Exploring protein sequence and functional spaces using adversarial autoencoder.

Tristan Bitard-Feildel
email: tristan@bitardfeildel.fr

November 9, 2020

Abstract

Shedding light on the relationship between protein sequences and their functions is a challenging task with implication in our understanding of protein evolution, diseases, or protein design. However, due to its complexity, protein sequence space is hard to comprehend with potential numerous human bias. Generative models able to learn and recreate the data specificity can help to decipher complex systems. Applied to protein sequences, they can help to pointing out relationships between protein positions and functions or to capture the different sequence patterns associated to functions. In this study, an unsupervised generative approach based on auto-encoder (AE) is proposed to generate and explore new protein sequences with respect to their functions. AE are tested on three protein families known for their multiple functions, of which one has manually curated annotations. Functional labelling of encoded sequences on a two dimensional latent space computed by AE for each family shows a good agreement regarding the ability of the latent space to capture the functional organization and specificity of the sequences. Furthermore, arithmetic between latent spaces and latent space interpolations between encoded sequences are tested as a way to generate new intermediate protein sequences sharing sequential and functional properties of sequences issued of families with different sequences and functions. Using structural homology modelling and assessment, it can be observed that the new protein sequences generated using latent space arithmetic display intermediate physico-chemical properties and energies relatively to the sequences of the families used to generate them. Finally, interpolated protein sequences between data points of the input data set show the ability of the AE to smoothly generalize and to produce meaningful biological sequences from un-charted area of the latent space. Code and data used for this study are freely available at https://github.com/T-B-F/aae4seq.

1 Introduction

Protein sequences diversity is the result of a long evolutionary process. This diversity, or sequence space, is constrained by natural selection and only a fraction of amino acid combinations out of all possible combinations are observed. Given its huge size, exploring the sequence space and understanding its hidden constrains is very challenging. Protein families are groups of related protein, or part of proteins, and represent a useful description to reduce the sequence space complexity. Many resources have been developed over the year to group protein sequences in families whose members share evidence of sequence similarity or structural similarity [8, 11, 22]. However, even a family is not without diversity and one family may group together several proteins with different molecular functions [5]. Navigating the sequence space with respect to the functional diversity of a family is therefore a difficult task whose complexity is even increased by the low number of proteins with experimentally confirmed function. In this regard, computer models are needed to explore the sequence space in relation to the functional space of the protein families.

Understanding the relationships between amino acids responsible of a particular molecular function has a lot of implication in molecular engineering, functional annotation and evolutionary biology. In this study, an unsupervised deep learning approach is proposed to model and generate protein sequences. The ability of this approach to capture the functional diversity and specificity of protein is evaluated on different protein families.

Variational autoencoders (VAE) have been applied on biological and chemical data to explore and classify gene expression in single-cell transcriptomics data [17], predict the impact of mutations on protein activity [26, 31] or to explore the chemical space of small molecules for drug discovery and design [14, 25] Their ability to reduce input data complexity in an latent space and performs inference on this reduced representation make them highly suitable to model, in an unsupervised manner, complex systems. In this study, Adversarial AutoEncoders (AAE) [19] are proposed as an application of a new and efficient ways to represent and navigate the functional space of a protein family.

Autoencoders are able, using the encoder, to reduce high dimensional data by projection to a lower dimensional space (also known as a latent space or hidden code representation) which in turn can be reconstructed by the decoder. AAE [19] architecture corresponds to a probabilistic autoencoder, but with a constraint on the hidden code representation of the encoder which follow a define prior distribution. This constraint is performed by a generative adversarial networks (GAN) [12] between the latent space and the prior distribution, and ensures that meaningful samples can be generated from anywhere in the latent space defined by the prior distribution. Therefore, applied to biological sequences of a protein family, it is possible to encode the sequence diversity to any prior distributions and to analyze the ability of the AAE to learn a representation with respect to the functions of the protein family considered. Furthermore, and contrary to dimensional reduction techniques such as PCA or t-SNE [18], AAE networks have the ability to perform inference on the latent space, meaning that any sampled point of this distribution can be decoded as as a protein sequence. Sampling of the latent space was therefore analysed, with a particular focus on latent space arithmetic between proteins of different sub-families with different functions to validate the ability of the model to learn a meaningful representation of the sequence space. Arithmetic operations on latent space have previously been reported to transfer features between images of different classes and may therefore have interesting potential for molecular design. Interpolations between points (encoded protein sequences) of the latent space were also performed to evaluate the ability of the AAE to navigate and generate new protein sequences from unseen data points.

To test these hypothesis, three different protein families were selectionned including one, the sulfatase family, for which the functions of some sub-family have been recently manually curated [2]. The sulfatases are a group of proteins acting on sulfated biomolecules. This group of proteins is found in various protein family databases, such as in Pfam (PF00884, Sulfatases). However, as mentioned previously, they can have different substrate specificity despite being in the same family. The SulfAtlas database [2] is a collection of curated structurally-related sulfatases centered on the classification of the substrate specificity. The majority of sulfatases (30,726 over 35,090 Version 1.1 September 2017) is found in family S1 and is sub-divised into 73 sub-families corresponding to different substrate specificity. Sub-families S1_0 to S1_12 possessed experimentally characterized EC identifier. The two other protein families, HUP and TPP families are not manually crated, but were selected as they are known to have multiple functions [5].

For each family an AAE network is trained on the protein sequence space and the functional diversity of the family in the AAE latent space is analyzed.

2 Results

A structurally constrained MSA was computed using Expresso [1] from T-Coffee webserver [9] between sequences of S1 sulfatases structures. This MSA was processed into a Hidden Markov Model and hmmsearch was used to retrieved aligned sequence matches were against the UniRef90 sequence database.

A total of 76,427 protein sequence hits were found matching UniRef90. The sequences were prepossessed to remove columns and hits with more than 90% and 75%, respectively, of gap characters. The final alignment comprised 41,901 sequences. The Sulfatases protein dataset was separated in a train, validation and test sets with a split ratio of: 0.8, 0.1, and 0.1.

A comparison of architectures for protein sequence family modelling is outside of the scope of this study whose focus is on the ability of the AAE architecture to model and infer protein sequences. Three different AAE architectures were trained (see Method section), but without extensive hyper-parameters optimization. The three architectures were trained on the train set and tested on the validation set. The test set was only evaluated on the final selected architecture. Table 1 shows the top k accuracy metric for k=1 and k=3 computed for the different autoencoders. The accuracy scores scaled down with the number of parameters, but without any large differences. To avoid over-fitting, the architecture with the fewest number of parameters (architecture 3) was therefore selected and the final accuracy scores on the test were of 62.5% and 80.2% (k=1 and k=3).

The selected architecture was subsequently independently trained on the protein sequences of the HUP and TPP families.

Table 1: Accuracy metrics (k=1 and k=3) on the train and validation sets of sulfatases using different models (see Method)

Architecture	Accuracy (%)		Top 3 accuracy (%)		Number of
	Train	Validation	Train	Validation	Parameters
1	67.3	66	83.9	82.3	11 198 553
2	65.1	64	82.4	81	9 769 593
3	63.3	62.4	81.2	80.1	8 073 337

2.1 Latent space projection

AAE can be used as a dimensional reduction techniques by fixing the dimension of the latent space to two or three dimensions for plotting and data exploration. The final MSA of the sulfatase family was therefore used to train an AAE using two latent dimension. For comparison, the MSA projection using the first two component of a PCA decomposition was also computed.

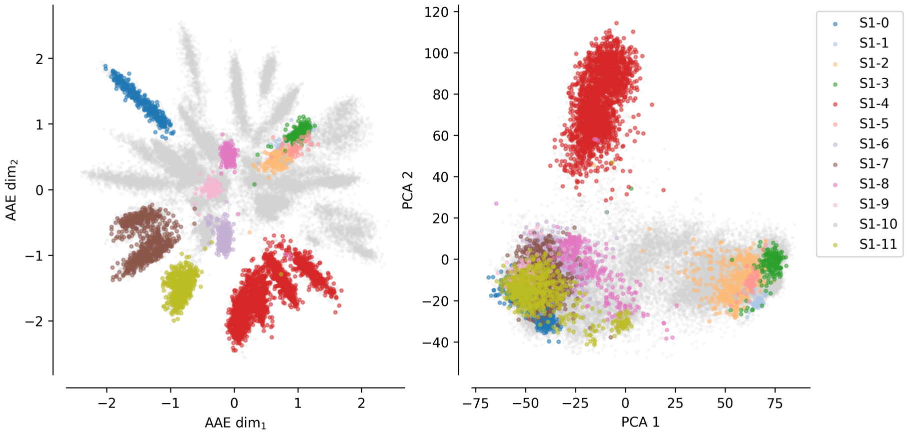

Figure 1: Sequences of the SulfAtlas MSA projected using the encoding learned sing an AAE (number of latent dimension: 2) and a PCA (two first component). Grey data points correspond to protein sequences not found in the first 12 sub-families.

