What is the name of your organization?
University of Toronto, the Vector Institute for Artificial Intelligence, and NYU
What is the best email address to contact you?
quaid.morris@gmail.com
What is the project?
We are trying to decode the genome. DNA is written in a language that we don't understand. If we could understand that language, we would not only be closer to understanding how cells work but we would also be able to understand how defective genes cause disease and cancer. Both of these insights would help us design better medicines and treatments.

The "genetic code" that was discovered in the 60's covers less than 2% of the genome, this project is about translating the rest of the genome, or at least the 10% of it that seems important. This part of the genome is called the "regulatory code" because it contains instructions to the cell that regulate when genes get turned off and on, and how active a gene should be in a cell.

Specifically, we will be analyzing the molecular machines (i.e. proteins) that recognize words of the vocabulary of the regulatory code. The overall goal is to determine what the vocabulary of the code is, and which proteins recognize which words. These words are known as "motifs".

To do this, we will be training types of deep neural networks called autoencoders to map from the amino acid sequence of a protein to a latent, real-valued representation of that protein. We will then use these latent representations to predict what DNA (or RNA) motifs a protein recognizes. If we could succeed at this task, we will have uncovered the vocabulary for the whole tree of life, at least the eukaryotic part of it.
What does the data look like?
We will be using largely publicly accessible data. Most of the data is sequences of letters.

The input to the algorithm will be protein amino acid sequences -- these are hundreds of thousands of sequences of different lengths of letters from a 20-letter alphabet. These sequences are readily available in online, public repositories.

We will also use information about orthology, i.e. which proteins in related organisms likely recognize the same words. This data is in an adjacency matrix format and is also publicly available.

Finally, we will use information about what words some of these proteins recognize. These data are hundreds of vectors of real numbers, each element of each of these vectors represent how strongly a particular protein recognizes a particular word. Much of this data is publicly available, but we might validate the results on unpublished data.

There will be some minor effort in collecting this data from the public repositories but this will not be hard or time-consuming

What is the proposed scope of the projects?

This project requires 1-3 people. This is a well-delimited problem. It should require a few months work. With more people, and with more bioinformatics expertise, we will evaluate more informative representations of the protein sequence -- e.g. using various tools to annotate the protein sequence according to predicted secondary structure, solvent accessibility, or even homology modelling.

What are the rubrics of success?

We have other methods to predict what DNA or RNA words a protein recognizes. These methods are fairly basic, and we suspect that we can do better. Success would be improving on those methods. Success could also be trying hard to solve the problem using an autoencoder, failing to do so, and understanding why the autoencoder is no better than competing methods.

What is the relevant background needed with respect to the project?

A basic background in biology would be extremely useful. A stronger background in bioinformatics would be wonderful. Some of this background can be gained during the project, it is not critical. But as I indicated above, the more background the student(s) have, the more interesting features we can use.

At minimum, students need to be comfortable with training deep neural networks, or at least understand the concept and be able to use any of the software packages (e.g. Keras, TensorFlow, Theano) for doing so. Probably, a good solution would involve some sort of convolutional neural networks and possibly even some type of recurrent neural network. It would help to understand those concepts as well.

Obviously, students need to be able to program. They should be willing to make figures too.

What is the relevant organizational, project or institutional history?

University of Toronto is a research university. This project would be a good introduction to machine learning research in biology. The Vector Institute for Artificial Intelligence is a not-for-profit NGO in Toronto where a number of good machine learning and deep learning researchers work.
How will the organization support and mentor the students?
Professor Quaid Morris is in NYU for a sabbatical. He will directly mentor the team, along with the help from his graduate students.

Professor Morris is a well-known machine learning researcher in computational biology. He trained in ML at MIT and the Gatsby Unit and worked on deep learning for medical diagnosis in his PhD thesis.
Additional Feedback?

