Generative peptide embedding

Patrick Breen

Institute of Bioinformatics, University of Georgia

Abstract

With the developments in the fields of machine learning, bioinformatics and combinatorial chemistry, scientists hope to be able to search exponentially sized combinatorial search spaces to identify drug targets and aid in synthetic evolution strategies. Because of the exponential size of the molecular search space, the complex nature of the objective function, and the non-numeric representation of peptides, efficient strategies for identifying optimal drug targets is difficult. Recent work has shown the feasibility of using an unsupervised machine learning method to produce a dense low dimensional latent representation that accounts for the distribution of naturally occurring peptides. Ideally this low dimensional latent distribution also captures functional similarity and can be used to iteratively optimize a peptide solution under some objective function. In this work we present such a latent representation for peptides and discuss how our model could be used for peptide design.

Keywords: Deep Learning, Bioinformatics, Generative Modeling

1. Introduction

Traditional pharmacological drug development has focused on the production of small molecules less than 500 Da due to their high oral bioavaliability, and relatively low cost to manufacture. The biggest drawback to these small molecule drugs is their relative lack of biological specificity, leading to complex and unpredictable off-target effects. Since the production of insulin in the early 20th century, biotechnology companies have been using injectable peptides to produce theraputic effects[1]. Recently peptides have been approved, and are in wide use for rheumatoid arthritis, non-Hodgkins B-cell lymphoma and breast cancer[2]. While many peptide theraputics, sometimes called biologics, are produced through mimicry or cloning of a deficient biological product, it is feasible that existing peptide theraputics could be optimized by searching the combinatorial search space of peptides.

The design space of all peptides of length N is combinatorial in N, or approximately 20^N (since there are 20 canonical amino acids). All recorded peptides that occur in the uniprot database is a much smaller number, though still quite large, at about 6.1 million peptides between 10 and 100 residues. Combinatorial chemistry developed in the 1990s suggested that new biologics could be produced by searching the massive search space of both existing and possible peptides for a peptide with an optimal binding or therapeutic property [3]. However, such a brute force search would be costly in time and resources prompting researchers to develop heuristics and alternative strategies to make combinatorial drug search more efficient[4]. By taking existing molecules and combining them through genetic algorithms and through rational design, there

was some success at producing useful molecules in more efficient ways than brute force search of the combinatorial search space[5].

More recently, the widespread adoption of machine learning has demonstrated that deep learning can produce efficient scalable models for data representation. Early unsupervised models such as principle components analysis (PCA) which is based simple linear regression, could now be scaled hierarchically and made non-linear with the development of autoencoders[6]. Use of these unsupervised approaches could produce an efficient "compressed" intermediate data representation that could be used in a variety of subsequent analyses. In our peptide design domain this would allow finding optimal molecular configurations in a semi-supervised manner. Unsupervised learning could also be used simply to cluster similar molecules for research purposes to categorize molecules of similar properties.

A specific type of autoencoder, known as a variational autoencoder (VAE), extends the autoencoder model by adding a layer of probabilistic modeling[7]. A VAE allows one to map a given data point to a distribution that if sampled will represent a probability distribution that can map back to the initial data point. In practice, these functional mappings are implemented with feedforward neural networks that can be optimized using backpropagation. Use of a VAE instead of a deterministic autoecoder allows one to not only determine a representation, but also to sample synthetic data points (not in the training data) that are similar to a given data point. This allows a probabilistic way of exploring similar synthetic data points in the input space.

In the application of molecular modeling, a recent paper "Automatic chemical design using data-driven continuous representation of molecules"[8] used a VAE to produce a generative model of small drug molecules. The cited paper creates a generative continuous representation of small molecules for drug-design related machine learning purposes. The work in this paper will describe a similar model with a different application, namely a generative continuous representation of peptides. In addition, the sequence to sequence network that we used takes advantage of some advanced features such as an "attention mechanism" and "bucketing" (see Model and Data section for more description). We demonstrate how our model can be used as a latent representation, and discuss how it could be extended for semi-supervised peptide design.