Figure 1 shows the protein sequences encoded by the AAE in two latent dimension and the PCA projections. Each dot correspond to a protein sequence, and the dot are colored according to their sub-family. Grey dot corresponds to protein sequences not belonging to any of the 12 curated sulfatases sub-families. In this figure, the AAE displays a clear superior power to disentangle the sequence and functional spaces of the S1 family than a PCA. Interestingly, it can be observed in the AAE projection some separated grey spikes (sequences not belonging to any curated sub-family). These spikes may correspond to groups of enzymes sharing common substrate specificity.

For some cases, the the sub-family with identical functions are projected closely on the encoded space. For instance, sub-families S1_6 (light magenta) and S1_11 (yellow) have both the activity EC 3.1.6.14 (N-acetylglucosamine-6-sulfatase) and are close in the encoded space. Interestingly some sub-family projections appear entangled such as the S1-1 sub-family (light blue, Cerebroside sulfatase activity EC 3.1.6.8), the S1-2 (orange) and the S1-3 (green) subfamilies (Steryl-sulfatase activity, EC 3.1.6.2) the S1-5 (pink) sub-family (N-acetylgalactosamine-6-sulfatase activity, EC 3.1.6.4), and the S1-10 (grey) subfamily (Glucosinolate sulfatase activity EC 3.1.6.-). The five families correspond to four different functions, but are made of Eukaryotic protein sequences only and their entanglement may be due to their shared common evolutionary history. This separation based on the sequence kingdoms can clearly be visualized in the PCA projections with Eukaryotic sequences on the right side on subfamilies with a majority of Bacteria sequences on the left side. The example of the protein B6QLZ0_PENMQ is also interesting. The protein is classified in the SulfAtlas database as part of the S1-4 and S1-11 sub-families but projected (yellow dot at coordinates (0.733, -1.289)) inside the space of the S1-4 family (red). Similar bi-classification can also be found for proteins between of the S1-4 and S1-8 sub-families: F2AQN8 (coordinates (0.795, -0.982)), M2AXU0 (coordinates (0.858, -0.994)), and Q7UVX7 (coordinates (0.883, -1.052)).

Projection of sequence spaces using AAE with 2 latent dimensions were also tested on the HUP and TPP families. The AAE projections can be visualized on Figure 2. There is fewer functional annotations for these two families but it can clearly be seen a strong separation between the major functions of the families.

HUP points colored in yellow correspond to protein with EC 6.1.1.1 and 6.1.1.2, pink colored points to proteins with EC 6.1.1.1 and violet colored points to proteins with EC 2.7.11.24 and 6.1.1.2. TPP sequences have more annotated functions than HUP sequences (57 different EC assignation), but a global pattern can be found in the projection corresponding to two groups of proteins (brown and violet) annotated with EC 2.2.1.1 (Transketolase), oxidoreductase proteins (EC 1.-.-.-, in orange, pink, red, and green shades), and proteins with function EC 2.2.1.9 (2-succinyl-5-enolpyruvyl-6-hydroxy-3-cyclohexene-1-carboxylic-acid synthase, in yellow and grey shades).

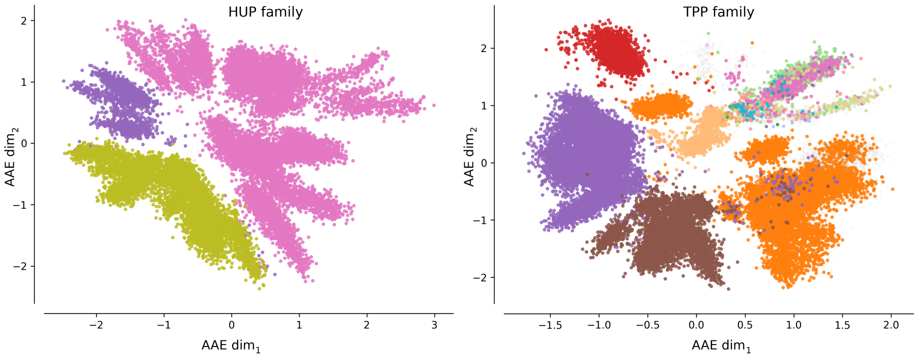

Figure 2: Encoded sequences of HUP (left) and TPP (right) multiple sequences alignments using AAEs with 2 latent dimensions. Data points are colored according to their enzyme classification annotation retrieved from GOA.

Latent spaces were evaluated for each protein family based on enzyme classification (EC) and taxonomic specificities. Given a set of protein sequences, the encoded sequences in latent space of dimension 100 were clustered using HDBSCAN and the clusters were evaluated according to the enzyme class or taxonomic group with the highest propensity inside a cluster.

For the sulfatase family 27 clusters were found for which taxonomic and EC annotations could be extracted (Supplementary Table 3). All these clusters displayed either strong taxonomic or EC specificities. Enzymatic specificity was found stronger than taxonomic specificity for 16 clusters, found equal in one cluster and lower for 10 clusters.

In the HUP family, all clusters have very high EC specificity (Supplementary table 4). Only two clusters out of 47 could be found with higher taxonomic specificity than EC specificity and for this two clusters enzymatic specificity values were high and only marginally different (cluster 5, taxonomic specificity of 100% an EC specificity of 99% and cluster 31, taxonomic specificity of 99 % and EC specificity of 97%). Five clusters were found with equal taxonomic and EC specificities.

Similarly, in the TPP family, all clusters have also very high EC specificity (Supplementary table 5). Five clusters out of 51 could be found with higher taxonomic specificity than EC specificity. For these 5 clusters the differences between taxonomic specificity and EC specificity is higher than the differences observed for the HUP clusters. Six clusters were found with equal taxonomic and EC specificity.

These results highlight the ability of the encoding space to capture both enzymatic specificity and taxonomic features.

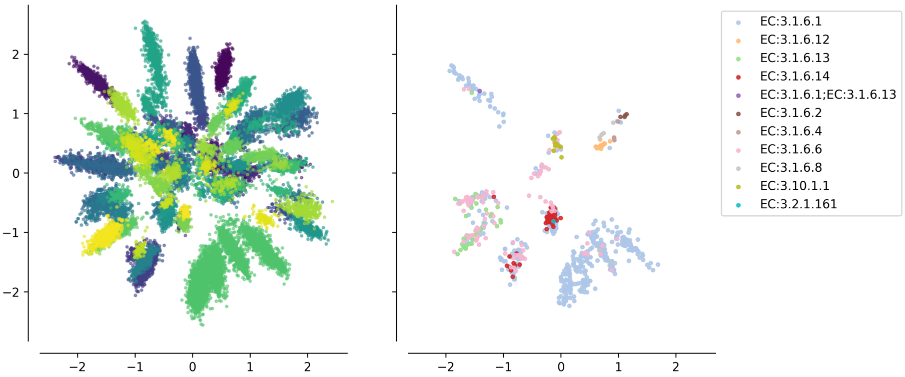

Figure 3: Projection of SulfAtlas MSA encoded sequences using an AAE (number of latent dimension: 2) and colored according to the retrieved GOA Enzyme Classification annotation (left) or computed clusters using HDBSCAN on the encoded sequences using an AAE with 100 latent dimensions (right).

2.2 Protein latent space arithmetic

It has been shown that latent space arithmetic was able to transfer learned features between different classes. This ability is interesting to test in the case of protein sequences has it may allow to explore protein families by transferring features of a first sub-family to a second sub-family while maintaining its the

general property (such as its structure).
To test this hypothesis, Sulfatases sub-families with at least 100 labelled members but with less than 1000 members (to avoid pronounced imbalance between classes) were selected: S1-0 (308 proteins), S1-2 (462 proteins), S1-3 (186 proteins), S1-8 (290 proteins), S1-11 (669 proteins). The only sub-family with more than 1000 members is the S1-4 sub-family (2064 proteins).
Different arithmetic strategies (see Methods and Figure 4) were tested between latent spaces of a query sub-family and a different source sub-family with the aim to transfer features of the source sub-family to the query sub-family.