Team Viserion: Project Update

Predicting RNA-Binding Preferences for Proteins

Shasha Lin, Julie Helmers, Millie Dwyer, Lihan Yao

November 9, 2017

1 Research Question

RNA Recognition Motifs (RRMs) are substructures of RNA-binding proteins that specify how the protein-RNA binding will occur. For the majority of identified RRMs, it is unknown which RNA sequences they bind to, out of tens of thousands of possibilities. However, there are many RRMs that are homologous, meaning that they share sequence patterns as well as binding behaviors. If it is possible to generate a representation of these similarities and relate this representation to binding preferences, it may be easier to tell how a newly identified RRM will behave. This approach would be faster and less expensive than experimentally determining behavior for individual RRMs. From a broader perspective, an informative RRM representation will lead to more accurate predictions of RNA binding preferences, and may help researchers to better understand RRMs' role in gene expression. Through this capstone project, our goal is to learn reliable and predictive low-dimensional representations of RRMs through deep learning techniques in order to predict RRM RNA-binding preferences.

2 Data

Due to the small number of known binding matches between RRMs and RNA, our RRM sequence data is broken into labeled and unlabeled datasets, pulled from different sources. In general, the RRM sequences are combinations of 20 letters, each of which represents a different amino acid, and are at maximum around 90 letters long.

2.1 Unlabeled Sequences

Identifying RRM sequences is easier than identifying which RRMs will bind with which RNAs, giving us a large unlabeled dataset. We specifically have approximately 100k sequences with unknown binding preferences. This unlabeled data is pulled from PFAM [1], which gives aligned RRM sequences of length 550. Each datum consists of an ID/Name and Sequence. The ID/Name matches generally matches the name of the protein associated with the RRM.

2.2 Labeled Sequences

For a small number of RRM sequences, the distribution of binding affinity over RNA sequences is known for the associated proteins. These matches are generally identified through manual testing, and known for approximately 200 RRM-containing proteins. Biologists have identified that RRM sequences with at least 70% sequence similarity have nearly identical binding preferences, which expands this "labeled" dataset to approximately 7k pairs. Our labeled dataset was provided by our advisor's lab at the University of Toronto, and includes protein sequences and z-score distributions over RNA sequences. These related affinities were generated using the RNAcompete method [2] defined by Ray et. al. Because this dataset is provided from a different source than the unlabeled data, we have run into some data processing issues when connecting our modeling methods. We discuss this in the following two sections as well as in the discussion below.

2.2.1 Data Processing

To generate an informative latent representation of the unlabeled sequences, we proceed from our raw data by filtering to exclude any positions that contain gaps in all of our sequences, then filtering to exclude sequences which still contain a significant percentage of gap positions in the remaining positions. A gap position is denoted by the symbol "_", noting the absence of an amino acid. In the first step, our sequences of length 550 are shortened to length 81 by eliminating the same gap positions uniformly across our dataset. This drastic decrease in length is due to the nature of alignment data: A majority of positions in the center of all sequences contain no meaningful symbols due to the PFAM collection process. See appendix for the distribution of symbol occupancies by position. Now, in this shortened length 81 dataset, we remove sequences for which less than 70% of the remaining symbols are non-gaps. We believe learning a representation while including sequences with significant numbers of gaps would not only introduce marginal information into the representation, but deteriorate overall representation quality. By setting the symbol occupancy percentage threshold at 70%, we filter out 41.5% of our dataset, with 58,500 sequences remaining. See appendix for the distribution of symbol occupancies by sequences.

2.3 Sequence Alignment

Because RRM sequences are not consistently of the same length, it is possible that learning a representation without aligning the sequences would incorrectly capture illusory local structure across RRMs. Sequence alignment provides organization across RRM sequences of different lengths, by aligning amino acids into similar positions. This allows for gaps to be placed in the sequences when certain RRMs are missing a given positional amino acid. The most common method of alignment uses hidden Markov models (HHM) to fit a sequence of amino acids to a profile. The alignments differ between our unlabeled PFAM and protein-labeled datasets, so some adjustments are needed for our unsupervised and supervised methods to cooperate. More details of the challenges of our different sequence alignments are included

in the discussion section.

