Bioinformatics, 34(15), 2018, 2642–2648
doi: 10.1093/bioinformatics/bty178
Advance Access Publication Date: 23 March 2018
Original Paper

OXFORD

Data and text mining

Learned protein embeddings for machine learning

Kevin K. Yang[1], Zachary Wu[1], Claire N. Bedbrook[2] and Frances H. Arnold[1,2,*]

[1]Division of Chemistry and Chemical Engineering and [2]Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, CA 91125, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on November 30, 2017; revised on March 20, 2018; editorial decision on March 21, 2018; accepted on March 22, 2018

Abstract

Motivation: Machine-learning models trained on protein sequences and their measured functions can infer biological properties of unseen sequences without requiring an understanding of the underlying physical or biological mechanisms. Such models enable the prediction and discovery of sequences with optimal properties. Machine-learning models generally require that their inputs be vectors, and the conversion from a protein sequence to a vector representation affects the model's ability to learn. We propose to learn embedded representations of protein sequences that take advantage of the vast quantity of unmeasured protein sequence data available. These embeddings are low-dimensional and can greatly simplify downstream modeling.

Results: The predictive power of Gaussian process models trained using embeddings is comparable to those trained on existing representations, which suggests that embeddings enable accurate predictions despite having orders of magnitude fewer dimensions. Moreover, embeddings are simpler to obtain because they do not require alignments, structural data, or selection of informative amino-acid properties. Visualizing the embedding vectors shows meaningful relationships between the embedded proteins are captured.

Availability and implementation: The embedding vectors and code to reproduce the results are available at https://github.com/fhalab/embeddings_reproduction/.

Contact: frances@cheme.caltech.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Machine learning (ML) has been used to predict protein properties from protein sequences to enable protein design and engineering (Bedbrook *et al.*, 2017b; Fox *et al.*, 2007; Romero *et al.*, 2013). ML models are useful for predicting the outcomes of complex processes, such as how a protein sequence encodes function, because they do not require prior knowledge of specific physical or biological mechanisms. Instead, after training with measured sequences, ML models infer the properties of unseen sequences. A model capable of predicting the properties of unseen protein sequences enables prediction and discovery of sequences with optimal properties. For ML models to learn about protein sequences, we must encode the protein sequence in a form compatible with the mathematical operations used in ML models. Generally, this requires that the protein sequence be encoded as a vector or matrix of numbers. How each protein sequence is encoded determines what can be learned (Domingos, 2012). Even the most powerful models produce poor results if an inappropriate encoding is used. We show that learning these encodings from data can streamline machine-learning pipelines while achieving high predictive accuracies.

A protein sequence can be encoded by its physical properties or directly by its amino acids (Alipanahi *et al.*, 2015; Bedbrook *et al.*, 2017b; Chang *et al.*, 2016; Fox *et al.*, 2007; Ofer and Linial, 2015; Romero *et al.*, 2013; Saladi *et al.*, 2018). When using physical

properties to encode a protein sequence, each individual amino acid is represented by a collection of physical properties, such as its charge or hydrophobicity, and each protein is taken to be a combination of those properties. Properties of the bulk protein, such as its predicted secondary structures, can also be used to represent the protein. However, there are countless physical properties that could be used to describe each amino acid/protein, and the molecular properties that dictate functional properties are unknown, highly constrained, and differ between different functional properties. Therefore, selecting informative properties is challenging because it is difficult to know *a priori* what properties will be predictive for a particular task.

Instead of representing a protein with physical properties, one can directly encode its amino acid sequence. A protein sequence of length L can be encoded as an L x n matrix, where n is the number of amino acids. Each row in the matrix consists of $(n - 1)$ 0 s and a single 1, with the position of the 1 indicating the amino acid residue at that position in the protein. This vectorization method for categorical data is known as one-hot encoding. One-hot encodings are inherently sparse, memory-inefficient and high-dimensional. In a one-hot encoding, there is no notion of similarity between sequence or structural elements: they are either identical, or not. For example, in a one-hot encoding of words, the words 'king', 'prince' and 'pot' are all not identical and thus equidistant from each other even though 'king' and 'prince' are intuitively more similar in meaning than 'king' and 'pot' or 'prince' and 'pot.' Similarly, to the biologist, an amino acid sequence of DDD is more similar in meaning to EEE than to PPP or HHH. Furthermore, one-hot encodings of the primary sequence require that all sequence variants of interest are aligned. This alignment must be updated as sequences are added to the model. If updating the alignment changes its length, or even if amino acids are added or removed, the dimensionality of the encoding changes. Multiple sequence alignments between distantly-related proteins require visual validation because there is no universal standard for choosing the best alignment. Even with visual validation, it is challenging to confidently align distantly-related sequences. If the sequences are misaligned, the inputs to ML model are flawed, and there can be little expectation of success.