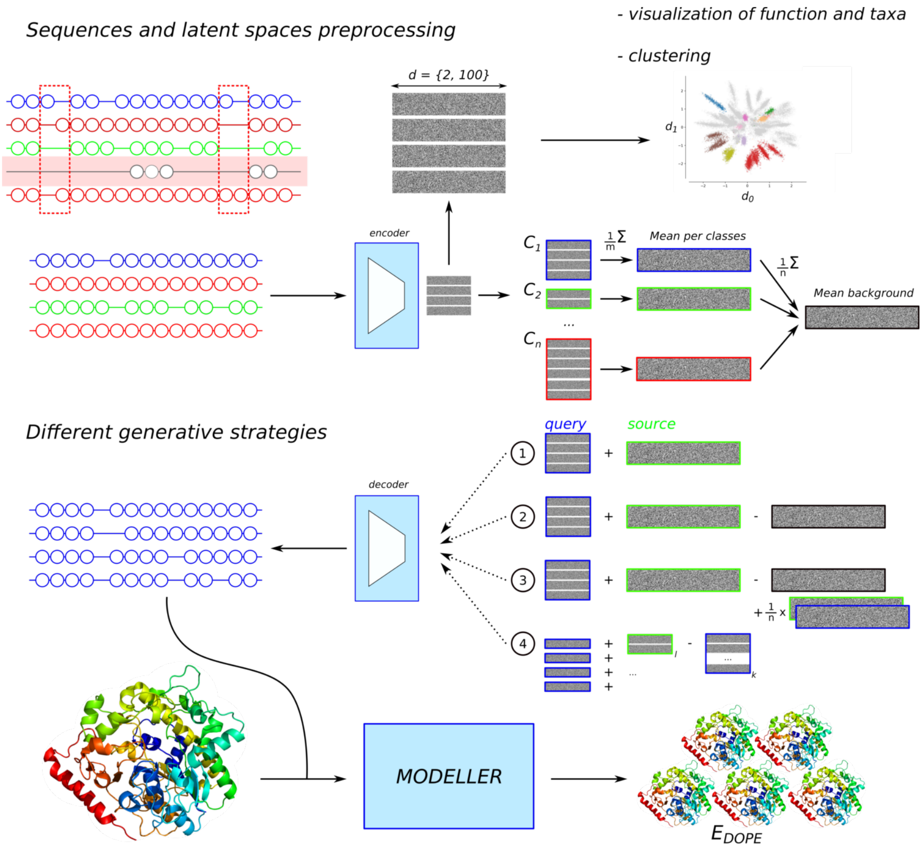

Figure 4: Modelling pipeline used to generate sequences sharing properties of two sub-families. The hmmsearch MSA is filtered and passed to the encoder to project each sequence to the latent space. Latent space projections can be used for visualization (see Figure 3). Different strategies (1 to 4) are tested to generate new latent space and generate new sequences through the decoder. The new sequences are used in combination with structures of the sub-families to create homology based structural models and evaluated using the DOPE energy function of MODELLER.

Figure 5 displays logo plots of two regions corresponding to Prosite motifs PS00523 and PS00149 to illustrate the amino acid content of the protein sequences generated by latent space arithmetic (Supplementary data Information for the full logo plots). The regions correspond to the most conserved regions of the sulfatase family and have been proposed as signature pattern for all the sultatses in the Prosite database. Panels A and D correspond to the sequences of the S1-0 sub-family and of the S1-2 sub-family respectively. Panels B and C correspond to generated protein sequences using either the Sulfatase sub-family S1-0 as source and S1-2 as query (Panel B) and to which the background latent space has been subtracted (strategy 2 on Figure 4) and reciprocally (Panel C).

Different amino acid patterns can be observed between the different motifs of the sequence groups. In the first fragment corresponding to motif PS00523, G55 and T55 are the most frequent amino acid of sub-families S1-0 and S1-2 (Panels A and D) and it can be observed a "competition" between these two amino acids for generated sequences (Panel B and C) with a slightly higher probability for the amino acid of the family used as query (G in Panel B and T in panel C). The residue S57, implicated in the active site of the sulfatases [3], is less frequent in the query sub-family S1-2 (panel D) than in sub-family S1-0 (panel A). The high frequency of S at position 57 in sub-family S1-0 compared to sub-family S1-2 may have an impact when performing the latent space arithmetic as S is predominant in the generated sequences. This influence of a very frequent amino acid in one of the source or query sub-family on the generated sequences can also be observed at position 70 and is less visible when multiple amino acid frequencies are more balanced. In the second fragment corresponding to motif PS00149, residue R at position 101 follows this pattern. It is highly frequent in sub-family S1-0 (panel A) and less frequent in sub-family S1-2 (panel D). The generated protein sequences display a R at position 101 with high frequency. The inverse can be observed for Y at position 105, highly frequent in sub-family S1-2 (panel D).

Other positions are however displaying much more complex pattern and cannot be summarize as a frequency competition between source and query sub-families For instance, G at position 71 is very frequent in sub-family S1-2 but have a comparable frequency with R in sub-family S1-0. The generated protein sequences don't display G has the only possible residue but seem to follow the frequency of their respective query sub-families. Amino acids at positions in generated sequences where the multiple amino acid share comparable frequencies in the source and query sub-families, such as in positions 53, 59, 67, 98, or 106, have usually also mixed frequencies.

These behaviours can be observed several times through the logo plots but are still positions specifics, meaning that the bits scores pattern observed in the source sub-families (Panels A and D) do not necessary allow to predict the amino acids bits scores in the generated sub-families (Panels B and C). For instance, W at positions 113 as a high bit score value in the MSA of sub-family S1-2 but little influences in the amino acids of the generated sequences where the T of sub-family S1-0 is found predominantly. Moreover, these observations are performed on residues of the Prosite motifs which are by definition conserved in sulfatases. In other positions, the patterns are harder to explains (see Supplementary Figure XX).

Furthermore, protein structural modelling was performed to assess and compare the properties of the sequences generated by latent space arithmetic and the protein sequences of the natural sub-families. For each sub-family, 100 original sequences were randomly selected along the corresponding generated sequences.

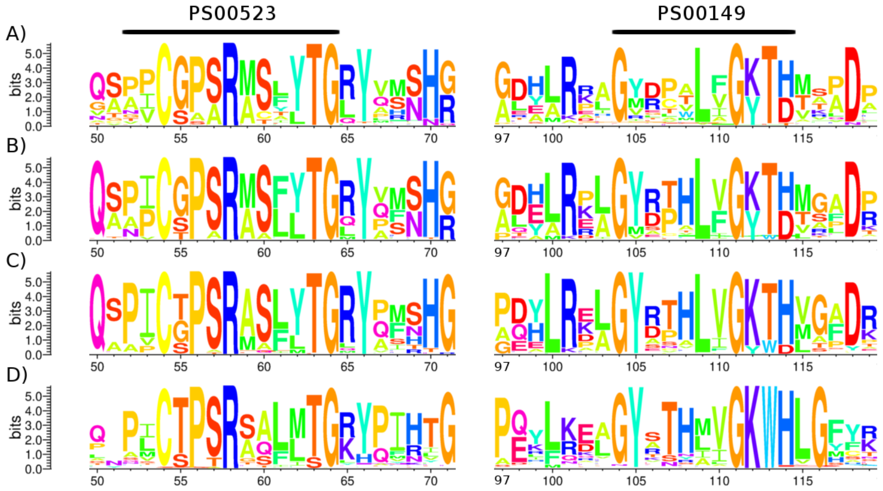

Figure 5: **Logo plot of MSAs parts from the S1-0 and S1-2 (panels A and D) sub-families, and generated sequences using S1-0 as query and S1-2 as source (panel B) and S1-2 as query and S1-0 as source (panel C)**.

All the generated sequences were aligned to protein structures of their corresponding source and query sub-families, and the alignments were used to create structural models by homology. The models were evaluated with the DOPE function of MODELLER.

Figure 6 shows an example of the energy distribution computed from models using the second strategy with query sub-family S1-0 and source sub-family S1-2.

The lowest energies (best models) are found when modelling the original protein sequences of a sub-family to the structures of the same sub-family (*Struct. 0 Seq. 0* and *Struct. 2 Seq. 2*). Inversely, the highest energies are found when modelling the original protein sequences of a sub-family to the structures of another sub-family (*Struct. 0 Seq. 2* and *Struct. 2 Seq. 0*). Interestingly, sequences generated using addition and subtractions of latent space have intermediate energy distributions. This can be clearly observed in Figure 7, where generated sequences are mostly between the two dotted lines representing the energies of original protein sequences modelled on their corresponding sub-family structures (vertical line at 0) and the energies of original protein sequences modelled on structures of another sub-family (top left diagonal line). Generated sequences modelled on structures belonging to the same sub-family than their query latent space sub-family (ex: *Struct. 0 Seq.S1-0m2* and *Struct. 2 Seq. S1-2m0* on Figure 6 and M_{QS}/Q on Figure 7) have slightly lower energy than when modelled on structures corresponding to the sub-family of their source latent space sub-family (ex: *Struct. 0 Seq. S1-2m0* and *Struct. 2 Seq. S1-0m2* on Figure 6 and M_{SQ}/Q on Figure 7). This trend is true for all query / source pairs of sub-families and all strategies except for sequences generated using the fourth strategy (local background subtraction of query latent space using a KD-tree and addition of source latent space), see Supplementary Figures 11, 12, 13 and Methods.