2. Model and Data

The generative peptide representation model was implemented using the Tensorflow deep learning library[9]. The model overall is a recurrent neural network (RNN) sequence to sequence autoencoder, in which a peptide, represented as a sequence of amino acids, is encoded into a latent distribution which is then sampled to generate a dense numerical state vector that is decoded reproduce the initial peptide. The RNN was composed of gated recurrent units (GRU)[10]. By setting each amino acid equal to a unit (word) in our input and output sequence, we could think of our model as a translation model with vocabulary size of about 20 (the number of canonical amino acids). We replaced non-canonical amino acids with an "unknown" token represented as a '_' character.

Our data was acquired from uniref90, a dataset that contains proteins and peptides with length of at least 10 amino acids. From this dataset we constrained our model to peptides of less than 100 amino acids resulting in about 6.1 million unique peptides. We uniformly sampled 1,300 of these peptides as a test set, and assigned the rest as our training set. Note that our dataset, while large, is many orders of magnitude less than the approximately 20^{100} total possible possible peptides of length 100. By constraining our model to only peptides that are found in nature

we heavily bias our model to represent only a small fraction of the combinatorial search space. Furthermore,

Both the training and testing peptides were ”bucketed” into 4 buckets based on peptide length. The number of resides per bucket were 10-40, 40-60, 60-80 and 80-100. Peptides falling in separate buckets have some bucket-specific model parameters and some some shared model parameters. The bucket-specific parameters allow the model to handle shorter peptides differently from longer peptides. For example, short peptides often do not start with a methionine, while almost all longer peptides start with a methionine. Shared parameters capture structure that is not specific to bucket length. The model also used attention and embedding mechanism, extensions to RNN's that have demonstrated success in learning complex patterns in sequences[11].

The model was trained as a conditional RNN[12], in which the output of each cell in the decoder is fed as an input to the next cell in the decoder. The model was evaluated as an unconditional RNN in which the decoder received only the state from the latent distribution See Figure 1 for a diagram of the model.

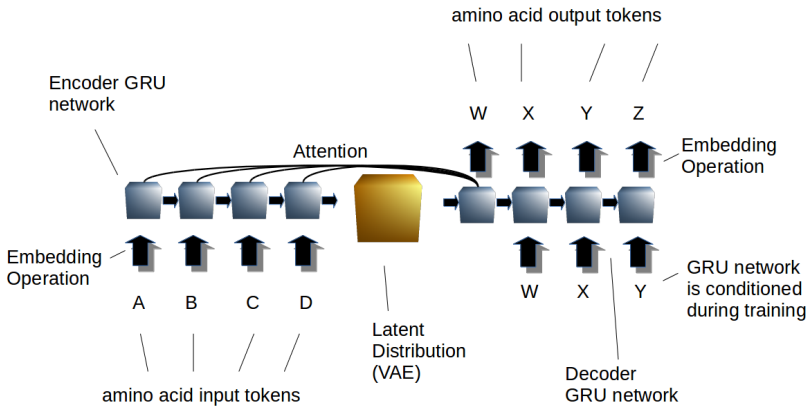

Figure 1: Diagram of sequence to sequence peptide model.

The objective function used to train our model is a sum of the reconstruction loss, the cross entropy between the input and output sequences, and the latent loss, the KL-divergence between the latent distribution and the unit Gaussian (both measured in log-perplexity). By adding the latent loss together with the reconstruction loss, the VAE is fit to the data while also optimizing the regularity of the latent distribution to prevent overfitting. The reconstruction loss was defined as the weighted cross entropy for a sequence, a loss function that has worked well in English to French translation, as well as other general sequence to sequence applications[13]. However, despite these successes, it remains to be seen if cross entropy is the most appropriate loss function for peptide sequences, which may be unique in their specific requirements, such as needing explicit handling of insertions and deletions. A loss function based on the edit distance between two peptide sequences may more explicitly handle these phenomena, though it would be more costly to compute slowing overall training speed.

