the generative network. This is analogous to searching the space around the best sequence to find potentially better sequences in the local area. This process is repeated several times to traverse the search space before finding a final best sequence - see Figure 3.

In order to select promising sequences at each iteration, generated sequences are threaded onto a relevant template structure for the desired fold using SCWRL4[44] with default options. Energy minimisation is carried out using the Rosetta relax protocol[45] with the thorough option and 3 structures generated per sequence. The top 5 sequences by score are taken forward for ab initio structure generation using Rosetta. Fragment generation is carried out and 10,000 structures are generated using the AbinitioRelax protocol with default parameters. Structures close to the template are found using TMAlign and the best sequence is input to the VAE in a new cycle. Each iteration step takes around 500 hours of computation on a single CPU but is easily parallelized, making the iteration process computationally tractable.

Sampling procedure - metalloprotein task

For design of metalloproteins the sampling process is simpler as removing or adding a binding site is done with a preexisting sequence. The protein chosen is fed into the inference mechanism with its conditioning attribute and 1,000 samples are then drawn from the generator using its mean and standard deviation tensors.

To select the best candidate sequence, while limiting human oversight, a neural network classifier was trained to predict metal binding potential using the same dataset employed in training the VAE. Instead of having the metal binding code c as the conditioning set it was used as a target set in a supervised fashion i.e. predicting c given sequence x.

The classifier was built using the same linear blocks used for the VAE (six sized 1024, 512, 256, 128, 64 & 8 hidden units respectively) where the last layer utilized a sigmoid activation function instead of a ReLU function. This is because predicting metal binding is a multi-class classification task, each class being the protein's ability to bind one of eight metals. The loss function used was binary cross entropy and optimization was carried out using the same parameters as the VAE. The inverse ratio of the number of proteins that bind a given metal was used to weight the loss function to reduce the effect of imbalanced classes.

Cross-validation was performed by creating a 90%/10% split between the training set and a validation set. To prevent proteins from the same families existing in both sets ECOD[46] was used. This was done by making sure no protein sequence in the validation set was a member of the same ECOD family as any protein in the training set. To prevent overfitting during training early stoppage was performed by saving after every epoch and choosing the model that best minimized the validation loss. The MDM2 and TSG-6 sequences explored in the results were also placed in the validation set to avoid them being used for training, though only non-metal binding sequences are present for these proteins. The trained classifier was given the 1,000 sampled sequences and the sequence with the highest predicted metal binding potential for the selected metal was chosen as the final candidate sequence.

Molecular dynamics

All molecular dynamics (MD) runs were carried out using the GROMACS package[47]. Energy minimisation was conducted using a steepest descent energy minimisation of 5,000 steps in a vacuum and the OPLS-AA force field. MD runs were conducted using periodic boundary conditions, SPC water, charge-neutralising counter ions, the OPLS-AA force field and a 2 fs timestep. An initial energy minimisation was followed by a constant temperature and volume equilibration for 100 ps, then a constant pressure and temperature equilibration for 100 ps. Production MD was run for 200 ns.

Results

In order to carry out protein design tasks we developed a CVAE that is able to generate protein sequences with certain properties. At its simplest the model is able to act as an encoder and decoder of protein sequences, with a mean sequence identity between training set sequences and their encoded-decoded form of 49.5% for the model conditioned on metal binding sites. Example sequences generated by the model are shown in Figure 4. The conservation of residues across protein families is broadly reproduced by the model. Residues conserved by evolution are generally not varied by the model, whereas residues not conserved by evolution are varied in the sequence output. For example, a MSA was formed from 1,000 sequences generated from one lysozyme sequence (PDB ID 1IIZ) in the dataset using the model conditioned on the grammar. The conservation per residue as given by Jensen-Shannon divergence[48] shows a Pearson correlation coefficient of 0.72 with values from a MSA from a blastp query of the sequence.

Generation of metal binding sites

The model is able to add potential metal binding sites to proteins. The SWIB domain of human MDM2, a protein in the dataset used to train the model, was investigated as it is not known to bind metal ions and it has high sequence identity when encoded and decoded by the model. 1,000 sequences were generated by encoding and decoding this sequence using the VAE with the copper flag turned on, i.e. copper binding was requested. In order to explore the benefits of using a VAE to generate sequences, we compared these sequences to those generated by a HMM. A HMM produces sequences from an MSA assuming a Markov process with unobserved states. Sequences generated with the VAE using the MDM2 sequence as input show less variability than sequences generated with a HMM produced from a MSA of the blastp hits of MDM2. This is expected as we are using a single sequence as input rather than a protein family.

In order to assess the stability of generated sequences in the structure of the input sequence, sequences were threaded through this structure (PDB ID 3LBL) and the Rosetta energy scores after relaxation were compared. As we are comparing different sequences adopting the same structure, a lower Rosetta energy score indicates a better suitability to the given structure. The median score is -182 energy units for the VAE sequences, -115 for the HMM sequences and -184 for the native sequence. This indicates that sequences generated by the VAE are likely to fold to the same structure as the input sequence, whereas the HMM sequences are likely to adopt a different structure or not fold to a stable structure. Hence HMM sequences show more sequence and structural variation than VAE sequences.

When a copper binding site is requested in the VAE, more copper-binding motifs[4] are observed in the output sequences than for the HMM. 10% of sequences have histidine residues close in space (8% for the HMM) and 4% of sequences have His-x$_3$-His on a α-helix (2% for the HMM). When the output sequences are limited to those with more histidines than the native sequence, 42% of sequences from the VAE have close histidines (17% for the HMM) and 16% of sequences have His-x$_3$-His on a α-helix (4% for the HMM). This indicates that the VAE is able to generate sequences with a high chance of folding to the same structure as the input sequence, yet can still add metal binding motifs to the sequences. Whilst copper-binding motifs are not being explicitly requested from the HMM, fewer copper binding motifs are seen than for the VAE despite the higher sequence variability. The ability to generate sequences with histidines close in space and with copper-binding motifs on helices shows that the VAE is learning structural information.