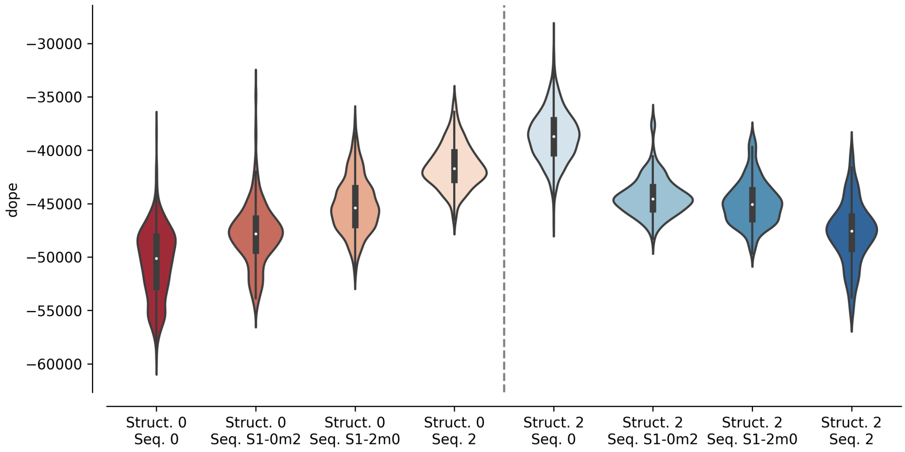

Figure 6: Energies distribution of models computed using structures from sub-family S1-0 (reds) or sub-family S1-2 (blues) and sequences from biological proteins or inferred using latent space arithmetic between spaces encoded by the S1-0 and S1-2 sub-families. Each violin plot corresponds to a specific targeted structures and sequences couples. For example, *Struct. 0 Seq. 0* indicates that the energy distribution corresponds to sequences of the S1-0 sub-family modelled on structures of the S1-0 sub-families and *Struct. 2 Seq. S1-0m2* corresponds to the energy distribution of sequences inferred using the latent space of sub-family S1-2 added to the latent space of sub-family S1-0 and modelled on structures of the S1-2 sub-family.

In this strategy, the generated sequences do not display energy distributions in-between the energy distributions of the original sequences modelled on structures of the query or the source sub-families. The energy distributions of generated sequences using the fourth strategy are closer to the energy distribution of their corresponding original query sequences. No clear differences could be observed between the first, second and third strategy.

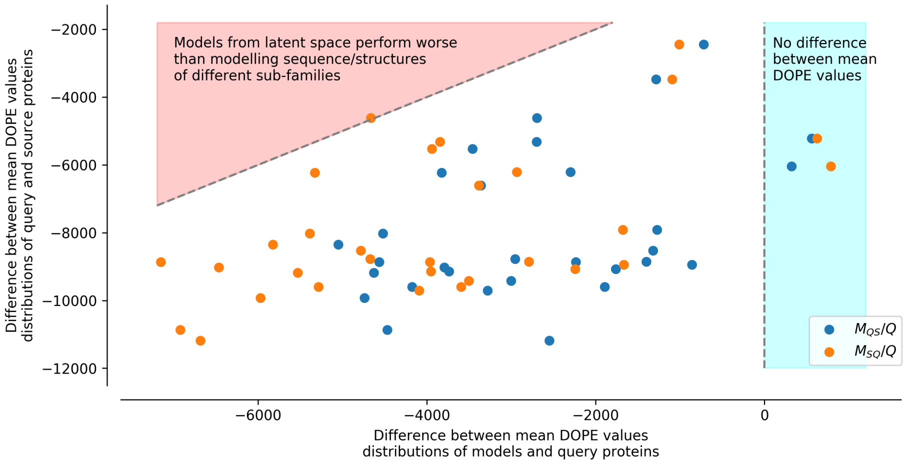

Figure 7: Differences between mean DOPE distributions. Mean value for each distribution, such as the distributions presented in Figure 6, were computed. The y axis represent the differences between the mean values computed for query sequences modelled on structures of the same sub-family and mean values computed for source sequences modelled on structures of the query sub-family (ex: differences between mean of Struct. 0 Seq. 0 and mean of Struct. 0 Seq. 2 distributions in Figure 6). The x axis corresponds ot the differences between the mean values computed for query sequences modelled on structures of the same sub-family and mean values computed for query sequences to which latent space of the source sub-family sequences have been added and modelled on structures of the query sub-family (M_{QS}/Q), or source sequences to which latent space of the query sub-family sequences have been added and modelled on structures of the source sub-family (M_{SQ}/Q) (ex: differences between mean of Struct. 0 Seq. S1-0m2 and mean of Struct. 0 Seq. 0 distributions in Figure 6). Points in the red area correspond to mean distribution values from generated sequences whose modelled structures have a higher energy than models created using pairs of sequences/structures from different sub-families. Points in the blue are correspond to mean distribution values from generated sequences whose modelled structures have a lower energy than models created using pairs of sequences/structures from the same sub-family.

2.3 Protein latent space interpolation

Interpolation between encoded sequences in the latent space can be used to "navigate" between proteins of two sub-families. Applied in computer vision, interpolation has proven its capacity to generate meaningful intermediate representations between different input images. In this study, ten pairs of sequences from sub-families S1-0 and S1-4 (respectively blue and red data points in figure 1) were selected to test the capacity of the AAE in this task, and 50 intermediates data points were generated between each pair.

The resulting sequences can be found in the Supplementary Data. Several interesting point can be observed. First, when gaps are found in the query sequence but not in the target sequence (and inversely), the gaped area is progressively filled (or break down) starting from the flanking amino acids to the center (or inversely from the center to the flanking amino acids). This indicates an organized and progressive accumulation of amino acids (or gaps) to extend (or shrink) the region of the sequence previously without residues. For instance gap reduction can be observed in the generated sequences between sequence ID 2 of the sulfatase S1-0 family (query) and sequence ID 2196 of the sulfatase S1-4 family (target) at positions 75 to 86 (Figure 8), and can be found in all the other generated intermediates. Second, amino acids specific to a family are progressively replaced in key positions. For instance, in the interpolation between the same query and target sequences, it can also be observed at positions 21 and 22 of the MSA a replacement of residues S and C by G and A (Figure 8).

Most transitions are not abrupt and do not occurs at the 25th generated intermediate sequences but are smooth and correspond to plausible sequences. An abrupt transition can be observed at position 53, G (S1-0 query) to S (S1-4 target), and 51 [VI] (S1-0 query) to T (S1-4 target), corresponding to very conserved residues in Prosite motif PS00523 (see Figure 5. The other positions of the motif are a less affected by abrupt transition but appear to be less fluctuating than other columns. A similar behaviour can only be observed for columns 111, T (S1-0 query) to W (S1-4 target), of motif PS00149 (positions 102 to 112). The other positions of the motifs are either very conserved (T105, G109, K110) or accepting more fluctuations (columns 103, 104, 107).

The ability of the AAE to generate interpolated sequences with emerging or disappearing features of two sub-families, reflects its capacity to generalize the decoding to points not corresponding to encoded sequences and thus not previously observed, and point out to a structured organization of the computed latent space.

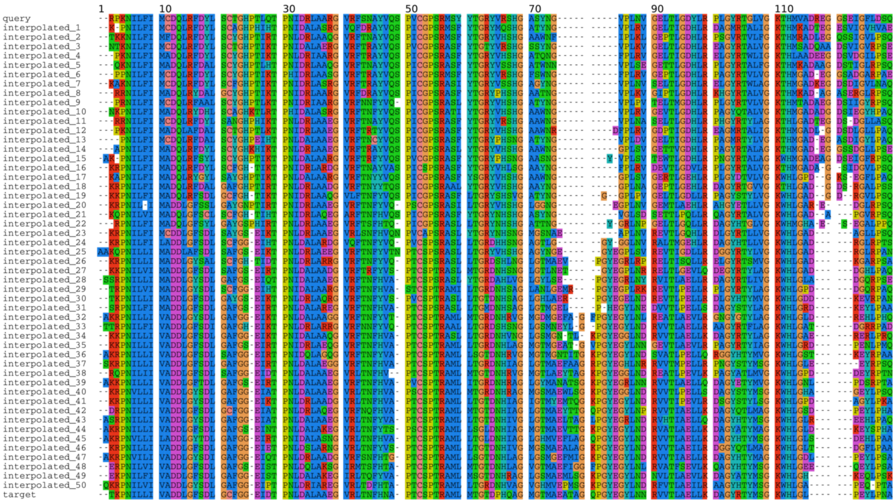

Figure 8: **First 130 amino acids fo sequences generated using interpolated latent space.** Interpolation is performed between latent spaces of protein ID 2 of the sulfatase S1-0 family (query) and of protein ID 2196 of the sulfatase S1-4 sub-family. Amino acids color coding is based on physo-chemical properties. Large transitions between gaped to amino acids and amino acids to gaps can be observed at positions 75 to 86 and positions 116 to 122. Amino acid columns transformation can be observed at multiple positions: 21 (S to G), 51 (V/I to T/S), 53 (G to S) etc.