3. Results

We trained our model on the training set and recorded the loss function averaged over each sequence-bucket. We recorded the the reconstruction loss and the latent loss over the course of training. As you can see the reconstruction loss decreased over the course of the optimization (Figure 2), however in many batches, even after extensive training, the latent loss represented only a small fraction of the overall loss. This may imply that the model is under-fitting the data. Additional tuning of the model's hyper parameters could potentially improve the model's performance.

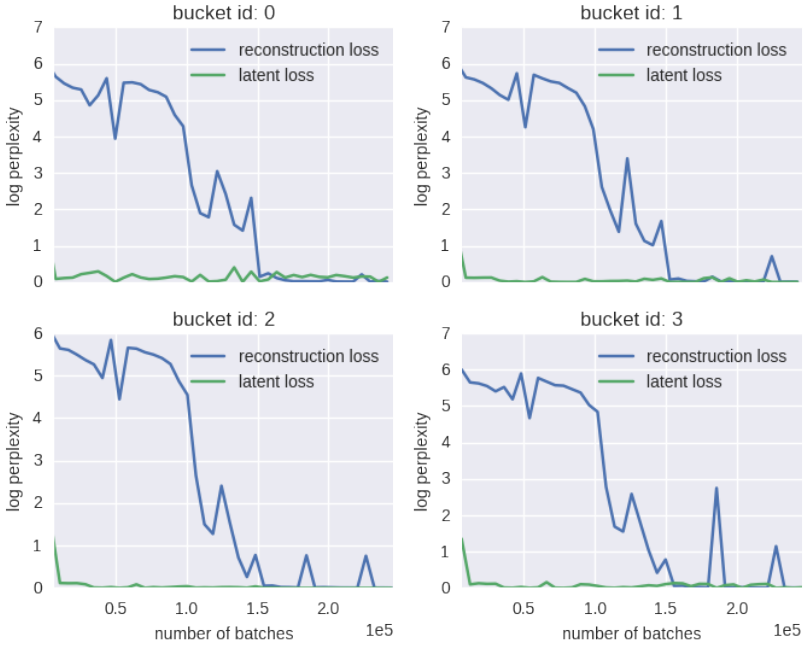

Figure 2: Reconstruction loss and latent loss, as a function of number of gradient decent batches.

Because a VAE is stochastic, the output is non deterministic. In Table 1 we show example input peptides from our testing set and corresponding outputs fed through the trained model. As you can see, the outputs approximate the inputs, with varying levels of insertion, deletion and substitution relative to the input. Formally the VAE decoder defines a distribution of outputs x given an input z, $p_{model}(x|z)$. This allows us to use our trained model to sample "similar" peptides to a given peptide. We can use this distribution to perform a biased search through peptide design space as part of a peptide design proceedure (see section: Proposed exploration of peptide design space).

3.1. Use as Representation

The most straightforward way that this unsupervised model can be used, is as a dense low dimensional representation of peptide sequences. This is significant, since variable length peptides are hard to input into traditional data analysis algorithms that expect a dense numerical vector

Table 1: Three peptides from the testing set were fed through the model and the output was sampled three times for each input peptide. Note that the output is stochastic, showing insertions, deletions and substitutions relative to the input.

input 1	MAEYGEKYAEPLISEYALRRAF_EG
output 1a	MLGEEQKYAEPLISVKLDPAAFTSEG
output 1b	MLGEAEILILPLISVKLDPATSEG
output 1c	MLLDGEKYAEPLISRELDPAAFTSEG
input 2	MDQVSRDLAFRVRVRATQVRYEKEMKIKFRKQ
output 2a	MLQVSRGQLLRVRVRAVRVREEKEIKKQ
output 2b	MLQVSRGQLLRVRVRAVRVRKEKEIK
output 2c	MLLLLLLLRVRVRVRVVRVREEKEIK
input 3	MMLNELIATAIGEVGIAWFDFYSIGTSALGLDYFYSIFLLFF
output 3a	MSKDIAIATAIGEVGIFDFDFDSIGTSASALDATIFIFLLLL
output 3b	MKKDIATAANGIGIGIFDFDFDSIGTSAGVLDATIFIFLLLL
output 3c	MSLNLNLLTAEVGIFDFDFDFDSIGTSALGLDATIFIFLL