While there are a massive number of known protein sequences, only a tiny fraction have measured properties relevant to any specific task. Sequences with a measurement for the prediction task are known as labeled sequences, while those that do not are unlabeled sequences. The number of known unlabeled sequences will continue to rise as the cost of DNA-sequencing decreases, but there is no universal method for measuring all relevant protein properties. Therefore, the gap between the number of unlabeled and labeled sequences will continue to grow. However, even unlabeled sequences contain information about the frequency and patterns of amino acids selected by evolution to compose proteins. Information contained in unlabeled sequences may be helpful when predicting properties for a specific set of sequences, especially if the set in question is small. Specifically, instead of selecting physical properties or using a one-hot encoding of the sequence, a continuous vector encoding of each sequence can be learned from unlabeled sequences. This representation contains relevant information about the protein sequence learned from the distribution of sequences in the unlabeled set and is known as an embedded representation because it embeds the protein sequences in a vector space.

The process of using unlabeled data to learn an embedded representation has been well-established by recent work in natural language processing, where word and document embeddings are used as an efficient way to encode text for use in sentiment analysis, machine translation and other tasks (Young *et al.*, 2016). These examples learn an embedded representation from a large collection of unlabeled texts by assuming that words that appear in similar contexts have similar meanings. The unlabeled texts are analogous to the large number of unlabeled protein sequences. For example, the word2vec model (Mikolov *et al.*, 2013a,b) uses a shallow two-layer neural network to learn embeddings using one of two architectures: skip-gram and continuous bag-of-words. In the skip-gram architecture, the model uses the current word to predict its surrounding context words. In contrast, in the continuous bag-of-words architecture, the current word is predicted from its surrounding context words. The doc2vec model (Le and Mikolov, 2014) extends word2vec by learning embeddings for entire sentences, paragraphs, or documents.

There have been efforts to apply word2vec and doc2vec to represent protein sequences (Asgari and Mofrad, 2015; Kimothi *et al.*, 2016; Mazzaferro, 2017; Ng, 2017). These embeddings treat the amino acid sequence as a document and fragments of the amino acid sequence of constant length k (k-mers) as words. As shown in Figure 1, a sequence of nine amino acids can be divided into three sets of non-overlapping 3-mers. The learned k-mer embeddings place k-mers that occur in similar contexts near each other in the embedded space by learning to predict a k-mer from its surrounding context k-mers and the sequence embedding. These embeddings have achieved high accuracy in differentiating ordered and disordered proteins and modest accuracy in classifying proteins from SwissProt into families based only on their primary sequence (Asgari and Mofrad, 2015). Our goal was to test if such embeddings can be used in ML to predict specific properties of related proteins. This is a fundamentally different problem than classifying proteins into families or predicting a universal binary property across all proteins because the model must tease apart the effects of subtle sequence changes from limited labeled data for a specific property.

In this work, we train embedded representations for four protein property prediction tasks. These tasks cover a range of protein

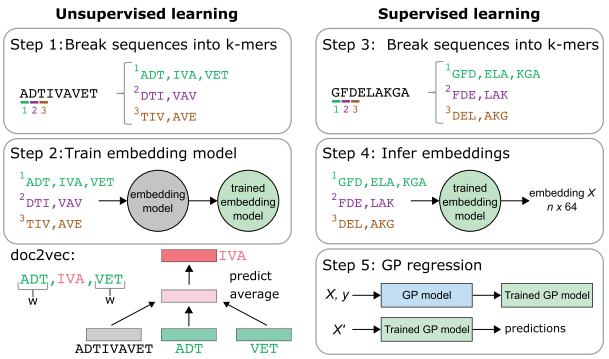

Fig. 1. The modeling scheme. First, an unsupervised embedding model is trained on 524 529 unlabeled sequences pulled from the UniProt database. The UniProt sequences are broken into k lists of non-overlapping k-mers (Step 1), and then the lists are used to train the embedding model (Step 2). The doc2vec embedding model learns to predict the vectors for center k-mers from the vectors for their surrounding context k-mers and the sequence vectors. These sequence vectors are then the embedded representations of the sequences. Next, information learned during the unsupervised phase is applied during supervised learning with labeled sequences. The labeled sequences for each task (localization, T50, absorption and enantioselectivity) are first broken into k lists of non-overlapping k-mers (Step 3). An embedding is then inferred for each sequence using the trained embedding model (Step 4). n is the number of labeled sequences. Finally, during GP regression (Step 5), the inferred training embeddings X' and the training labels y are used to train a GP regression model, which can then be used to make predictions

families, measured properties and library designs. We show that the predictive power of models trained using these embeddings is comparable to and sometimes exceeds those trained on one-hot encodings, physical amino acid properties, or string mismatch kernels (Leslie *et al.*, 2004). This suggests that embeddings enable accurate predictions despite having orders of magnitude fewer dimensions and being simpler to obtain because they do not require alignments, structural data, or selection of relevant amino-acid properties. Finally, we visualize the geometry of the embedding vectors, which captures meaningful relationships between the embedded proteins.