3 Discussion

In this study, a new framework is presented to analyse and explore the protein sequence space regarding functionality and evolution. Some previous attempt, such as the FunFam database [6, 7], were built upon CATH protein families and trained in a supervized manner to construct model specific of a functionality. Variational Autoencoder (VAE) have previously been reported and used to disentangle complex biological information and used for classification tasks (such as single cell gene expression data) [23, 34] or for generation of new molecules (such as drug) [14, 24, 25, 27, 28]. AAE are able to disentangle the information contained in protein sequences to capture functional and evolutionary specific features and more importantly without supervision. They also have the advantage over VAE to constrain the latent space over a prior distribution which allow sampling strategies to explore the whole latent distribution. It can also be noted that Restricted Boltzmann Machine [33] have also been recently proposed in a similar task showing very promising results despite the difficulty to train RBM which required markov chain sampling.

AAE are trained on protein sequence families known to have different subfunctions. The results present the ability of AAEs to separate sequences according to functional and taxonomic properties for the three studied family. This point out to the ability of the AAEs to extracted and encoded features in the latent space which are biological relevant.
Furthermore, and contrary to dimensional reduction techniques, AAE can be used to generate new protein sequences. Latent space arithmetic have been used in image generation tasks with striking ability to produce relevant images with intermediate features. Latent space arithmetic is an interesting concept for protein generation particularly in the context of extracting and transferring features between protein families. Three out of the four different experiments carried on where able to generate sequences with intermediate features as measured from their sequence identity distributions and modelling energy assessment. Biological experiments will be needed to confirm the functional relevance of the transferred features, but the strategies could have many application should it be validated.

The absence of measured differences between three out of four strategies used to generate intermediate sequences may also indicates that a more optimal approaches could be design. In this regard, the model architecture could also be improved. Currently, the model used as input a filtered MSA, but improved architectures could probably benefit from full protein sequences of different sizes without filtering. It is for instance known that motifs in the TPP and HUP family plays important roles in the family sub-functions **??**. As protein specific motifs, they are not necessary conserved and may not reach filtering thresholds. Recent advances have been made regarding the application of GAN to text generation [13, 37, 38] and transferring these progresses to the field of protein sequence generation could greatly benefit the design of functionally relevant proteins.

The results of this study show that AAE, in particular, and deep learning generative models in general, can provide original solutions for protein design and functional exploration.

4 Methods

4.1 Data

4.1.1 The sulfatase family

A initial seed protein multiple sequence alignment was computed from sequences of the protein structures of SulfAtlas [2] database sub-families 1 to 12. This seed was used to search for homologous sequence on the UniRef90 [32] protein sequence database using hmmsearch [10] and with reporting and inclusion e-values set at $1e-3$.
A label was assigned to each retrieved protein if the protein belongs to on of 12 sub-families. The MSA computed by hmmsearch was filtered to remove columns and sequences with more than 90% and 75% gap character respectively. Proteins with multiple hits on different part of their protein sequence were also merged into a single entry. From 105181 initial protein sequences retrieved by hmmsearch, the filtering steps lead to a final set of 41901 proteins.

4.1.2 HUP and TPP protein families

A similar protocol was followed for the HUP and TPP protein families. Instead of using a initial seed alignment made of sequence of protein structure, the CATH protein domain HMM [21, 30] was used to search homologous sequences in the UniRef90 database. CATH model 3.40.50.620 corresponds to the HUP protein family and model 3.40.50.970 corresponds to the TPP protein family. A sequence filtering pipeline identical than the one used for the sulfatase family was applied to each of the resulting multiple sequence alignment.
The final number of protein in each dataset was: 25041 for the HUP family (32590 proteins before filtering) and 33693 for the TPP family (133701 before filtering).

4.2 Model

4.2.1 Generative Adversarial Network

A complete description of Generative Adversarial Network can be found in Goodfellow et al. [12]. To summarize, the GAN framework correspond to a min-max adversarial game between two neural networks: a generator (G) and a discriminator (D). The discriminator computes the probability that a input x corresponds to a real point in the data space rather than sample from the generator. Concurrently, the generator maps samples z from prior $p(z)$ to the data space with the objective to confuse the discriminator. This game between generator and discriminator can be expressed as :

$$min_G \; max_D \; E_{x \sim p_{data}}[\log D(x)] + E_{z \sim p(z)}[log(1 - D(G(z))] \tag{1}$$

4.2.2 Adversarial auto-encoder

Adversarial autoencoders were introduced by Makhzani et al. [19]. The proposed model is constructed using an encoder and decoder networks, and a GAN to match the posterior distribution of the encoded vector with an arbitrary prior distribution. Such, the decoder of the AAE learned from the full space of the prior distribution. The model used in this study compute the aggregated posterior $q(z|x)$ (the encoding distribution) using a Gaussian prior distribution. The mean and variance of this distribution is predicted by the encoder network: $z_i \sim N(\mu_i(x), \sigma_i(x))$. The re-parameterization trick introduced by Kingma and Welling [16] is used for back-propagation through the encoder network.

Network architecture. Three different architectures were evaluated. The general architecture is as follow and Table 2 provided an overview of the differences between architectures. A representation of architecture number 3 can be found on Supplementary Figure 9. The encoder comprises one or two 1D convolutional layers with 32 filters of size 7 and with stride length of 2, and one or two densely connected layers of 256 or 512 units. The output of the last layer is passed to two different densely connected layers of hidden code size units to evaluate μ and σ of the re-parameterization trick [16].
The decoder is made of two or three densely connected layers of length of the sequence family time alphabet units for the last layers and of 256 or 512 units for the first or the two first layers. The final output of the decoder is reshaped and a softmax activation function is applied which corresponds to a probability for every positions associated to each possible amino acids. To convert the probability matrix of the decoder to a sequence, a random sampling according to the probability output was performed at each position. The selected amino acid at a given position is therefore not necessary the amino acid with the highest probability. The sampling was also performed using a temperature factor T of 0.5 to scale the probability values as follow:

$$P_i = \frac{P_i^{1/T}}{\sum_j^N P_{ij}^{1/T}}$$

with P_i the probability vector at position i of length N which corresponds to the number of amino acids considered for decoding, and T the temperature factor.

The discriminator network is made of two or three densely connected layers, the last layers has only one unit and corresponds to the discriminator classification decision through a sigmoid activation function, the first or the first two layers are made of 256 or 512 units.

Table 2: Differences between layers of the evaluated model architectures

Architecture	1	2	3
Encoder	Conv 1D (32, 7) Dense (512)	2 x Conv 1D (32, 7) 2 x Dense (512)	2 x Conv 1D (32, 7) 2 x Dense (256
Decoder	Dense (512)	2 x Dense (512)	2 x Dense (256)
Discriminator	Dense (512)	2 x Dense (512)	2 x Dense (256)

Training. The network was trained for each of the protein family independently. The autoencoder is trained using a categorical cross-entropy loss function between the input data and the predicted sequences by the autoencoder. The discriminator is trained using binary cross-entropy loss function between the input data encoded and the samples from the prior distribution.

4.3 Analyses

4.3.1 Dimensionality reduction

The AAE model can be used to reduce the dimensionality of the sequence space by setting a small latent size. Two dimensionality reductions were tested with latent size of 2 and 100. Latent size of 2 can be easily visualized and a larger latent size of 100 should represent the input data more efficiently as more information can be stored.

4.3.2 Clustering

HDBSCAN [4, 20] was used to cluster the sequences in the encoded space due to its capacity to handle clusters of different size and density and its performances in high dimensionality. The Euclidean distance metric was used to compute distances between points of the latent space. A minimal cluster size of 60 was set to consider a group as a cluster as the number of protein sequences is rather large. The minimal number of samples in a neighborhood to consider a point as a core point was set to 15 to maintain relatively conservative clusters.

4.3.3 Functional and taxonomic analyses

Enzyme functional annotation (EC ids) and NCBI taxonomic identifiers were extracted when available from the Gene Ontology Annotation portal (Januaray 2019) using the UniProt-GOA mapping [15]. Protein without annotation were not taken into account in these analyses.

The annotation homogeneity for each computed cluster. Considering a cluster, the number of different EC ids and taxonomic ids were retrieved. For each different EC ids (taxonomic ids) its percentage in the cluster was computed. A EC id (taxonomic id) of a cluster with a value of 90% indicates that 90% of the cluster member have this EC id (taxonomic id). A cluster with high values correspond to functionally or evolutionary related sequences.

Homogeneous clusters computed from the AAE encoded space will therefore indicates the ability of the AAE model to capture and to distinguish protein sequences with functionally or evolutionary relevant features without supervision.

4.3.4 Latent space arithmetic

Subtraction and addition of latent spaces have been shown to be able to transfer features specific to some sub-group of data to other sub-group of data (ex.: men women with and without glasses). This property is tested in the context of protein sub-families. Different arithmetic strategies (Figure 4) were tested between latent spaces of a query sub-family and a different source sub-family with the aim to transfer features of the source sub-family to the query sub-family.