input. After training our model, all peptides can be represented by their latent state, allowing the peptides to be fed into further data analyses. We visualized the distribution of the numerical representations of the peptides in our test set by mapping the 28 dimensional representation vectors into 2 dimensions using tSNE, and plotting the resulting 2 dimensional representation of the peptides. We colored the 2 dimensional peptide representations by sequence length (Figure 3), and average amino acid hydrophobicity (Figure 4). There is a clear clustering of peptides by sequence length, and also a weaker clustering of peptides by average hydrophobicity (hydrophobicity represented by Kd). These results give some evidence that our model, which is only optimized to capture sequence similarity, is also capturing biological functional properties. The similarities encoded in this embedded representation can be exploited in a downstream supervised machine learning application.

3.2. Proposed exploration of peptide design space

Finally, we propose an optimization strategy could naturally take advantage of the distribution encoded in our model by using a biased Metropolis-Hastings-like search strategy (see Algorithm **??**). Recall that due to the stochastic nature of the VAE, for a given input peptide y, we can generate an output distribution of peptides x that are related to the input peptide, $p_{model}(x|y)$. Assume we have an objective function for how well a given peptide performs at binding or for drug treatment outcome, $o(x)$. This objective function could be filled with local optima, not differentiable, and expensive to sample (requiring external experiment or molecular simulation). This rules out gradient based methods, and makes sampling $o(x)$ expensive. Given these constraints, we could construct an optimization strategy of this function potentially more effective than Monto Carlo or exhaustive search strategies (since we are using the training data to bias our search), if we use our proposed model as the "proposal distribution function" in a Metropolis-Hastings-like optimization. Such a proposed application would be similar in theory to previous work, but with a different proposal distribution function[14]. It is important to note that this while we could describe this search strategy as Metropolis-Hastings-like it is not equivalent, and does not inherit any guarantees on performance or guarantees of finding the global optimum.

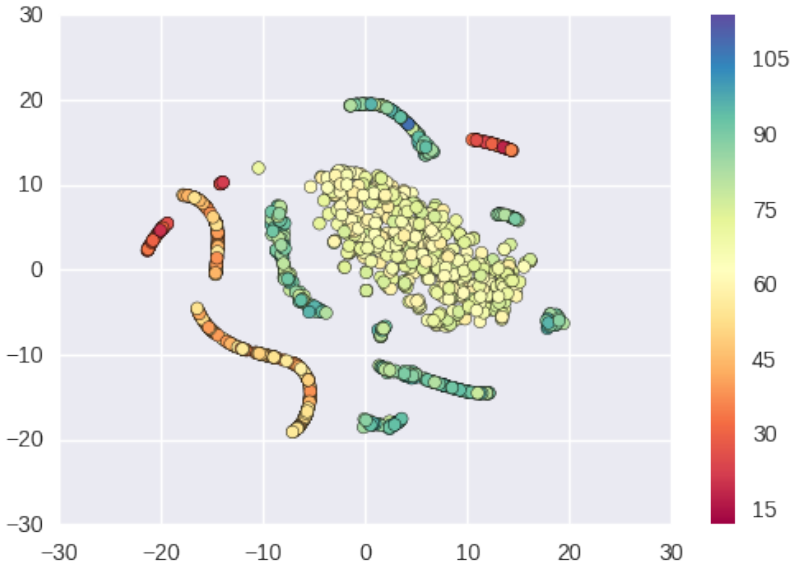

Figure 3: Visualization of latent space for the test set colored by sequence length.

Because this strategy for "synthetic peptide design" would require an experimentally relevant objective function defined on the entire space of possible molecules, this proposal, while facilitated by the distribution produced in this paper, would require substantial experimental resource expenditure to evaluate empirically.

4. Discussion

In summary we have constructed and trained a representation of peptides using an unsupervised sequence to sequence VAE. By using this representation we showed that peptides with similar biological function cluster close together in latent space, though the model was trained on biological sequences alone. Thus we can represent the biological properties of the millions of peptides and proteins stored in public database repositories in a cheap unsupervised manner without having to quantify or label their precise biological attributes through external experimentation. This allows future researchers to take our dense numerical representation and in conjunction with labeled data, to perform conventional semi-supervised learning for peptide drug discovery or more generally for some optimization search under a given objective function. We have also proposed a downstream sampling method that leverages our VAE model to explore peptide design space for increasingly optimal synthetic peptides.