2 Materials and methods

2.1 Modeling scheme

Figure 1 shows the two-part modeling scheme. Unsupervised doc2vec embedding models were trained on 524, 529 protein sequences with lengths between 50 and 999 amino acids (mean length 326) obtained from UniProt using the distributed memory architecture (The UniProt Consortium, 2017). In the distributed memory architecture, the model learns to predict the central k-mer based on the sequence embedding and the embeddings for a context window of k-mers on either side of the central k-mer. The size of the context, i.e. how many k-mers on either side to consider, is the window width (w), which can be adjusted in the embedding model. Each sequence was broken into k lists of non-overlapping k-mers. For example, for $k=3$, there are three lists and each list begins at one of the first three amino acid positions of the sequence, as shown in Figure 1. Unsupervised embedding model training was performed using the lists derived from the UniProt sequences. After unsupervised embedding model training, the embedding model was used to infer encodings of sequences for input to supervised Gaussian process (GP) regression models (Rasmussen and Williams, 2006). Embeddings for the sequences relevant to each task were determined by averaging the embeddings for the k lists of k-mers corresponding to each task sequence. It was found that GP performance was highly dependent on the order in which the embeddings for these sequences were inferred. Therefore, embeddings for each of the three tasks studied were calculated as the average of 100 inference runs with random input orders. These embeddings represent each task sequence in a very compact, low-dimensional form. We learn embeddings with between 4 and 128 dimensions. By comparison, the other representations used for comparison in this work have between 10^3 and 10^5 dimensions. In addition, sequences from disparate protein families are embedded in the same vector space, allowing comparisons between distant sequences and streamlining down-stream modeling. All doc2vec training and inference was performed in Gensim (Rurek and Sojka, 2010).

For some tasks, it was found that randomizing the UniProt sequences by shuffling or resampling before unsupervised embedding model training improved downstream performance. Shuffling refers to scrambling the order of amino acids for each sequence. Alternatively, resampling refers to drawing sequences of the original lengths according to the overall observed amino acid frequency for the UniProt sequences (resample-UniProt) or according to uniform amino acid frequency (resample-uniform). The embedding model is then trained on these randomized sequences instead of the original UniProt sequences. We suspect that this has a regularizing effect on the embedding model: randomization prevents the embedding model from overfitting to a set of protein sequences that is not representative of those in the task. This also suggests that one of the key pieces of information the unsupervised embedding model learns is the frequency with which different amino acids occur in the same proteins.

The data for each task are taken from different protein engineering projects. When building a model that must generalize across diverse families of proteins, the best practice is to minimize sequence redundancy between the training and test sets (Abbasi and Minhas, 2016). However, protein-engineering projects typically generate data in a stepwise manner, where each subsequent set of sequences characterized is determined by previously characterized sequences. Therefore, we split the training and test sets such that the training sets contain sequences from earlier steps than those in the test sets, which come from later steps. This provides a realistic simulation of machine learning usage in protein engineering.

All embedding models were trained for 25 epochs. Embedding hyperparameters were chosen using 20-fold cross-validation on the training sets. We set the dimension to 64 and considered values of k between 1 and 5, and values of w between 1 and 7. We used GP regression models with Matérn kernels with $\nu = 5/2$. The noise and kernel hyperparameters were optimized by maximizing the marginal likelihood (Rasmussen and Williams, 2006). A GP model trained on the entire training set was then used to predict the relevant properties for test set sequences. GP models trained on embedded representations were compared to models trained on one-hot representations of amino acid sequence, mismatch string kernels with $k=5$ and $m=1$, ProFET (Ofer and Linial, 2015) and a subset of AAIndex (Kawashima *et al.*, 2008). ProFET represents each sequence by extracting elementary biophysical and sequence-derived features. AAIndex is a set of 553 properties for each of the 20 amino acids. 64 of these properties were chosen by greedily maximizing the average cosine distance between the chosen properties. Each amino acid is therefore represented by a vector of 64 properties, and each protein is represented by concatenating the property vectors for its amino acid sequence. For two of the four tasks, structural information was available. For those tasks, models were also compared to a GP model trained on a one-hot representation of both the sequence and the structure. The structure was encoded in these cases by a binary indicator vector for the identity of each pair of amino acids within 4.5 Angstroms in the crystal structure (Romero *et al.*, 2013).

2.2 Tasks

We tested embeddings on four tasks with diverse proteins, different measured properties and various methods of generating the original library. The data for these tasks were collected from previous studies and will only briefly be described here.

Channelrhodopsin (ChR) localization ('Localization') Two separate, ten-block recombination libraries were designed from three parental ChRs (CheRiff, C1C2 and CsChrimsonR). Each chimeric ChR variant in these libraries is composed of blocks of sequence from the parental ChRs. The data for this task comprise a total of 248 sequences. Genes for these sequences were synthesized and expressed in human embryonic kidney (HEK) cells, and their membrane localization was measured (Bedbrook *et al.*, 2017a).