A fist strategy consists to add the mean latent space, computed using the encoder on the sequences of the source sub-family, to the encoded sequences of the query sub-family. In the second tested strategy differs from the first one by subtracting the mean background latent space, computed from the latent space of all sub-families, from the latent space of the query sub-family. The third strategy differs to the second as the background strategy is computed using all sub-families except sub-families source and query. Finally, in the fourth strategy, the subtraction is performed using a local KD-tree to only remove features shared by closest members of a given query and addition is performed randomly selecting a member of the the source family and it's closest 10 members.
For each strategy, new sequences were generated using the latent spaces of all query proteins in the sub-families. Thus, for one original encoded protein sequences there is a direct corresponding with the original amino acid sequence and the amino acid sequences generated with the different strategies and different source sub-families. The generate sequences by latent space arithmetic are compared to the initial sub-families in term of sequence and structural constrains.

To evaluate the generated sequences by latent space arithmetic, the sequences are compared to the biological sequences of the two initial sub-families using the hamming metric excluding positions corresponding to two gaps in the sequences. The hamming distances between sub-families are also computed. The distributions of hamming distances allow to explore the ability of the latent space arithmetic operations and of the decoder to produce meaningful intermediate protein sequences from unexplored encoded data points.

Furthermore, protein structural models are computed using the structures of the initial sub-families as template for MODELLER [35] and evaluated using the DOPE energy [29]. Models are computed using the generated sequences by latent space arithmetic on structure of their source and query sub-families. Their DOPE energy are also compared to models computed using the sequences of their source sub-family modelled on structures of their query sub-family and using sequences of their query sub-family modelled on structures of their source sub-family. If the generated sequences by latent space arithmetic correspond to intermediate proteins with properties from two sub-families they should have intermediate DOPE energy compared to computed structures using source sequences on source structures (or query sequences on query structures) and source sequences on query structures (or query sequences on source structures).

4.3.5 Latent space interpolation

Ten pairs of protein sequences were randomly chosen between sub-families S1-0 and S1-4. The two sub-families were chosen based on their opposite positions in the projection performed with the AAE using two dimensions (see Figure 1). The coordinates of the selected sequences in the encoded space with 100 dimensions were retrieved and spherical interpolation using 50 steps were performed for each of the pairs. Spherical interpolation has previously been reported to provide better interpolation for the generation of images [36]. A linear interpolation was also tested but no clear differences could be observed. The interpolated points were given to the decoder and new sequences were generated according to the procedure previously describe.

Scripts and notebooks are available for reproducibility at https://github.com/T-B-F/aae4seq.

5 Acknowledgement

TBF thanks the NVIDA society for providing a TitanXp GPU to perform computations.

References

[1] F. Armougom, S. Moretti, O. Poirot, S. Audic, P. Dumas, B. Schaeli, V. Keduas, and C. Notredame. Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. *Nucleic Acids Res.*, 34(Web Server issue):W604–608, Jul 2006.

[2] T. Barbeyron, L. Brillet-Gueguen, W. Carre, C. Carriere, C. Caron, M. Czjzek, M. Hoebeke, and G. Michel. Matching the Diversity of Sulfated Biomolecules: Creation of a Classification Database for Sulfatases Reflecting Their Substrate Specificity. *PLoS ONE*, 11(10):e0164846, 2016.

[3] Charles S Bond, Peter R Clements, Samantha J Ashby, Charles A Collyer, Stephen J Harrop, John J Hopwood, and J Mitchell Guss. Structure of a human lysosomal sulfatase. *Structure*, 5(2):277–289, 1997.

[4] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[5] S. Das, N. L. Dawson, and C. A. Orengo. Diversity in protein domain superfamilies. *Curr. Opin. Genet. Dev.*, 35:40–49, Dec 2015.

[6] Sayoni Das and Christine A Orengo. Protein function annotation using protein domain family resources. *Methods*, 93:24–34, 2016.

[7] Sayoni Das, Ian Sillitoe, David Lee, Jonathan G Lees, Natalie L Dawson, John Ward, and Christine A Orengo. Cath funfhmmer web server: protein functional annotations using functional family assignments. *Nucleic acids research*, 43(W1):W148–W153, 2015.

[8] Natalie L Dawson, Tony E Lewis, Sayoni Das, Jonathan G Lees, David Lee, Paul Ashford, Christine A Orengo, and Ian Sillitoe. Cath: an expanded resource to predict protein function through structure and sequence. *Nucleic acids research*, 45(D1):D289–D295, 2016.

[9] P. Di Tommaso, S. Moretti, I. Xenarios, M. Orobitg, A. Montanyola, J. M. Chang, J. F. Taly, and C. Notredame. T-Coffee: a web server for the multiple sequence alignment of protein and RNA sequences using structural

information and homology extension. *Nucleic Acids Res.*, 39(Web Server issue):W13–17, Jul 2011.

[10] Sean R Eddy. Accelerated profile hmm searches. *PLoS computational biology*, 7(10):e1002195, 2011.

[11] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, et al. The pfam protein families database in 2019. *Nucleic acids research*, 47(D1):D427–D432, 2018.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. pages 2672–2680, 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[13] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long Text Generation via Adversarial Training with Leaked Information. *arXiv e-prints*, art. arXiv:1709.08624, Sep 2017.

[14] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572. URL https://doi.org/10.1021/acscentsci.7b00572.

[15] Rachael P Huntley, Tony Sawford, Prudence Mutowo-Meullenet, Aleksandra Shypitsyna, Carlos Bonilla, Maria J Martin, and Claire O'donovan. The goa database: gene ontology annotation updates for 2015. *Nucleic acids research*, 43(D1):D1057–D1063, 2014.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Oral presentation at the International Conference on Learning Representations, Banff, Alberta, Canada*, 14–16 April 2014.

[17] Romain Lopez, Jeffrey Regier, Michael B Cole, Michael I Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053, 2018.

[18] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[19] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. *CoRR*, abs/1511.05644, 2015. URL http://arxiv.org/abs/1511.05644.

[20] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*, pages 33–42. IEEE, 2017.

[21] Christine A Orengo, AD Michie, S Jones, David T Jones, MB Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[22] Arun Prasad Pandurangan, Jonathan Stahlhacke, Matt E Oates, Ben Smithers, and Julian Gough. The superfamily 2.0 database: a significant proteome update and a new webserver. *Nucleic acids research*, 47(D1): D490–D494, 2018.

[23] E. Pierson and C. Yau. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.*, 16:241, Nov 2015.

[24] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7), 2018. doi: 10.1126/sciadv.aap7885. URL http://advances.sciencemag.org/content/4/7/eaap7885.

[25] Ladislav Rampasek, Daniel Hidru, Petr Smirnov, Benjamin Haibe-Kains, and Anna Goldenberg. Dr.VAE: Drug Response Variational Autoencoder. *arXiv e-prints*, art. arXiv:1706.08203, Jun 2017.

[26] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods*, 15:816–822, 2018.

[27] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018. ISSN 0036-8075. doi: 10.1126/science.aat2663. URL http://science.sciencemag.org/content/361/6400/360.

[28] Benjamin Sanchez-Lengeling, Carlos Outeiral, Gabriel L. Guimaraes, and Alán Aspuru-Guzik. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). 8 2017. doi: 10.26434/chemrxiv.5309668.v3. URL https://chemrxiv.org/articles/ORGANIC_1_pdf/5309668.

[29] M. Y. Shen and A. Sali. Statistical potential for assessment and prediction of protein structures. *Protein Sci.*, 15(11):2507–2524, Nov 2006.

[30] Ian Sillitoe, Natalie Dawson, Tony E Lewis, Sayoni Das, Jonathan G Lees, Paul Ashford, Adeyelu Tolulope, Harry M Scholes, Ilya Senatorov, Andra Bujan, et al. Cath: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic acids research*, 47(D1):D280–D284, 2018.

[31] Sam Sinai, Eric Kelsic, George M. Church, and Martin A. Nowak. Variational auto-encoding of protein sequences. *arXiv e-prints*, art. arXiv:1712.03346, Dec 2017.

[32] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2014.

[33] Jérôme Tubiana, Simona Cocco, and Rémi Monasson. Learning protein constitutive motifs from sequence data. *arXiv e-prints*, art. arXiv:1803.08718, Mar 2018.

[34] Gregory P. Way and Casey S. Greene. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *bioRxiv*, 2017. doi: 10.1101/174474. URL https://www.biorxiv.org/content/early/2017/10/02/174474.

[35] Benjamin Webb and Andrej Sali. Comparative protein structure modeling using modeller. *Current protocols in bioinformatics*, 47(1):5–6, 2014.

[36] Tom White. Sampling generative networks: Notes on a few effective techniques. *CoRR*, abs/1609.04468, 2016. URL http://arxiv.org/abs/1609.04468.

[37] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. *arXiv e-prints*, art. arXiv:1609.05473, Sep 2016.

[38] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial Feature Matching for Text Generation. *arXiv e-prints*, art. arXiv:1706.03850, Jun 2017.