[1] C. Weyer, C. Bogardus, D. M. Mott, R. E. Pratley, The natural history of insulin secretory dysfunction and insulin resistance in the pathogenesis of type 2 diabetes mellitus, The Journal of clinical investigation 104 (1999) 787–794.

[2] D. J. Craik, D. P. Fairlie, S. Liras, D. Price, The future of peptide-based drugs, Chemical biology & drug design 81 (2013) 136–147.

[3] R. S. Youngquist, G. R. Fuentes, M. P. Lacey, T. Keough, Generation and screening of combinatorial peptide libraries designed for rapid sequencing by mass spectrometry, Journal of the American Chemical Society 117 (1995) 3900–3906.

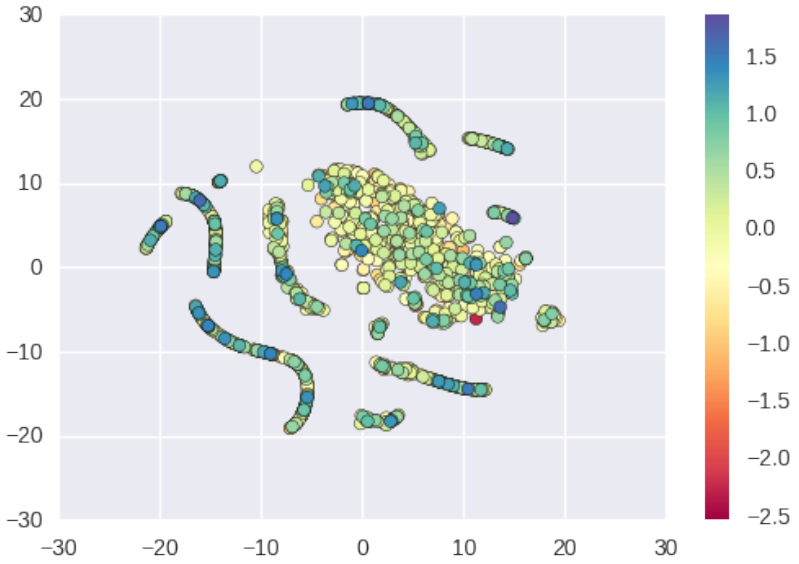

Figure 4: Visualization of latent space for the test set colored by average amino acid Kd (hydrophobicity).

[4] M. Feher, J. M. Schmidt, Property distributions: differences between drugs, natural products, and molecules from combinatorial chemistry, Journal of Chemical Information and Computer Sciences 43 (2003) 218–227.

[5] I. Belda, S. Madurga, X. Llora, M. Martinell, T. Tarragó, M. G. Piqueras, E. Nicolás, E. Giralt, Enpda: an evolutionary structure-based de novo peptide design algorithm, Journal of Computer-Aided Molecular Design 19 (2005) 585–601.

[6] G. E. Hinton, R. S. Zemel, Autoencoders, minimum description length, and helmholtz free energy, Advances in neural information processing systems (1994) 3–3.

[7] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[8] R. Gómez-Bombarelli, D. Duvenaud, J. M. Hernández-Lobato, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, arXiv preprint arXiv:1610.02415 (2016).

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[10] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555 (2014).

[11] J. Chorowski, D. Bahdanau, K. Cho, Y. Bengio, End-to-end continuous speech recognition using attention-based recurrent nn: first results, arXiv preprint arXiv:1412.1602 (2014).

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[13] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, G. Hinton, Grammar as a foreign language, in: Advances in Neural Information Processing Systems, pp. 2773–2781.

[14] S. Giguere, F. Laviolette, M. Marchand, D. Tremblay, S. Moineau, É. Biron, J. Corbeil, Improved design and screening of high bioactivity peptides for drug discovery, arXiv preprint arXiv:1311.3573 (2013).