3 Methodology

Because we hope to utilize our larger dataset of unlabeled data, we approach this problem with unsupervised methods. This problem resembles results found in Natural Language Processing (NLP) tasks, where training generative models learns a low-dimensional embedding of words where distance resembles different semantic properties.

3.1 Baseline Model: Seq2vec

Seq2Vec [3] is a Doc2Vec application to biological sequences. It generates a distributed representation where each protein sequence is embedded according to its document summary vector in a context prediction task. A natural ambiguity, and where the analogy between our domain and NLP weakens, is the choice of "words" in a protein sequence, which contains no inherent segmentation between its symbols, or a clear segmented structure. However, it is common practice in the literature to use a sliding window of width 3, where at a given position $[w_i, w_{i+1}, w_{i+2}]$, the window contents form a 3-symbol "word" which is then embedded. Our implementation follows this practice, and places resulting 3-mers into a Doc2Vec model via the Gensim package.

3.2 CNN + LSTM

To utilize modern NLP methods, we have used a CNN encoder with an LSTM decoder to learn the latent space representation of the RRM sequences. In the original proposal and early conversations with our advisor, we considered using PCA methods in addition to the CNN and LSTM. The goal of the PCA step would be to capture global structure in the sequences. However, after further discussion we decided to omit the PCA, as layers in the CNN or LSTM can achieve the global results as needed. We have since approached this project using two different CNN+LSTM architectures: a character-level autoencoder and a ResNet implementation.

3.2.1 ResNet Autoencoder + LSTM decoder

A CNN encoder + LSTM decoder architecture has been applied in image captioning settings [4]. In the image captioning task, the CNN is used to extract high-level features from images. Those features are then used as the context vector for a downstream language modeling LSTM, which generates target words throughout time conditioned on both the image context vector and the sequence prior to the current word. We take the same architecture and apply it to our RRM modeling: RRMs are first run through a 50-layer ResNet CNN, the output of which is fed into a downstream LSTM, which uses the CNN features as context for modeling the original RRM sequence. The main difference between our task and the image captioning task is that the input for the CNN and LSTM comes from the same data source

in our case: an RRM sequence, whereas they come from the image and from a human-generated caption, respectively, in the original task. The justification that we think this adapted architecture will work is that the LSTM will learn to model RRM sequences in general (i.e. learning the statistical mapping that generates RRMs), whereas the CNN will learn local structure that is specific to each particular RRM sequence, helping the LSTM model predict the next amino acid letter based on the case of the current RRM in particular. We will use the hidden vector generated at the last time step of the LSTM as the learned representation for each sample, as this is the representation that not only takes local features into consideration (as the CNN encoder does), but also learns the sequential information (as the LSTM does). We are still developing this model, and predict some challenges to arise with our adaptation:

1. We have to be careful not to overfit.
2. We need to be practical about training time: Both the CNN encoder and LSTM decoder may take a long time to train.

We plan to train this model in the week following the submission of this report, and provide an update on the feasibility/performance of this model as soon as possible.

3.2.2 Character-Level Autoencoder

A character-based autoencoder such as [5] may be generalized to encode symbol sequences with character level morphemes and with no knowledge of a vocabulary. It learns feature maps corresponding to biological motifs (typically local features of a protein sequence that are 3 to 8 amino acids in length), then at the LSTM layer, encodes the entire sequence according to its motif activation pattern. The long-term dependencies extracted by the LSTM layer may exploit higher-dimensional structures of the protein sequence, i.e. alpha helices and beta coils intertwined with the RRM, providing its 3-dimensional shape. These structures appear among sequence symbols with semi-periodicity and over long distances, for which the LSTM is best suited.

3.3 Affinity Regression

Affinity regression is a supervised learning method developed by Pelossof et. al. [6] to learn binding relationships between proteins and RNAs through a bilinear function. Because the learned association is a linear relationship between the RRM and RNA sequences, applying affinity regression with a low-dimensional representation of RRMs instead of the original sequences will reduce the number of weights included in this model, allowing better results with our relatively small labeled dataset. For our project, we plan to use the representations learned by our deep learning models to improve affinity regression performance for RNA-binding prediction on the labeled data, and, more interestingly, to make meaningful predictions for unlabeled data. So far, we have implemented affinity regression and are working on matching the alignments between our labeled and unlabeled datasets to get substantive results.