6 Supplementary

6.1 Deep Neural Network architecture

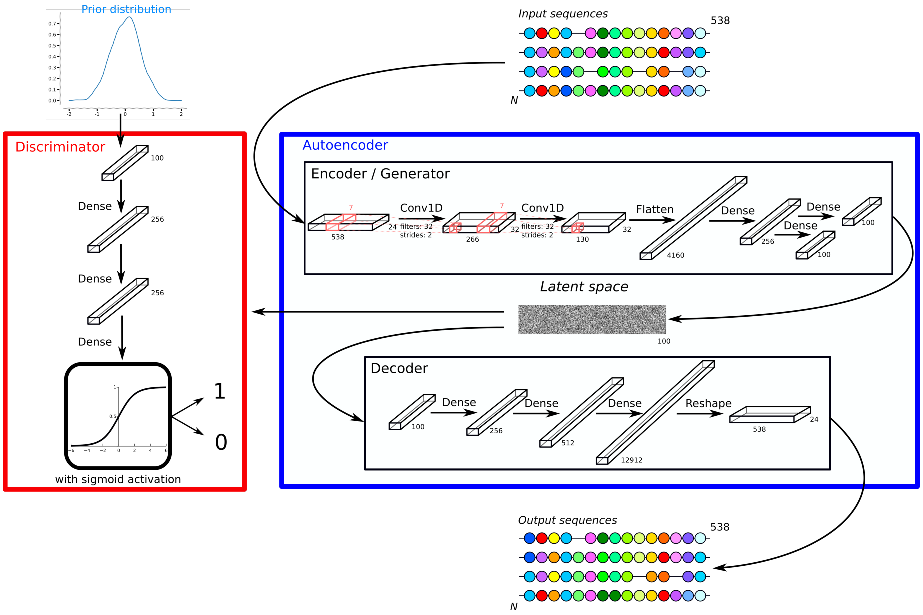

Figure 9: The Adversarial AutoEncoder architecture number 3 presented in Table 2. The discriminator (in red) take as input data from a prior distribution or the latent space computed by the encoder/generator. Using a sigmoid activation function, the discriminator is trained to distinguish between the two types of data. By updating the weight of the encoder/generator based on the discriminator performances, the encoder/generator learn to approximate the prior distribution and fool the discriminator. The autoencoder architecture (in blue) corresponds to a variational autoencoder. Latent space is decoded by a decoder and new sequences are generated using a softmax activation function.

6.2 Enzymatic and taxonomic specificity

Table 3: Enzymatic classes and taxonomic homogeneity of encoded sulfatases after clustering by HDBSCAN.

cluster index	Taxonomic group	percentage of proteins with identical taxa	Enzyme class	percentage of proteins with identical EC
12	Proteobacteria	0.34	EC:3.1.6.1	0.96
27	Proteobacteria	0.98	EC:3.1.6.1	0.89
28	Proteobacteria	0.93	EC:3.1.6.1	1.00
34	Bacteroidetes	0.65	EC:3.1.6.6	0.67
38	Ascomycota	0.63	EC:3.10.1.1	0.50
39	Arthropoda	0.48	EC:3.10.1.1	1.00
44	Ascomycota	0.91	EC:3.1.6.1	1.00
45	Chordata	0.60	EC:3.1.6.14	1.00
46	Actinobacteria	1.00	EC:3.1.6.14	0.67
47	Bacteroidetes	0.69	EC:3.1.6.6	0.69
63	Bacteroidetes	0.67	EC:3.1.6.1	0.60
64	Bacteroidetes	0.81	EC:3.1.6.1	0.70
65	Bacteroidetes	0.64	EC:3.1.6.14	0.50
66	Bacteroidetes	0.69	EC:3.1.6.1	1.00
67	Ascomycota	0.49	EC:3.1.6.1	0.50
68	Bacteroidetes	0.95	EC:3.1.6.6	0.67
73	Planctomycetes	0.46	EC:3.1.6.6	0.71
74	Bacteroidetes	0.80	EC:3.1.6.6	0.50
76	Planctomycetes	0.51	EC:3.1.6.6	0.67
84	Chordata	0.42	EC:3.1.6.13	1.00
86	Bacteroidetes	0.62	EC:3.1.6.13	0.40
87	Bacteroidetes	0.74	EC:3.1.6.13	0.83
101	Chordata	0.88	EC:3.1.6.2	0.44
102	Chordata	0.80	EC:3.1.6.4	1.00
103	Chordata	0.89	EC:3.1.6.8	1.00
111	Chordata	0.82	EC:3.1.6.12	1.00
112	Arthropoda	0.97	EC:3.1.6.12	1.00

Table 4: Enzymatic classes and taxonomic homogeneity of encoded HUP proteins after clustering by HDBSCAN.

cluster index	Taxonomic group	percentage of proteins with identical taxa	Enzyme class	percentage of proteins with identical EC
0	Candidatus	0.70	EC:6.1.1.1	1.00
1	Candidatus	0.74	EC:6.1.1.1	1.00
2	Candidatus	0.38	EC:6.1.1.1	1.00
3	Euryarchaeota	0.98	EC:6.1.1.2	1.00
4	Euryarchaeota	0.96	EC:6.1.1.1	1.00
5	Streptophyta	1.00	EC:6.1.1.1	0.99
6	Chloroflexi	0.90	EC:6.1.1.1	1.00
7	Arthropoda	0.98	EC:6.1.1.1	1.00
8	Ascomycota	0.99	EC:6.1.1.1	1.00
9	Crenarchaeota	0.46	EC:6.1.1.1	1.00
10	Chordata	1.00	EC:6.1.1.1	1.00
11	Euryarchaeota	0.94	EC:6.1.1.1	1.00
12	Streptophyta	0.36	EC:6.1.1.1	1.00
13	Euryarchaeota	0.24	EC:6.1.1.2	1.00
14	Ascomycota	0.32	EC:6.1.1.2	0.94
15	Bacteroidetes	1.00	EC:6.1.1.1	1.00
16	Candidatus	0.35	EC:6.1.1.1	1.00
17	Ascomycota	0.41	EC:6.1.1.1	1.00
18	Euryarchaeota	0.96	EC:6.1.1.2	1.00
19	Ascomycota	0.99	EC:6.1.1.2	1.00
20	Candidatus	0.86	EC:6.1.1.2	1.00
21	Chloroflexi	0.28	EC:6.1.1.2	1.00
22	Proteobacteria	0.92	EC:6.1.1.1	1.00
23	Candidatus	0.71	EC:6.1.1.2	1.00
24	Cyanobacteria	0.98	EC:6.1.1.1	1.00
25	Proteobacteria	0.99	EC:6.1.1.1	1.00
26	Firmicutes	0.96	EC:6.1.1.1	1.00
27	Firmicutes	0.98	EC:6.1.1.1	1.00
28	Proteobacteria	0.98	EC:6.1.1.1	1.00
29	Firmicutes	1.00	EC:6.1.1.1	1.00
30	Firmicutes	0.98	EC:6.1.1.1	1.00
31	Actinobacteria	0.99	EC:6.1.1.2	0.97
32	Chordata	0.86	EC:6.1.1.2	1.00
33	Bacteroidetes	0.96	EC:6.1.1.1	1.00
34	Firmicutes	0.98	EC:6.1.1.1	1.00
35	Actinobacteria	0.99	EC:6.1.1.1	1.00
36	Actinobacteria	1.00	EC:6.1.1.1	1.00
37	Proteobacteria	0.99	EC:6.1.1.1	1.00
38	Proteobacteria	0.99	EC:6.1.1.1	1.00
39	Proteobacteria	0.56	EC:6.1.1.2	1.00
40	Proteobacteria	0.99	EC:6.1.1.2	1.00
41	Firmicutes	0.73	EC:6.1.1.2	1.00
42	Proteobacteria	0.99	EC:6.1.1.2	1.00
43	Bacteroidetes	0.97	EC:6.1.1.2	1.00
44	Actinobacteria	1.00	EC:6.1.1.2	1.00
45	Actinobacteria	1.00	EC:6.1.1.2	1.00
46	Proteobacteria	0.96	EC:6.1.1.2	1.00

Table 5: Enzymatic classes and taxonomic homogeneity of encoded TPP proteins after clustering by HDBSCAN.