A discriminator trained to predict metal binding sequences (see the Methods) was used to predict the sequence with the highest copper binding character from the 1,000 generated. The top-ranked sequence has two potential new copper binding sites when compared to the input sequence, as shown in Figure 5. One of the potential sites, site A, involves the modification of a site with a histidine residue to a site with two histidines and a cysteine to form a linear motif. The other potential site, site B, involves the addition of two histidine residues close in structure but 18 residues apart in sequence to form a non-linear motif. This sequence has 54% identity to the input sequence.

Another protein, the link module of human TSG-6 (PDB ID 1O7B), was selected for similar reasons to the SWIB domain. In this case calcium binding was requested. The generated sequences show more aspartate residues, known to bind calcium[8], than the native sequence in 29% of cases. The metal binding discriminator was used to select the sequence from 1,000 generated sequences most likely to bind calcium. The second-highest ranked sequence shows a potential new calcium binding site formed of a glutamate added by the model and a native glutamate 3 residues apart, as shown in Figure 5. This sequence has 89% identity to the input sequence. In both cases above a single sequence was used as input and a sequence was returned that shows potential metal binding character. These results suggest that the model is able to add metal binding sites without having to manually examine individual protein structures.

Another test of the model's ability to add metal binding sites is whether it can add a known site back to a protein excluded from the training set. This was tested with the classic copper-binding protein plastocyanin. The instances of plastocyanin with native sequences were removed from the dataset; the instances with metal binding residues mutated to other amino acids were left in. When copper binding sequences are requested with the mutated plastocyanin sequence as input from the model trained on this new dataset, 87% of 1,000 sequences have the crucial copper-binding residue His37 present even though it was mutated in the training examples. 11% of the sequences have the full copper-binding site of His37, Cys84 and His87 present.

Generation of proteins with a given topology

Next, we explore the ability of a CVAE to generate sequences for a given protein topology. In this case, the output is conditioned on a grammar of protein structures[30]. Note that when generating sequences for a given topology only the topology is used as input; this is different to the metal binding case above when a protein sequence was used as input along with the metal binding information. First, sequences were generated for a topology common in the dataset, the two-layer sandwich with three β-strands and one α-helix (topology '+B+0-C+0+B+2-B+1' - see the Methods). In this case the sequences generated show similarity to the training sequences with the given topology, with 55% of generated sequences having 50% or greater sequence identity to at least one training sequence. Threading generated sequences through four PDB structures with the given topology (PDB IDs 1AHO, 1JZA, 2FKL and 2M8B), followed by Rosetta energy minimisation, gives energy scores that are similar to the PDB structures in many cases. 18% of generated sequences have energy scores within 10% of the native PDB scores for at least one of the above four structures, which only represent a subset of available structures of the fold. In this case the CVAE seems to be generating viable homologues of the sequences used for training.

Next, the model was used to generate sequences for a novel topology. This topology was made by picking a structure with a well-assigned Taylor topology - the reductase-related bacterial protein with PDB ID 2CU6 - and modifying the loops connecting the central β-strands with ModLoop[49]. See Figure 6. The secondary structural elements are arranged identically in 3D space but the different connection of the loops creates a new topology and leads to a large rearrangement of the sequence. The resulting topology ('-C+0+B+0-B-2+C-1-B-1') is not assigned to any structure in the PDB and a structural search with the modified protein using DALI[50] does not give any close matches, the only hits being similar in topology to 2CU6.

Sequences were generated for this topology using the model, and an iterative procedure of selecting promising sequences via ab initio structure generation and localized sampling was used to refine the sequence generation (see the Methods). After three iterations of this process a sequence of 86 residues was generated that had structures in the pool of ab initio predicted structures similar to the modified 2CU6 structure. This is shown in Figure 6. The region between the β-strand elements contains two α-helices rather than one and there is less β-strand character, but the overall topology of α-helices behind a β-strand with certain connections is correct. In fact the PSIPRED[51] prediction of the region not forming a β-sheet in the generated structure is of a β-sheet, and with minor rearrangement of the structure a β-sheet could be formed.

Although the modified 2CU6 backbone was used to guide the search, no sequence information from 2CU6 was utilised and the sequence is generated purely by the CVAE. There are no similar sequences in the PDB, searching with an E-value threshold of 10. The sequence shows some similarity only to amidophosphoribosyltransferase sequences, with the highest scoring blastp hit having 46% sequence identity over 55% of the query protein. However structures of amidophosphoribosyltransferases in the PDB show no similarity to the requested topology. Interestingly, the sequence shows 28% sequence identity to a sequence generated by RosettaDesign[52] from the generated structure backbone in Figure 6, indicating some agreement with established design protocols.

In order to probe the stability of this structure and gain information on whether the sequence adopts it, MD simulations were carried out. MD simulations can lead to stabilisation of collapsed structures other than the native structure due to favouring hydrophobic interactions[53], but they are still a useful tool for probing potential structures[54]. The structure showed little deviation in structure over three runs of 200 ns simulation time when considering root mean square displacement (RMSD) and radius of gyration. This is shown in Figure 7. Although longer simulations would be required to check more thoroughly for unfolding or for a different folded state, this is some indication that the fold is stable. Experimental testing would be required to verify if this prediction is correct. However, these results indicate that a CVAE is able to generate viable sequences for protein topologies both known and not previously seen. It is hoped that these results generalize to the vast number of topologies described by the grammar.

Discussion

This work has indicated that CVAEs are able to carry out protein design tasks by conditioning output sequences on desired properties such as metal binding sites and given topologies. The model is able to learn structural properties, even with fewer latent dimensions than the 16 used in the final model. For example, with 2 latent dimensions the homologues of each input sequence cluster together in the latent space. The sequences also show some separation by CATH class, and sequences of the same CATH architecture tend to be close in the latent space. These properties are shown in Figure 8. As would be expected, the first latent dimension correlates with molecular weight as the model learns the length of the input sequence.

Sequences generated by the model described here show similar conservation patterns to native sequence families. However, the covariation of residues present across sequence families is not reproduced by the model. This is to be expected for two reasons: each PDB sequence in the dataset is limited to 50 sequence homologues, and the input to the model is a single sequence rather than a multiple sequence alignment. There are developments to the model that could lead to sequences being generated with realistic covariation signals. These include training directly on multiple sequence alignments, having a data set with proteins that have more homologues, and using regularization techniques to promote covarying outputs. It has been shown that taking into account covariation is important for effective protein design[55,56]. Another potential use of this is to generate additional sequences for use in residue contact prediction methods.