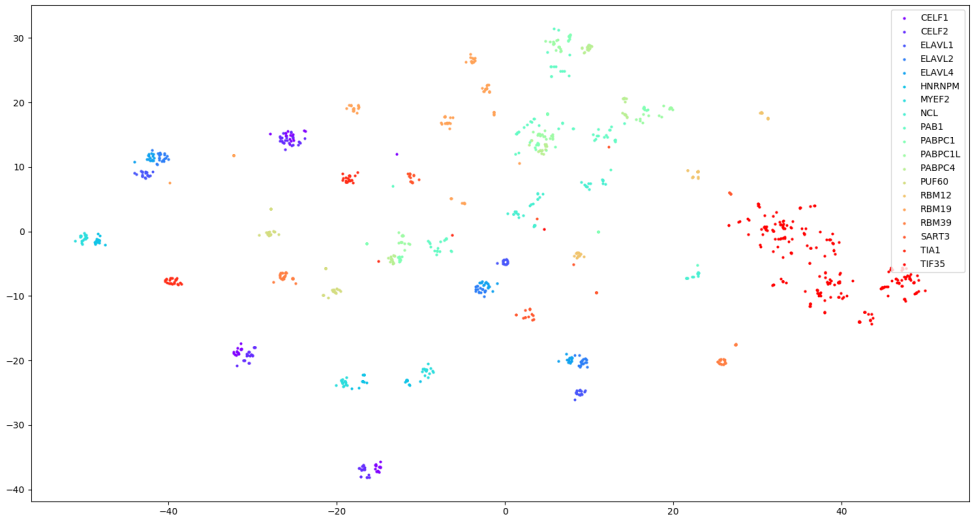
4 Results

4.1 Evaluation Methods

As discussed previously, our high level goal is to learn reasonable latent representations of RRMs. To that end, we judge our results based on three different metrics. First, we look to the reconstruction error of our encoder/decoder models, which provides a natural way to evaluate the reasonableness of our representations. Second, to judge how well our learned latent representations capture true similarity between RRMs, we look at similarity in representation and similarity in predicted binding behavior (using affinity regression) within homologous RRMs (as identified by gene symbol). Third, we evaluate affinity regression, as applied to the low-dimensional representations, on our labeled data based on the correlation between predicted binding intensity & experimental binding intensity (the “ground truth").