cluster index	Taxonomic group	percentage of proteins with identical taxa	Enzyme class	percentage of proteins with identical EC
0	Firmicutes	0.75	EC:2.2.1.1	1.00
1	Bacteroidetes	1.00	EC:2.2.1.1	1.00
2	Thermotogae	0.32	EC:2.2.1.1	1.00
3	Ascomycota	0.73	EC:1.2.4.1	1.00
4	Proteobacteria	0.44	EC:1.2.7.3	1.00
5	Proteobacteria	0.84	EC:1.2.4.4	1.00
6	Proteobacteria	0.67	EC:2.2.1.1	0.96
7	Proteobacteria	0.73	EC:2.2.1.1	1.00
8	Proteobacteria	0.90	EC:2.2.1.7	1.00
9	Actinobacteria	0.82	EC:1.2.4.1	0.68
10	Candidatus	0.80	EC:2.2.1.1	1.00
11	Firmicutes	0.67	EC:2.2.1.7	1.00
12	Proteobacteria	0.64	EC:2.2.1.1	1.00
13	Firmicutes	0.54	EC:2.2.1.1	1.00
14	Firmicutes	0.89	EC:2.2.1.1	0.98
15	Euryarchaeota	0.91	EC:2.2.1.1	1.00
16	Actinobacteria	0.44	EC:2.2.1.1	0.99
17	Proteobacteria	0.74	EC:1.2.4.1	1.00
18	Actinobacteria	0.68	EC:1.2.3.3	0.33
19	Proteobacteria	0.57	EC:1.2.7.1	1.00
20	Verrucomicrobia	0.77	EC:2.2.1.7	1.00
21	Candidatus	0.46	EC:2.2.1.1	1.00
22	Bacteroidetes	0.97	EC:2.2.1.1	0.86
23	Firmicutes	0.62	EC:3.7.1.22	0.95
24	Verrucomicrobia	0.88	EC:2.2.1.1	1.00
25	Proteobacteria	0.97	EC:3.7.1.22	1.00
26	Proteobacteria	1.00	EC:3.7.1.22	1.00
27	Firmicutes	1.00	EC:2.2.1.7	1.00
28	Proteobacteria	0.98	EC:1.2.4.1	1.00
29	Actinobacteria	0.87	EC:1.2.4.1	1.00
30	Proteobacteria	0.93	EC:2.2.1.1	0.99
31	Firmicutes	0.98	EC:2.2.1.7	1.00
32	Cyanobacteria	0.33	EC:1.2.4.1	1.00
33	Ascomycota	0.99	EC:2.2.1.3	0.50
34	Actinobacteria	0.92	EC:3.7.1.22	0.72
35	Firmicutes	0.99	EC:2.2.1.7	1.00
36	Proteobacteria	0.83	EC:2.2.1.7	1.00
37	Bacteroidetes	0.98	EC:2.2.1.7	1.00
38	Firmicutes	0.35	EC:1.2.7.1	0.68
39	Bacteroidetes	0.97	EC:2.2.1.7	1.00
40	Bacteroidetes	0.98	EC:2.2.1.7	1.00
41	Actinobacteria	0.99	EC:2.2.1.7	1.00
42	Actinobacteria	0.98	EC:2.2.1.7	1.00
43	Actinobacteria	0.99	EC:2.2.1.6	0.99
44	Proteobacteria	0.93	EC:2.2.1.6	1.00
45	Cyanobacteria	1.00	EC:2.2.1.6	1.00
46	Proteobacteria	0.88	EC:2.2.1.6	1.00
47	Proteobacteria	0.98	EC:2.2.1.6	1.00
48	Ascomycota	1.00	EC:2.2.1.6	1.00
49	Proteobacteria	0.58	EC:2.2.1.1	1.00
50	Proteobacteria	0.58	EC:2.2.1.1	0.92

6.3 Latent space arithmetic

6.3.1 Strategy 1

First strategy consists to add the mean latent space, computed using the encoder on the sequences of the source sub-family, to the encoded sequences of the query sub-family.

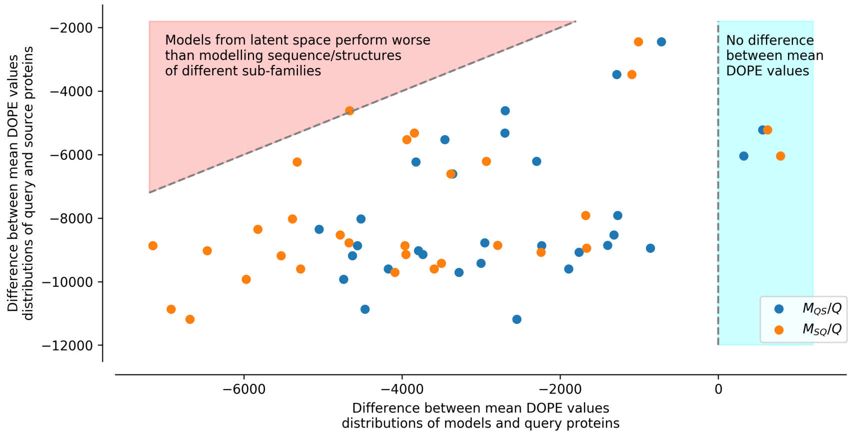

Figure 10: See Figure 7 for legend.

6.3.2 Strategy 2

Second tested strategy differs from the first one by subtracting the mean background latent space, computed from the latent space of all sub-families, from the latent space of the query sub-family.

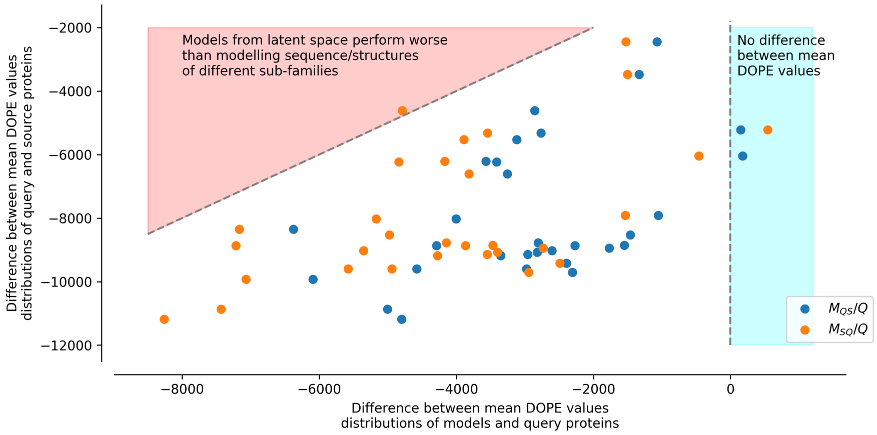

Figure 11: See Figure 7 for legend.

6.3.3 Strategy 3

Third strategy differs to the second as the background strategy is computed using all sub-families except sub-families source and query.

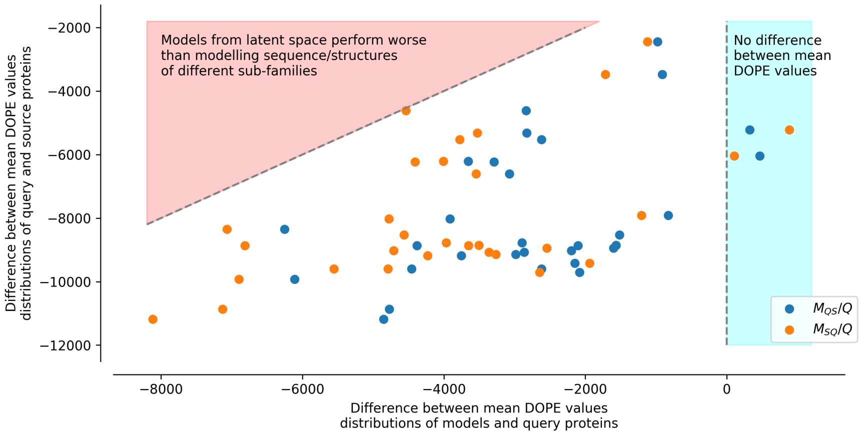

Figure 12: See Figure 7 for legend.

6.3.4 Strategy 4

In the fourth strategy, the subtraction is performed using a local KD-tree to only remove features shared by closest members of a given query and addition is performed randomly selecting a member of the the source family and it's closest 10 members.

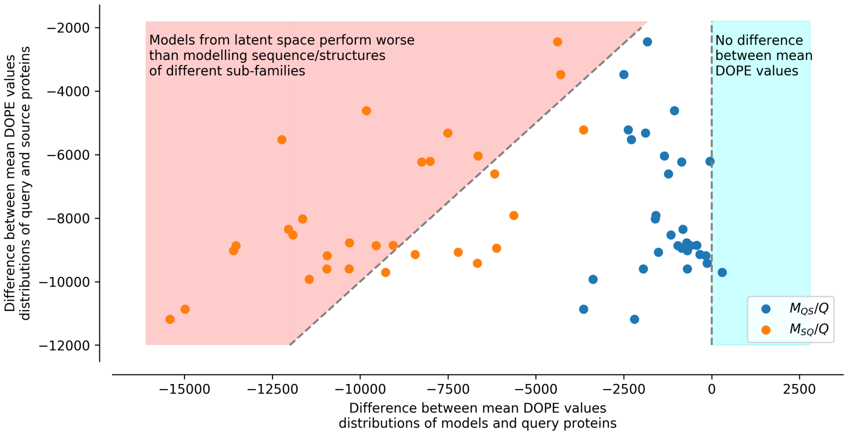

Figure 13: See Figure 7 for legend.