Our model has shown promise in conditioning output sequences in terms of metal binding or a given topology. There are other properties of proteins that could be explored using the model in a similar way. These include protein-protein interactions, which have been organised systematically in an analogous way to Taylor topologies[57]; allosteric sites, which have been designed into proteins previously[16]; taking account of multiple conformations in the design process[58,59]; and designing proteins by linking together large fragments from the PDB[60]. Another approach is to use activation maximization[61] on trained models to directly move towards sampling proteins that have measurable and desired attributes. For example, sampling a sequence containing a particular motif.

After years of effort, protein design is becoming a systematic tool showing promise in many areas. However, the difficulties of navigating a vast search space and scaling design efforts from specific proteins to the whole of protein space remain a challenge. The rise of deep learning has the potential to meet this challenge due to its ability to learn complex patterns without supervision and the small computational cost of using a model once it is trained. In addition, the increasing amount of sequence[62,63] and structural[64] data will aid data-intensive methods such as deep learning. It is hoped that methods similar to the CVAE presented here will be a valuable addition to the protein design pipeline.

Data availability

Code and documentation required to reproduce the results of the paper and generate further sequences, along with a copy of the trained model, is freely available at `https://github.com/psipred/protein-vae`.

References

1. Huang, P., Boyken, S. E. & Baker, D. The coming of age of de novo protein design. *Nature* **537**, 320–327 (2016).
2. Samish, I., MacDermaid, C. M., Perez-Aguilar, J. M. & Saven, J. G. Theoretical and computational protein design. *Annu. Rev Phys Chem* **62**, 129–149 (2011).
3. Yue, K. & Dill, K. A. Inverse protein folding problem: designing polymer sequences. *Proc Natl Acad Sci USA* **89**, 4163–4167 (1992).
4. Regan, L. Protein design: novel metal-binding sites. *Trends Biochem. Sci* **20**, 280–285 (1995).
5. Kuhlman, B. *et al.* Design of a novel globular protein fold with atomic-level accuracy. *Science* **302**, 1364–1368 (2003).
6. Rocklin, G. J. *et al.* Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* **357**, 168–175 (2017).
7. Chevalier, A. *et al.* Massively parallel de novo protein design for targeted therapeutics. *Nature* **550**, 74–79 (2017).
8. Andreini, C., Cavallaro, G., Lorenzini, S. & Rosato, A. MetalPDB: a database of metal sites in biological macromolecular structures. *Nucleic Acids Res* **41**, D312–D319 (2013).
9. Andreini, C., Bertini, I. & Rosato, A. Metalloproteomes: A bioinformatic approach. *Acc Chem Res* **42**, 1471–1479 (2009).
10. Fung, H. K., Welsh, W. J. & Floudas, C. A. Computational De Novo Peptide and Protein Design: Rigid Templates versus Flexible Templates. *Ind Eng Chem Res* **47**, 993–1001 (2008).
11. Yang, H. *et al.* Evolving artificial metalloenzymes via random mutagenesis. *Nat Chem* **10**, 318–324 (2018).
12. Akcapinar, G. B. & Sezerman, O. U. Computational approaches for de novo design and redesign of metal-binding sites on proteins. *Biosci. Reports* **37** (2017).
13. Brylinski, M. & Skolnick, J. FINDSITE-metal: Integrating evolutionary information and machine learning for structure-based metal-binding site prediction at the proteome level. *Proteins* **79**, 735–751 (2011).
14. Lin, H. H. *et al.* Prediction of the functional class of metal-binding proteins from sequence derived physicochemical properties by support vector machine approach. *BMC Bioinforma.* **7** (2006).
15. Sodhi, J. S. *et al.* Predicting metal-binding site residues in low-resolution structural models. *J Mol Biol* **342**, 307–320 (2004).
16. Dagliyan, O. *et al.* Rational design of a ligand-controlled protein conformational switch. *Proc Natl Acad Sci USA* **110**, 6800–6804 (2013).
17. Taylor, W. R., Chelliah, V., Hollup, S. M., MacDonald, J. T. & Jonassen, I. Probing the "dark matter" of protein fold space. *Structure* **17**, 1244–1252 (2009).
18. Goodfellow, I. J. *et al.* Generative Adversarial Networks. *ArXiv e-prints* (2014).
19. Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *ArXiv e-prints* (2013).
20. Rezende, D. J., Mohamed, S. & Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *ArXiv e-prints* (2014).
21. van den Oord, A., Kalchbrenner, N. & Kavukcuoglu, K. Pixel Recurrent Neural Networks. *ArXiv e-prints* (2016).
22. Jaques, N., Gu, S., Turner, R. E. & Eck, D. Tuning Recurrent Neural Networks with Reinforcement Learning. *ArXiv e-prints* (2016).
23. van den Oord, A. *et al.* Conditional Image Generation with PixelCNN Decoders. *ArXiv e-prints* (2016).
24. Gomez-Bombarelli, R. *et al.* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent Sci* **4**, 268–276 (2018).
25. Müller, A. T., Hiss, J. A. & Schneider, G. Recurrent Neural Network Model for Constructive Peptide Design. *J Chem Inf Model.* **58**, 472–479 (2018).

26. Sinai, S., Kelsic, E., Church, G. M. & Nowak, M. A. Variational auto-encoding of protein sequences. *ArXiv e-prints* (2017).

27. Li, Z., Yang, Y., Faraggi, E., Zhan, J. & Zhou, Y. Direct prediction of profiles of sequences compatible with a protein structure by neural networks with fragment-based local and energy-based nonlocal profiles. *Proteins* **82**, 2565–2573 (2014).

28. Wang, J., Cao, H., Zhang, J. Z. H. & Qi, Y. Computational Protein Design with Deep Learning Neural Networks. *ArXiv e-prints* (2018).

29. Kusner, M. J., Paige, B. & Hernández-Lobato, J. M. Grammar Variational Autoencoder. *ArXiv e-prints* (2017).

30. Taylor, W. R. A 'periodic table' for protein structures. *Nature* **416**, 657–660 (2002).

31. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res* **28**, 235–242 (2000).