Cytochrome P450 thermostability ('T50') An eight-block recombination library was designed from three parental cytochrome P450s (CYP102A1, CYP102A2 and CYP102A3) (Li *et al.*, 2007). The data for this task include 242 sequences from this library and 19 chimeric cytochrome P450s generated from other parents or cross-over points (Romero *et al.*, 2013), for a total of 261 sequences. Genes for these sequences were expressed in *Escherichia coli* and their T50s (temperature at which half of the protein was irreversibly inactivated after a 10-minute incubation) were measured.

Table 1. Summary of tasks used to evaluate embedded representations

Task	n	Protein	Library	Property	Citation
Localization	248	Channelrhodopsin	Recombination	Plasma membrane localization	Bedbrook *et al.* (2017a)
T50	261	Cytochrome P450	Recombination	Thermostability	Li *et al.* (2007) and Romero *et al.* (2013)
Absorption	81	Bacterial rhodopsin	Site-saturation	Peak absorption wavelength	Engqvist *et al.* (2015)
Enantioselectivity	152	Epoxide hydrolase	Site-saturation	Enantioselectivity	Zaugg *et al.* (2017)

Table 2. Comparison of learned, dense, embedded representations, ProFET, AAIndex properties, mismatch string kernels and one-hot representations of sequence and structure for predicting protein properties using GP regression

Task	n_{train}	n_{test}	Representation	d	*MAE*	τ	log P
Localization	215	33	Embedding	**64**	**0.73**	**0.60**	−43.5
			One-hot seq. and struct.	600 747	0.76	**0.60**	**−43.2**
			One-hot sequence	7161	0.76	0.59	−43.7
			Mismatch kernel	–	0.86	0.55	−54.6
			ProFET	1173	1.03	0.32	−54.9
			AAIndex properties	21 824	0.76	0.55	−44.3
T50	242	19	Embedding	**64**	**2.91**	**0.61**	−59.5
			One-hot seq. and struct.	994 980	2.98	0.53	−57.3
			One-hot sequence	9786	2.94	0.57	−57.2
			Mismatch kernel	–	4.03	0.38	−58.5
			ProFET	1173	4.93	0.43	−63.7
			AAIndex properties	29 824	2.95	0.51	**−56.2**
Absorption	62	19	Embedding	**64**	23.3	0.57	−109.2
			One-hot sequence	6258	22.1	0.63	−111.0
			Mismatch kernel	–	**17.8**	**0.68**	**−103.9**
			ProFET	1173	53.5	0.32	−174.7
			AAIndex properties	19 072	30.1	0.35	−116.4
Enantioselectivity	136	16	Embedding	**64**	9.14	**0.64**	−64.5
			One-hot sequence	8358	8.16	0.50	**−63.3**
			Mismatch kernel	–	**7.50**	0.46	−65.1
			ProFET	1173	27.9	0.27	−76.7
			AAIndex properties	25 472	12.5	0.25	−65.7

Notes: n_{train} and n_{test} are the number of training and test examples, respectively. d is the dimension of the representation. *MAE* is the mean absolute error between predicted test values and the actual test values. τ is the Kendall τ between the predicted test values and the actual test values. log P is the log Gaussian likelihood of the actual test values given the predicted distributions. All reported metrics are for the held-out test set. All embedding hyperparameters were chosen using 20-fold cross-validation on the training set. The best performance on each metric for each task is shown in bold.

Rhodopsin absorption wavelength ('Absorption') Amino acid substitutions were made in the retinal-binding pocket of *Gloeobacter violaceus* rhodopsin (GR) in order to tune its peak absorption wavelength. GR is a light-activated proton pump Engqvist *et al.*, (2015). The data for this task consist of GR and 80 blue- and red-shifted variants with 1–5 mutations generated in the course of tuning its absorption wavelength, for a total of 81 sequences.

Epoxide hydrolase enantioselectivity ('Enantioselectivity') Amino acid substitutions were made in the binding pocket of the epoxide hydrolase (EH) from *Aspergillus niger* in order to improve its preference for the (*S*)-enantiomer of glycidyl phenyl ether. The data for this task consist of EH and 151 variants with 1–8 mutations generated in the course of improving its enantioselectivity, for a total of 152 variants (Zaugg *et al.*, 2017).

These four tasks include light-sensitive integral membrane proteins (ChR and GR) and soluble enzymes (cytochrome P450 and EH). The tasks include libraries constructed via recombination and site-directed mutagenesis and examine a variety of protein properties. The diversity of tasks allows us to evaluate the generality of embedded representations. Table 1 summarizes the tasks. Sequences and measurements are provided as Datasets 1–5 in the Supplementary Material.