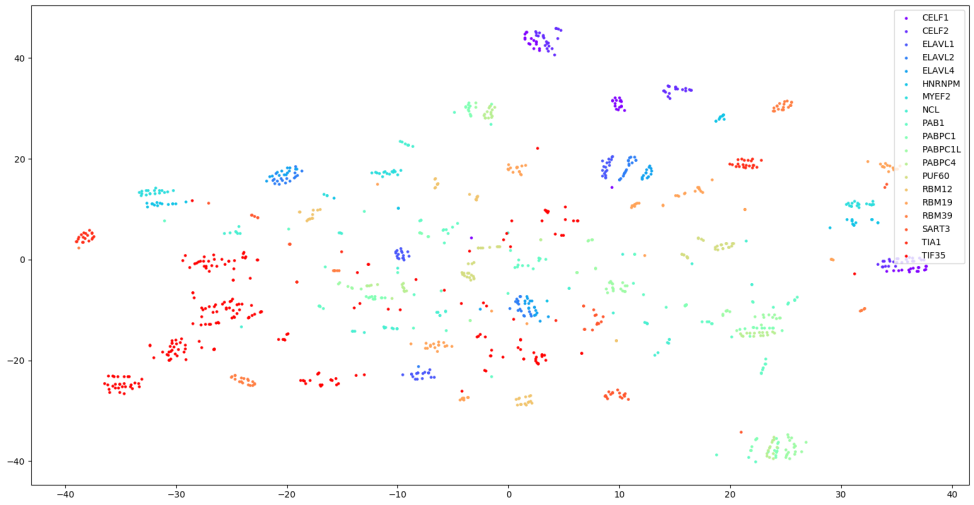
4.2 Similarity Results

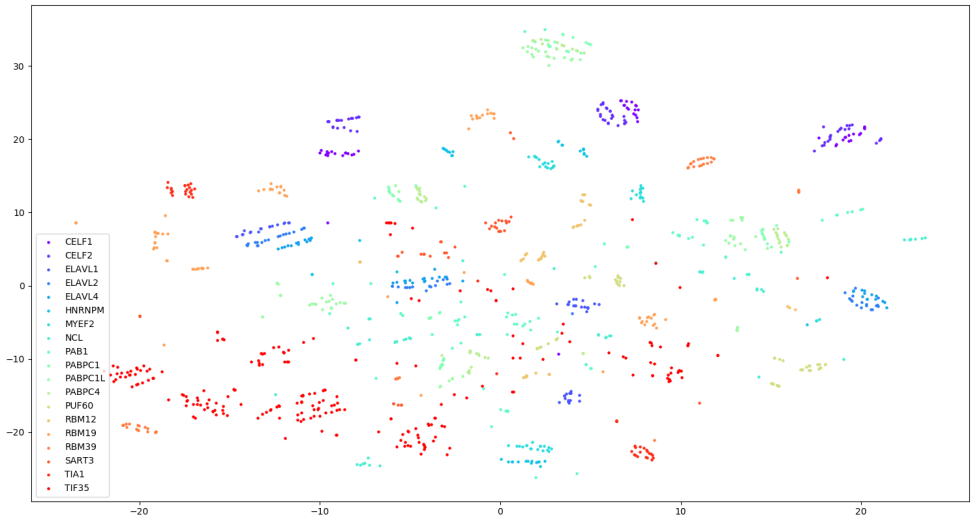
We generated a Seq2Vec distributed representation as well as the latent representations from our CNN+LSTM models. We then mapped the RRM IDs to the corresponding gene symbols to help us understand which RRMs should be represented similarly. In the tSNE projections below, a point is colored according to its gene identification, where certain color groupings correspond to functionally similar gene identifiers (such as CELF1 and CELF2). In the Seq2Vec distributed representation, we observe distinct clustering according to gene identifiers. This indicates that the high-dimensional representation, from which the plot is created, does contain important information for our downstream task of affinity regression.



As a preliminary autoencoder to compare with Seq2Vec, while we continue work on our more sophisticated CNN+LSTM architectures, we extracted the latent representation from an encoder of one convolutional layer followed by pooling, and a decoder of an unpooling layer followed by one linear layer. The following tSNE projections have respective perplexities of 40 and 80 (for lower perplexity, global

features become less prominent):





Taking into consideration tSNE's inherent stochasticity and the high dimensionality of our data, our analysis of these projections will be limited. However, we notice that the simple autoencoder has difficulty encoding certain TIF35 sequences, in contrast to Seq2Vec's clean separation. This is evidence that a more nuanced algorithm, such as the ones we are already developing, is needed.

5 Discussion

5.1 Data Challenges

One of the main challenges we have faced with our dataset stems from the sequence alignment difference between our labeled dataset and our unlabeled data. Before we can run affinity regression, which is a

supervised algorithm, we need to match the alignments to generate latent representations for the RRMs in our labeled dataset. We lack substantial domain knowledge about RRM sequence alignments, so it has been difficult to make sure that the resulting alignment is correct, aside from the lengths of the sequences. This is a problem we are still tackling with the advice of our advisor and his lab members. As a result, we do not yet have meaningful affinity regression results.

Another challenge we encountered is that to recognize homologous RRMs within our unlabeled training set, we needed to match the associated protein names with gene symbols. Our datasets map RRM sequences to protein names, but do not additionally include the gene identifier. We used a third-party tool to provide this necessary mapping. [7] Broadly, RRMs associated with the same or similar gene symbols are likely to be homologous, but this is not always the case. Without domain knowledge for each specific gene symbol, our identification of homologous RRMs may be incorrect in places, but we have at least been able to approximate the true underlying relationships.