32. Sillitoe, I. *et al.* CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res* **43**, D376–D381 (2015).

33. Altschul, S. F. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* **25**, 3389–3402 (1997).

34. The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Res* **45**, D158–D169 (2017).

35. Eddy, S. R. Accelerated Profile HMM Searches. *PLoS Comput. Biol* **7**, e1002195 (2011).

36. Murzin, A. G., Brenner, S. E., Hubbard, T. & Chothia, C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* **247**, 536–540 (1995).

37. Taylor, W. R. Personal communication (2017).

38. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proc. 32nd Int. Conf. on Mach. Learn.* **37**, 448–456 (2015).

39. Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K. & Winther, O. Ladder Variational Autoencoders. *Adv. Neural Inf. Process. Syst.* **29**, 3738–3746 (2016).

40. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv e-prints* (2014).

41. Paszke, A. *et al.* Automatic differentiation in PyTorch. *NIPS-W* (2017).

42. Tian, P. & Best, R. B. How Many Protein Sequences Fold to a Given Structure? A Coevolutionary Analysis. *Biophys J* **113**, 1719–1730 (2017).

43. Kuhlman, B. & Baker, D. Native protein sequences are close to optimal for their structures. *Proc Natl Acad Sci USA* **97**, 10383–10388 (2000).

44. Krivov, G. G., Shapovalov, M. V. & Dunbrack, R. L. Improved prediction of protein side-chain conformations with SCWRL4. *Proteins* **77**, 778–795 (2009).

45. Leaver-Fay, A. *et al.* ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Meth Enzym.* **487**, 545–574 (2011).

46. Cheng, H. *et al.* ECOD: An Evolutionary Classification of Protein Domains. *PLoS Comput. Biol* **10**, e1003926 (2014).

47. Abraham, M. J. *et al.* GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **1-2**, 19–25 (2015).

48. Capra, J. A. & Singh, M. Predicting functionally important residues from sequence conservation. *Bioinformatics* **23**, 1875–1882 (2007).

49. Fiser, A. & Sali, A. ModLoop: automated modeling of loops in protein structures. *Bioinformatics* **19**, 2500–2501 (2003).

50. Holm, L. & Rosenstrom, P. Dali server: conservation mapping in 3D. *Nucleic Acids Res* **38**, W545–W549 (2010).

51. Buchan, D. W., Minneci, F., Nugent, T. C., Bryson, K. & Jones, D. T. Scalable web services for the PSIPRED Protein Analysis Workbench. *Nucleic Acids Res* **41**, W349–W357 (2013).

52. Liu, Y. & Kuhlman, B. RosettaDesign server for protein design. *Nucleic Acids Res* **34**, W235–W238 (2006).

53. Robustelli, P., Piana, S. & Shaw, D. E. Developing a molecular dynamics force field for both folded and disordered protein states. *Proc Natl Acad Sci USA* **115**, E4758–E4766 (2018).

54. Lindorff-Larsen, K., Piana, S., Dror, R. O. & Shaw, D. E. How fast-folding proteins fold. *Science* **334**, 517–520 (2011).

55. Socolich, M. *et al.* Evolutionary information for specifying a protein fold. *Nature* **437**, 512–518 (2005).

56. Tian, P., Louis, J. M., Baber, J. L., Aniana, A. & Best, R. B. Co-Evolutionary Fitness Landscapes for Sequence Design. *Angew Chem Int Ed Engl* **57**, 5674–5678 (2018).

57. Ahnert, S. E., Marsh, J. A., Hernandez, H., Robinson, C. V. & Teichmann, S. A. Principles of assembly reveal a periodic table of protein complexes. *Science* **350**, aaa2245 (2015).

58. Davey, J. A., Damry, A. M., Goto, N. K. & Chica, R. A. Rational design of proteins that exchange on functional timescales. *Nat Chem Biol* **13**, 1280–1285 (2017).

59. Ambroggio, X. I. & Kuhlman, B. Computational design of a single amino acid sequence that can switch between two distinct protein folds. *J Am Chem Soc* **128**, 1154–1161 (2006).

60. Jacobs, T. M. *et al.* Design of structurally distinct proteins using strategies inspired by evolution. *Science* **352**, 687–690 (2016).

61. Nguyen, A., Yosinski, J., Bengio, Y., Dosovitskiy, A. & Clune, J. Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space. *ArXiv e-prints* (2016).

62. Kunin, V., Copeland, A., Lapidus, A., Mavromatis, K. & Hugenholtz, P. A bioinformatician's guide to metagenomics. *Microbiol Mol Biol Rev* **72**, 557–578 (2008).

63. Asgari, E. & Mofrad, M. R. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLoS ONE* **10**, e0141287 (2015).

64. Callaway, E. The revolution will not be crystallized. *Nature* **525**, 172–174 (2015).

Acknowledgements

We thank members of the group for valuable discussions and comments, and William R Taylor for providing data and assistance for the protein structure grammar. This work was supported by the European Research Council Advanced Grant 'ProCovar' (project ID 695558) and the Biotechnology & Biological Sciences Research Council (BBSRC) UK, grant BB/M011712/1. This work was supported by the Francis Crick Institute which receives its core funding from Cancer Research UK (FC001002), the UK Medical Research Council (FC001002), and the Wellcome Trust (FC001002).

Author contributions statement

JGG, LM and DTJ conceived and designed the study and reviewed the manuscript. JGG and LM carried out the computational work, did the analysis and drafted the manuscript.

Additional information

Competing interests

The authors declare no competing interests.