3 Results and discussion

We compared the quality of predictions for GP models trained on different encodings. Table 2 compares the GP regression results on the test set for each task using embeddings, physical properties from AAIndex, ProFET, a mismatch kernel with $k = 5$ and $m = 1$, and one-hot encodings. Supplementary Figures S1–S4 compare the actual test values to those predicted by GP regression models trained using each encoding. The embedding hyperparameters chosen for localization are shuffled, $k = 3$ and $w = 5$. For T50, they are no randomization, $k = 3$ and $w = 7$. For absorption, they are resample-uniform, $k = 4$ and $w = 1$. For enantioselectivity, they are resample-UniProt, $k = 3$ and $w = 7$. The cross-validation metrics for each task and each set of embedding hyperparameters are included as Datasets 5–8 in the Supplementary Material. GP regression predicts a Gaussian distribution, defined by its mean and variance, for each evaluation sequence. Predictions were evaluated using the mean absolute error (MAE), the Kendall τ (τ) and the Gaussian log-likelihood (log P). The MAE measures deviation between predicted and actual values, τ measures ordinal accuracy, and log-likelihood provides a probabilistic measurement of model fit. Together, these three metrics provide a multifaceted comparison between different models.

Table 3. Negative controls

Task	Control	*MAE*	τ	log *P*
Localization	–	0.73	0.60	−43.5
	Task sequences only	0.86	0.50	−50.0
	Shuffled task sequences	1.21	0.16	−57.4
	Shuffled training labels	1.16	−0.39	−58.3
T50	–	2.91	0.61	−59.5
	Task sequences only	5.02	0.45	−63.3
	Shuffled task sequences	4.49	0.31	−61.8
	Shuffled training labels	5.72	−0.35	−67.1
Absorption	–	23.3	0.57	−109.2
	Task sequences only	61.4	0.34	−162.1
	Shuffled task sequences	61.4	−0.03	−162.0
	Shuffled training labels	61.4	−0.43	−162.0
Enantioselectivity	–	9.14	0.64	−64.5
	Task sequences only	41.3	−0.06	−85.2
	Shuffled task sequences	42.7	0.27	−84.7
	Shuffled training labels	42.8	0.06	−84.8

Notes: Each task was repeated using embeddings only learned on the task sequences (Task sequences only), using embeddings inferred from task sequences with the order of amino acids in each sequence randomized (Shuffled task sequences), and with the original embeddings but the training labels randomized between the sequences (Shuffled training labels). Results for the embeddings selected by cross-validation for each task are included for comparison.

For localization, embeddings trained on UniProt sequences slightly outperform one-hot encodings of sequence and structure. Previously, we showed that models built on one-hot encodings were sufficiently accurate to identify sequences that maximize localization (Bedbrook *et al.*, 2017b). For T50, embeddings achieve the best MAE and τ, while AAIndex achieves the highest log-likelihood. The one-hot encodings are comparable, while the mismatch kernel and ProFET perform much worse. Likewise, models built on one-hot encodings were previously shown to be sufficiently accurate in identifying sequences that maximize the T50 (Romero *et al.*, 2013). For absorption, the mismatch kernel achieves the best performance across metrics, embeddings and the one-hot sequence encoding are comparable, while ProFET and AAIndex perform much worse. Finally, for enantioselectivity, embeddings achieve comparable performance to the one-hot sequence encoding and the mismatch kernel while ProFET and AAIndex are much worse.

For three of the four tasks, the embeddings are the most accurate by at least one metric even though they have several orders of magnitude fewer dimensions than the other representations. Mismatch string kernels are calculated directly from the amino acid sequences without an intermediate vector representation and therefore have no dimension. This shows that embeddings can be used as a low-dimensional representation of protein sequences for building machine-learning models of protein function. The training time for GP regression is dominated by the $O(n^3)$ time to invert the covariance matrix. However, on a 2016 Macbook Pro, models using 64-dimensional embeddings train approximately 10 times faster than those using one-hot embeddings of sequence and structure.

To better evaluate the information gained by the embedding model, we performed three negative controls, which are summarized in Table 3. First, we trained embedding models only on those sequences used in the task: during unsupervised embedding model training, we replaced the ∼500 000 UniProt sequences with the 81–261 sequences to be inferred. This decreased GP regression performance, suggesting that information from the unlabeled sequences