5.2 Improvement of Current Models

We have already built several encoder/decoder models, which would benefit from additional parameter tuning and/or variations in architecture. We plan to continue to improve our models as we get affinity regression results.

5.3 Further Architectures

Our current models have relied on an alignment to feed RRM structure into the model, meaning all inputs need to conform to a fixed length and matching alignment. We plan to explore some additional models that use this type of input, as well as some more RNN-based models that can handle variable length sequences without alignments. Removing this alignment will make the process of generating low-dimensional representations easier and more flexible.

To explore representations that do not rely on alignments, we plan to look at LSTM-to-LSTM encoder/decoder models. We hope that these will learn reasonable low-dimensional representations, allowing association of homologous RRMs without needing to match alignments. We plan on basing these models on existing sequence-to-sequence translation models, particularly those used by generative translation models [8]. We also plan to try a model that uses our Seq2Vec results as a basis, since those seemed to perform as desired in tSNE projections, but with an additional LSTM [9].

5.4 Data Topology

From our preliminary examination of the simple autoencoder results, we may interpret the topological structure in our high-dimensional data as part of the signal to be encoded. If time allows, we propose to compute our data's persistent homology: topological features which are robust to noise. Each data point will be augmented with an extra symbol denoting the topological component containing it. This

is motivated by distinct, nonlinear structures observed in our latent representation (displayed as low-dimensional tSNE projections): clusters, chains, and torii that delineate a RRM's gene identifier.

References

[1] R. D. Finn, P. Coggill, R. Y. Eberhardt, S. R. Eddy, J. Mistry, A. L. Mitchell, S. C. Potter, M. Punta, M. Qureshi, A. Sangrador-Vegas, G. A. Salazar, J. Tate, and A. Bateman, "The pfam protein families database: towards a more sustainable future," *Nucleic Acids Research*, vol. 44, no. D1, pp. D279–D285, 2016.

[2] D. Ray, H. Kazan, K. B. Cook, M. T. Weirauch, H. S. Najafabadi, X. Li, S. Gueroussov, M. Albu, H. Zheng, A. Yang, H. Na, M. Irimia, L. H. Matzat, R. K. Dale, S. A. Smith, C. A. Yarosh, S. M. Kelly, B. Nabet, D. Mecenas, W. Li, R. S. Laishram, M. Qiao, H. D. Lipshitz, F. Piano, A. H. Corbett, R. P. Carstens, B. J. Frey, R. A. Anderson, K. W. Lynch, L. O. F. Penalva, E. P. Lei, A. G. Fraser, B. J. Blencowe, Q. D. Morris, and T. R. Hughes, "A compendium of rna-binding motifs for decoding gene regulation," *Nature*, vol. 499, pp. 172 EP –, 07 2013.

[3] D. Kimothi, A. Soni, P. Biyani, and J. M. Hogan, "Distributed representations for biological sequence analysis," *CoRR*, vol. abs/1608.05949, 2016.

[4] "Imagecaptioning." https://github.com/yunjey/pytorch-tutorial/tree/master/tutorials/03-advanced/image_captioning. Accessed: 2017-11-09.

[5] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *CoRR*, vol. abs/1610.03017, 2016.

[6] R. Pelossof, I. Singh, J. L. Yang, M. T. Weirauch, T. R. Hughes, and C. S. Leslie, "Affinity regression predicts the recognition code of nucleic acid–binding proteins," *Nature Biotechnology*, vol. 33, pp. 1242 EP –, 11 2015.

[7] "Uniprot gene mapping." http://www.uniprot.org/uploadlists/. Accessed: 2017-10-30.

[8] "Pytorch seq2seq tutorial." http://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html.

[9] X. Min, W. Zeng, N. Chen, T. Chen, and R. Jiang, "Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding," *Bioinformatics*, vol. 33, no. 14, pp. i92–i101, 2017.

Appendix: Further results

Figures mentioned in the data processing section.