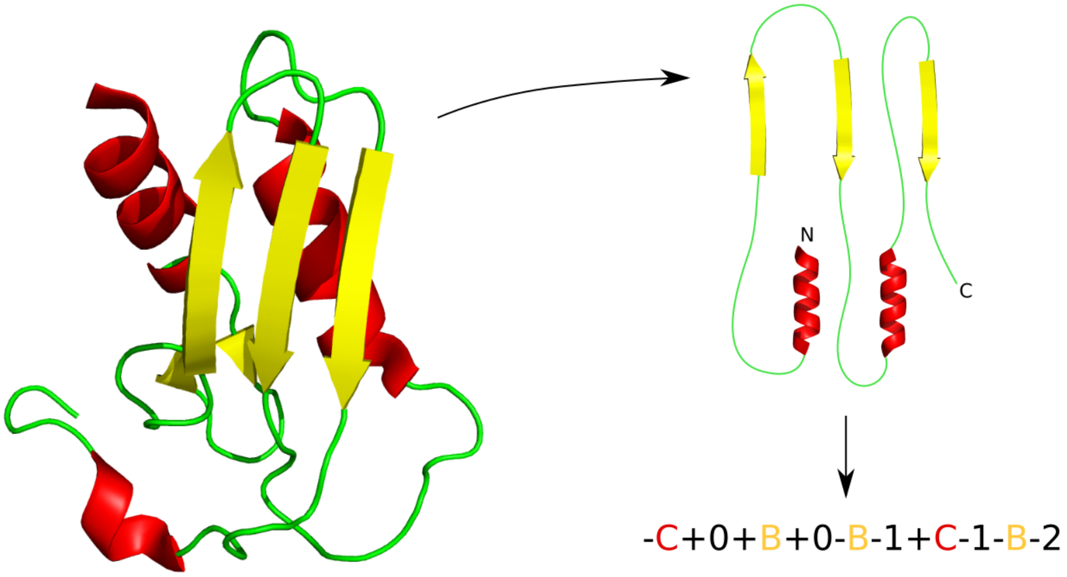


Figure 1. The Taylor grammar of protein structures shown for a reductase-related bacterial protein (PDB ID 2CU6). The orientation of the main secondary structural elements is examined in order to assign a topology string. See Taylor 2002[30] for a full description of the grammar.

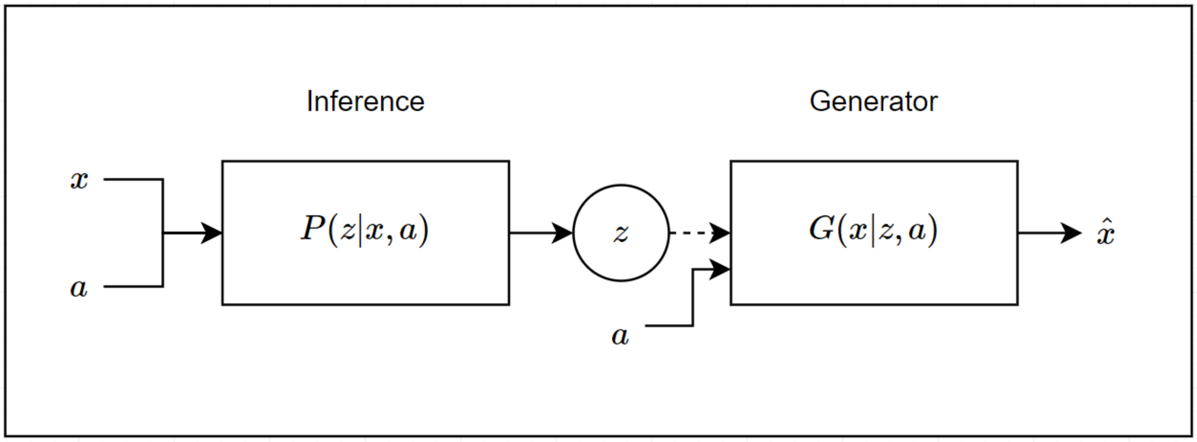


Figure 2. The inference/generator (encoder/decoder) structure of the VAE. The data x and the conditioned attribute a are concatenated and passed through the inference model to produce the latent code z. The attribute is then concatenated to the sampled z (denoted by a dashed line) which going through the generator produces the reconstructed sequence $\hat{x}$.

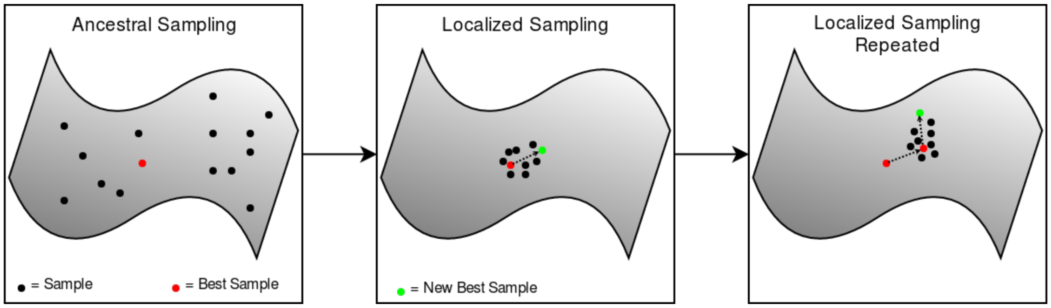


Figure 3. Graphic showing how the latent space was explored through iterative sampling and analysis of protein sequences.

A

```
AEVPSGEQLFNSNCSACHIGGNNVIISHKTLRKEALEKYAMNSLEAIRYQVVNGKNAMPAFGGRLNEEEIDAIATYVLGQAELD
                                          ↓
ADLEAGEQIFSANCAACHGGGNNIIMPEKTLKKDALEENGMKSVEAITYQVTNGKNAMPAFGGRLSDEDIEDVANYVLSQAEKGW
ADLEHGAQIFSANCAACHAGGNNVIMPDKTLKKDALEKNGMNSIEAITYQVTNGKNAMPAFGGRLSDEDIEDVANYVLSQAEKGW
ADLENGGKVFSGACAACHIGGENIVRPEKTLKKDALEEGGMDSIEAITAQVTNGKNAAPAFGERLVDEDIEDVAEYVL
ADLAAGEQIFSANCAACHAGGNNVVMPDKTLKKDALEKYGMNSIEAITTQVTNGKNAMPAFGGRLEAEDIEDVAAYVLSQAEG
ADLEHGEQIFSANCAACHAGGNNVIMPEKTLKKDALEKYGMNSVEAITTQVTNGKNAMPAFGGRLEDEQIEDVANYVLSQSEW
ADIEHGEKIFSANCAACHAGGNNAIMRNKTLKKEALEPNGMNSIEAITYQVTNGKNAMPAFGGRLSDEDIEDVANYVLKQAEKGW
ADLAAGEQIFSANCAACHAGGNNIIMPEKTLKKEALEKYSMNSIEAITTQVTNGKNAAPAFGGRLSDEDIEDVANYVLSQAEKGW
ADIITGEQIFSANCAACHIGGNNAIRPEKTLKKPALETNGMNSVDAITTQVVNPKNAMPAFGGRLEDEDIEDVANYVLSQAEK
```