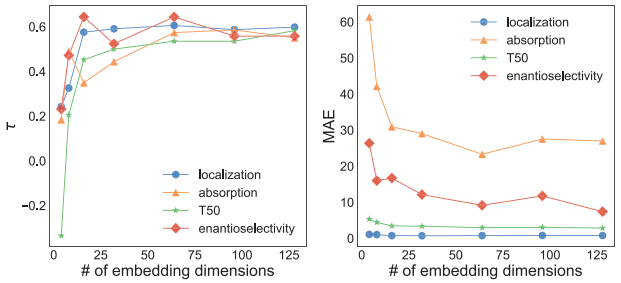

Fig. 2. Effect of embedding dimension on predictive accuracy. For each task, embeddings of varying dimensions were trained and then used for GP regression. The resulting model quality was then evaluated using the Kendall τ and MAE

improves predictions and therefore that the unsupervised embedding model is learning sequence-specific information from the unlabeled sequence data. Second, we confirmed that scrambling the order of the amino acids in the task-specific sequences before inferring their embeddings also decreases regression performance. This demonstrates that the embedding model is encoding useful information about the task sequences during the inference step, including information related to the order of the amino acids. Finally, we shuffled the training labels (i.e. the measured properties) for each sequence in the training set but not the test labels, which should remove the model's ability to learn anything about the test set from the training set. These negative controls show that the embedding model is applying information from the unlabeled sequences to learn meaningful embeddings for the labeled sequences.

In order to determine how many dimensions are required to represent a protein sequence, we compared GP model performance for embeddings inferred from lower-dimensional models with other hyperparameters held constant. Figure 2 shows that τ and MAE tend to worsen gradually as d decreases until $d = 16$, and then worsen very steeply. It is likely that predictive performance could be improved by optimizing d simultaneously with the other embedding hyperparameters. These results suggest that ∼32 dimensions encode enough information about a 250–500 amino acid sequence to make predictions of the protein's functional properties.

Likewise, we compared GP model performance for embeddings inferred from subsets of the UniProt sequences with other hyperparameters held constant in order to determine the number of unlabeled sequences necessary for unsupervised embedding model training. Figure 3 illustrates that for localization and T50, both τ and MAE show little improvement as the number of unlabeled sequences increases past 100 000. However, for absorption, MAE continues to decrease as the number of unlabeled sequences increases. For enantioselectivity, τ continues to increase as the number of unlabeled sequences increases. The training sets for absorption and enantioselectivity are smaller than those for localization and T50. In addition, the localization and T50 tasks use data from recombination libraries, and the training sets for these tasks are chosen to maximize information about the unseen members of these libraries, including those in the test sets. However, the absorption and enantioselectivity tasks use data from site-directed mutagenesis experiments, and the training sets are not designed to be informative about the test sets. Therefore, these tasks may benefit more from the additional information gained by unsupervised model training.

To visualize the geometry of the learned embeddings, we used t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) to project the inferred embeddings, AAIndex,

ProFET and one-hot encodings of sequence onto a 2-dimensional space. Projections for ProFET use perplexity 10; the other projections use perplexity 50. Compared to other methods for dimensionality reduction, t-SNE focuses on local structure and tends to extract clustered local groups. Projections were calculated using scikit-learn's implementation of t-SNE with default parameters except where otherwise specified. Figure 4 shows these 2-dimensional projections.

The embeddings for localization cluster around each of the three recombination parents, and variants with fewer mutations from the parents are closer to the parents. The projections for the AAIndex properties, ProFET and one-hot encoding for localization show a similar pattern. The embeddings for T50 also cluster around each of the three recombination parents, and variants with fewer mutations are also closer to the parents. The projections for the AAIndex properties and one-hot encoding for T50 also place variants with fewer mutations closer to the parents, but there are not three clear clusters. The projection for ProFET does not show any clear structure. The embeddings for absorption roughly separate red-shifted and blue-shifted sequences, with the most blue-shifted sequences in a separate cluster. The projections for the AAIndex properties, ProFET and one-hot encoding for absorption show the same blue-shifted cluster and rough separation. The embeddings for enantioselectivity place the sequences with the fewest mutations closest to the parent. The projections for the AAIndex properties, ProFET and one-hot encoding also place variants with fewer mutations closer to the parent. Across the four diverse tasks, the inferred embeddings capture relationships between the sequences in the library.

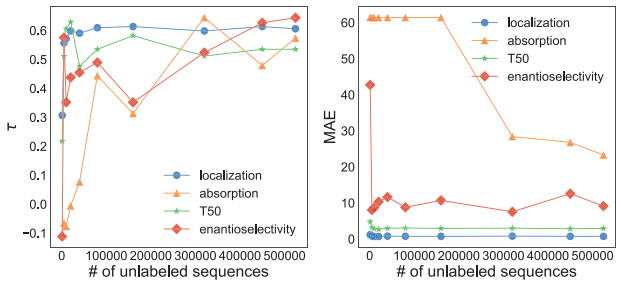

Fig. 3. Effect of number of unlabeled sequences on predictive accuracy. For each task, embeddings were trained on subsets of the UniProt sequences and then used for GP regression. The resulting model quality was then evaluated using the Kendall τ and MAE

The embedding model embeds sequences for all four tasks into the same vector space, so relationships between all the task sequences can also be interrogated. Figure 5 shows a 2-dimensional projection obtained using t-SNE with perplexity 50 for all of the embedded representations. The embeddings for each protein family form their own cluster. Supplementary Figure S5 shows that the clustering of sequences most similar to each parent can still be observed for localization and T50, the absorption sequences still roughly separate by whether they are blue- or red-shifted, and the enantioselectivity sequences are roughly separated by their enantioselectivity.