B

```
              +B+0-C+0+B+2-B+1
                     ↓
ESGYAVVCDTTCSYDGECNNECTCCCLKVKQKGNDGGYCWLWECGCLCLGAPVLVPEDTKCK
KKGCLVSRGTGCGSGCSNNNCAKGLKISNGAKGKEGHRGYKCGCGCFCWPDR
CDGYLVESKTGCGFCGLNNSCCNLCCNKNGAKAGYCACGYKCKCECLPLPLPN
RDGYPVHDKGCKISCFGNNYCWKECKKKGKSKGYCYCWWLACWCYGLPDPEKVWDYA
KKGYPVVSDDCCKYCCLNNKYCNYCCNKCGAKSGYCAWCCKSGCACWCLDLPK
ERDGYIADPTNCGYTCANNSCCNGLCTKNGAKAGYCAWIGPYGKACWCIPLPDKVP
KDYYPKDDKTCCSCCFNNNYCNKECKKEGKASGYCYGWCPACWCWCLPDDE
KKGKYINDGTNCKYTCANNAKNNCCDKKCGAKGGYGHWGYPFGKACWCFPLPE
```

Figure 4. Example protein sequences generated by the model. (A) A sequence can be given as input, in which case similar sequences are returned by encoding and decoding the input sequence. (B) A topology string can be given as input, in which case sequences are generated that the VAE believes match the topology.

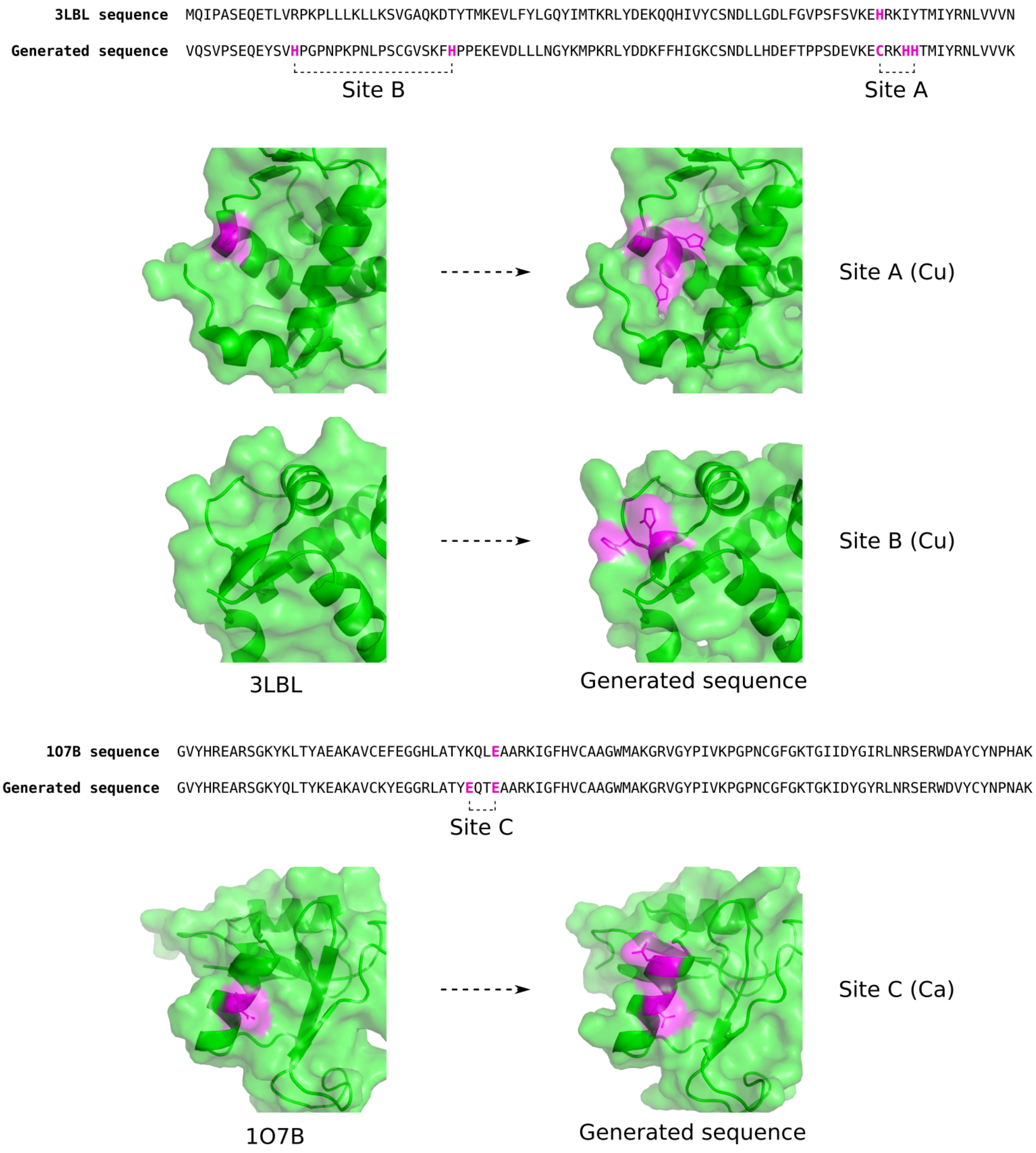
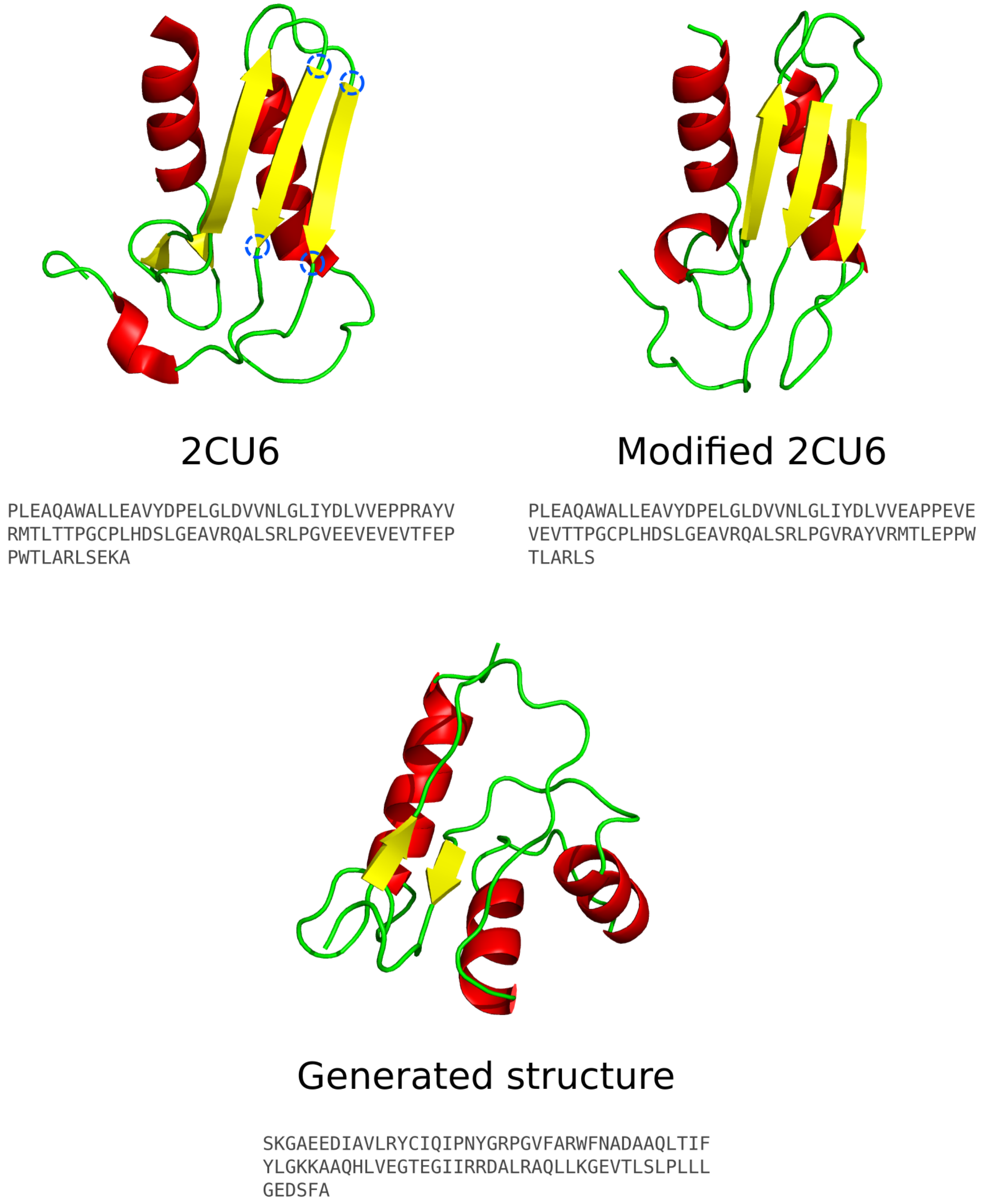

3LBL sequence MQIPASEQETLVRPKPLLLKLLKSVGAQKDTYTMKEVLFYLGQYIMTKRLYDEKQQHIVYCSNDLLGDLFGVPSFSVKEHRKIYTMIYRNLVVVN
Generated sequence VQSVPSEQEYSVHPGPNPKPNLPSCGVSKFHPPEKEVDLLLNGYKMPKRLYDDKFFHIGKCSNDLLHDEFTPPSDEVKECRKHHTMIYRNLVVVK
Site B
Site A
Site A (Cu)
Site B (Cu)
3LBL
Generated sequence
1O7B sequence GVYHREARSGKYKLTYAEAKAVCEFEGGHLATYKQLEAARKIGFHVCAAGWMAKGRVGYPIVKPGPNCGFGKTGIIDYGIRLNRSERWDAYCYNPHAK
Generated sequence GVYHREARSGKYQLTYKEAKAVCKYEGGRLATYEQTEAARKIGFHVCAAGWMAKGRVGYPIVKPGPNCGFGKTGKIDYGYRLNRSERWDVYCYNPNAK
Site C
Site C (Ca)
1O7B
Generated sequence

Figure 5. Automated addition of potential metal binding sites to two proteins. The sequence of the SWIB domain of human MDM2 (PDB ID 3LBL) and a sequence generated by the model that is predicted to have high copper-binding character are shown. Highlighted in purple are the residues that form the potential copper binding sites. The sites are shown on the structure of the generated sequence using 3LBL as a template, and compared to 3LBL. The same is shown for a potential site on the link module of human TSG-6 (PDB ID 1O7B) when calcium binding is requested.



2CU6

PLEAQAWALLEAVYDPELGLDVVNLGLIYDLVVEPPRAYV
RMTLTTPGCPLHDSLGEAVRQALSRLPGVEEVEVEVTFEP
PWTLARLSEKA

Modified 2CU6

PLEAQAWALLEAVYDPELGLDVVNLGLIYDLVVEAPPEVE
VEVTTPGCPLHDSLGEAVRQALSRLPGVRAYVRMTLEPPW
TLARLS

Generated structure

SKGAEEDIAVLRYCIQIPNYGRPGVFARWFNADAAQLTIF
YLGKKAAQHLVEGTEGIIRRDALRAQLLKGEVTLSLPLLL
GEDSFA

Figure 6. Structures to explore a novel fold. The structure and sequence of 2CU6 are shown. By remodelling the loops at the locations of the blue circles using ModLoop[49] a modified structure with a novel topology is generated. This is used as a backbone template to select structures from a pool of ab initio Rosetta structures generated from sequences output by the CVAE. The closest structure to the template is shown.