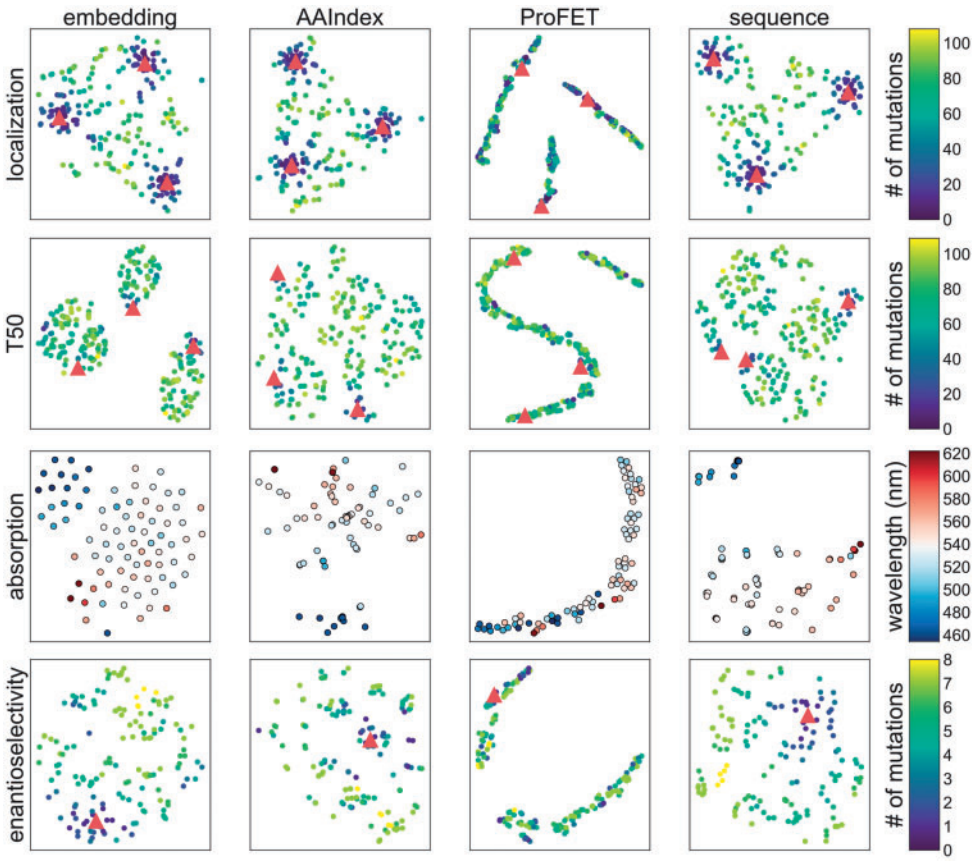

Fig. 4. Visualization of learned vector representations of protein sequences. Vector representations projected onto 2 dimensions using t-SNE with perplexity 50 (embeddings, AAIndex, sequence) or 10 (ProFET). The sequences for the localization, the T50 and the enantioselectivity tasks are colored by the number of mutations from the nearest parent. The sequences for the absorption task are colored by peak absorption wavelength. Parents for localization, T50 and enantioselectivity are indicated by red triangles

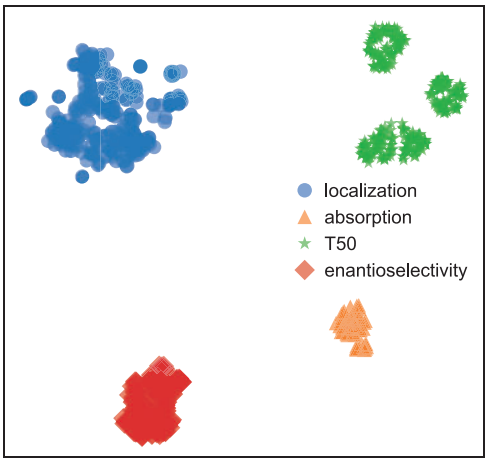

Fig. 5. Combined visualization of vector representations for each of the four tasks. Sequences are colored to show separation between the embeddings for each task (Color version of this figure is available at *Bioinformatics* online.)

4 Conclusions

This work shows that embedding models trained on proteins from UniProt can be applied to predict the functional properties of a small number of related proteins, such as those often encountered in protein engineering. Models trained using embeddings are comparable to and often outperform those trained on one-hot encodings of sequence and structural contacts, mismatch string kernels, or amino acid physical properties across four tasks, showing that embeddings generalize across protein families, library designs and protein properties. As few as 32 dimensions are sufficient to achieve competitive model performance. However, the optimal embedding hyperparameters are highly dependent on the specific task. Negative controls show that the unsupervised embedding model incorporates information from the unlabeled sequences. Furthermore, the inferred embeddings show patterns consistent with the library designs when visualized in a 2-dimensional space. While the number of known protein sequences is rapidly increasing, it remains time-consuming and difficult to measure many protein properties of interest. By first training an unsupervised embedding model on unlabeled protein sequences, we are able to transfer information encoded in these unlabeled sequences to a specific task. This allows predictive models while bypassing many of the difficulties associated with using one-hot encodings and physical properties to represent protein sequences.

Acknowledgements

The authors wish to thank members of the Arnold lab, Justin Bois and Yisong Yue for general advice and discussions on this project.

Funding

This work is supported by the U.S. Army Research Office Institute for Collaborative Biotechnologies [W911F-09-0001 to F.H.A., K.K.Y.], the Donna and Benjamin M. Rosen Bioengineering Center [to K.K.Y.], the National Institutes of Health [F31MH102913, to C.N.B] and the National Science Foundation [GRF2017227007 to Z.W.].

Conflict of Interest: none declared.

References

Abbasi,W.A. and Minhas,F.U.A.A. (2016) Issues in performance evaluation for host-pathogen protein interaction prediction. *J. Bioinform. Comput. Biol.*, **14**, 1650011.

Alipanahi,B. *et al.* (2015) Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, **33**, 831–838.

Asgari,E. and Mofrad,M.R.K. (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLoS One*, **10**, e0141287.

Bedbrook,C.N. *et al.* (2017a) Structure-guided SCHEMA recombination generates diverse chimeric channelrhodopsins. *Proc. Natl. Acad. Sci. USA*, **114**, E2624–E2633.

Bedbrook,C.N. *et al.* (2017b) Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLOS Comput. Biol.*, **13**, e1005786.

Chang,C.C.H. *et al.* (2016) Periscope: quantitative prediction of soluble protein expression in the periplasm of *Escherichia coli*. *Sci. Rep.*, **6**, 21844.

Domingos,P. (2012) A few useful things to know about machine learning. *Commun. ACM*, **55**, 78–87.

Engqvist,M.K.M. *et al.* (2015) Directed evolution of *Gloeobacter violaceus* rhodopsin spectral properties. *J. Mol. Biol.*, **427**, 205–220.

Fox,R.J. *et al.* (2007) Improving catalytic function by ProSAR-driven enzyme evolution. *Nat. Biotechnol.*, **25**, 338–344.

Kawashima,S. *et al.* (2008) AAindex: amino acid index database, progress report 2008. *Nucleic Acids Res.*, **36**, 202–205.

Kimothi,D. *et al.* (2016) Distributed representations for biological sequence analysis. *arXiv preprint*, arXiv:1608.05949.

Le,Q. and Mikolov,T. (2014) Distributed representations of sentences and documents. *Int. Conf. Mach. Learn. ICML 2014*, **32**, 1188–1196.

Leslie,C.S. *et al.* (2004) Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20**, 467–476.

Li,Y. *et al.* (2007) A diverse family of thermostable cytochrome P450s created by recombination of stabilizing fragments. *Nat. Biotechnol.*, **25**, 1051–1056.

Maaten,L.V.D. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

Mazzaferro,C. (2017) Predicting protein binding affinity with word embeddings and recurrent neural networks. *bioRxiv preprint*, bioRxiv:128223.

Mikolov,T. *et al.* (2013a) Distributed representations of words and phrases and their compositionality. In: Burges,C.J.C. *et al.* (eds) *Advances in Neural Information Processing Systems*. Lake Tahoe, Nevada, Vol. **26**, pp. 3111–3119.

Mikolov,T. *et al.* (2013b) Efficient estimation of word representations in vector space. *arXiv preprint*, arXiv:1301.3781.

Ng,P. (2017) dna2vec: consistent vector representations of variable-length k-mers. *arXiv preprint*, arXiv:1701.06279.

Ofer,D. and Linial,M. (2015) ProFET: Feature engineering captures high-level protein functions. *Bioinformatics*, **31**, 3429–3436.

Rasmussen,C.E. and Williams,C.K.I. (2006) *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, USA.

Rurek,R. and Sojka,P. (2010) Software framework for topic modelling with large corpora. In: *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, University of Malta, Valletta, Malta, pp. 45–50.

Romero,P.A. *et al.* (2013) Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl. Acad. Sci. USA*, **110**, E193–E201.

Saladi,S.M. *et al.* (2018) A statistical model for improved membrane protein expression using sequence-derived features. *J Biol Chem.*, doi: 10.1074/jbc.RA117.001052.

The UniProt Consortium. (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, **45**, 158–169.

Young,T. *et al.* (2016) Recent trends in deep learning based natural language processing. *arXiv Prepr*, arXiv:1708.02709.

Zaugg,J. *et al.* (2017) Learning epistatic interactions from sequence-activity data to predict enantioselectivity. *J. Comput. Aided Mol. Des.*, **31**, 1085–1096.