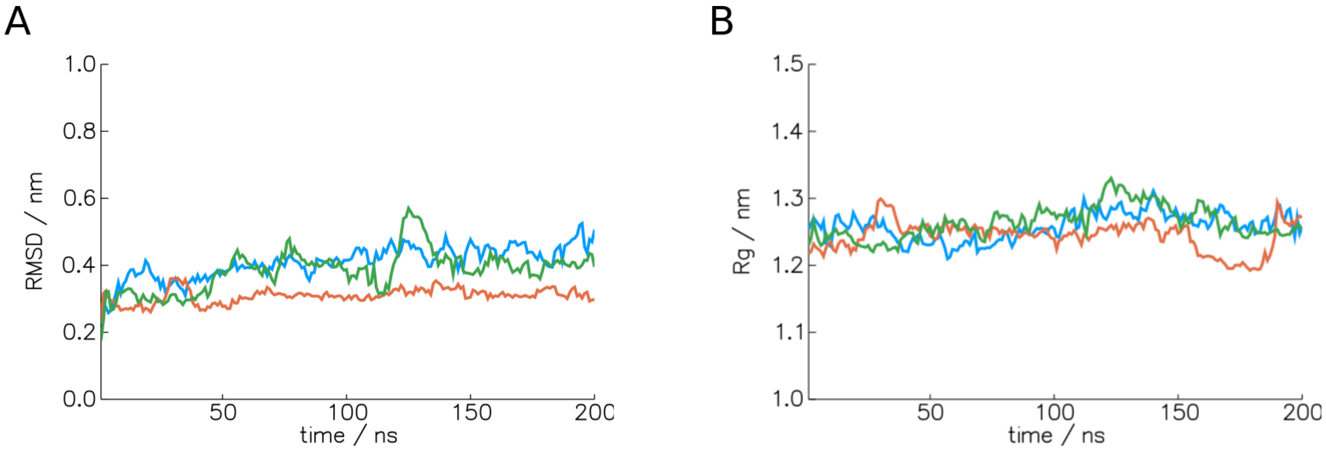


Figure 7. Analysis of MD runs of the generated structure shown in Figure 6. Three runs of 200 ns are shown in blue, orange and green. (A) Backbone RMSD of trajectory structures to the energy-minimized starting structure. (B) Backbone radius of gyration (Rg) of trajectory structures.

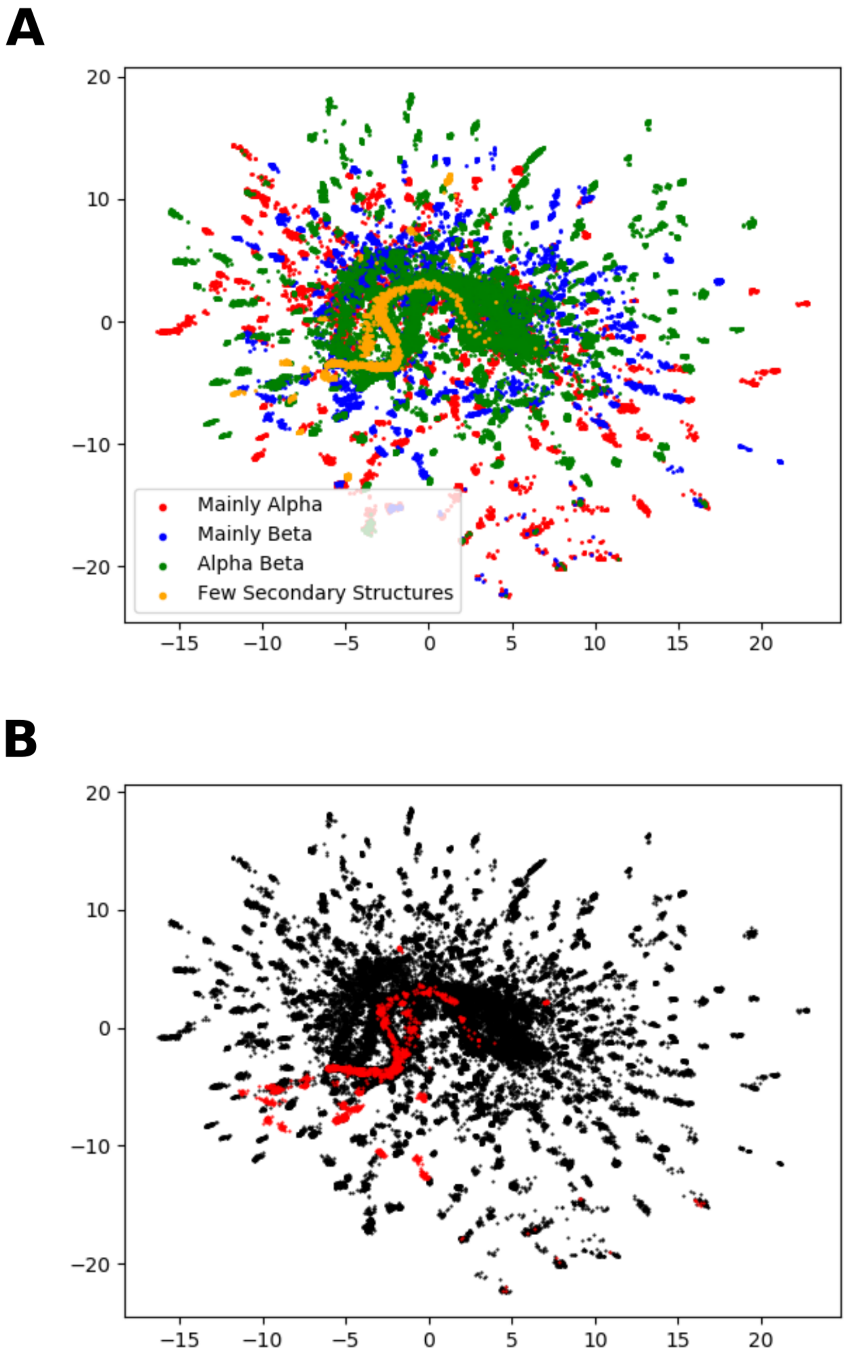


Figure 8. Separation by structural properties in the latent space when 2 latent dimensions are used in the model. The axes are the 2 latent dimensions and each point is the encoded representation in the 2 dimensions of one input sequence. Clusters generally correspond to the homologues collected for each sequence. (A) Each sequence is coloured by CATH class[32] according to the colours shown. (B) Sequences for one CATH architecture, 'mainly beta single sheet' (CATH ID 2.20), are highlighted in red.