GeneGPT-2: Predicting differential gene expression from histone modifications with GPT-2

Alex Ruan
alexruan@umich.edu

Abstract

Isolating the histone modifications that affect chronic inflammation in those with diabetes is a pressing problem within healthcare. One way to tackle this is through the creation of a GPT-2 model that predicts differential gene expression from histone modifications. This approach hasn't been taken before, likely as GPT-2 is traditionally used for language tasks. However, as DNA follows semantic strctures as well as it is the "language of the genome", and as there is some promising results from prior work, there is some merit in exploring this approach. Through this project, we achieve correlations around 0.55 on a histone modification/differential gene expression dataset that the state-of-the-art model, DeepDiffChrome, has achieved around 0.6 on. This shows that an approach with GPT-2 is viable, and is worth further exploration in the future.

1 Introduction

In the United States, roughly 80,000 amputations occur due to non-healing wounds as a result of chronic inflammation from diabetes. With a 20-50% three-year mortality rate even after these amputations, the fact that there is no effective therapeutic approach to treat diabetes-induced chronic inflammation needs remedying.

To identify a therapeutic approach, we need to understand the underlying mechanism behind diabetes-induced chronic inflammation. An increasingly popular approach is to analyze histone modification patterns. Histones are proteins that DNA wraps around, causing certain genes to be expressed more the tighter the DNA is wrapped, and less expressed the looser the DNA is wrapped. As we go through out lives, our genome (DNA) remains the same, but our epigenome (histone modification patterns) change based on life experiences and conditions, changing the degree to which different genes are expressed. Those with diabetes manifest a different histone modification pattern than those without, causing differential gene expression. If we can identify which histone modification patterns cause increased expression of inflammatory genes in those with diabetes, we can identify contributing factors to chronic inflammation, and then follow up with further studies isolating both the modifications and the genes they are modifying to identify the broader biological mechanism they influence. These mechanisms can then be targeted with specialized epigenetic medicine to treat diabetes-induced chronic inflammation.

A great deal of work has been done in this area utilizing a wide range of techniques from bidirectional LSTMs (Sekhon et al., 2018) to HMMs (Xu et al., 2008) to try and predict which modifications influence inflammation.

An approach that hasn't been attempted yet however, is the creation of a GPT-2 model trained on histone modification patterns rather than human language. This likely is because GPT-2 has traditionally been used with human language, and its applicability in other domains is shaky.

However, given that histone modifications are based on the underlying DNA they are modifying, and DNA is the language of the genome, it also follows semantic rules as well. In particular, there are many "parts of speech" within DNA, that determine how they start and end (promoter regions), how they modify what they are describing (enhancer regions), and how different sections are connected. As a result, we believe there is merit in exploring the training of GPT-2 on histone modification patterns for predicting differential inflammatory gene expression from histone modifications.

2 Approach

This project seeks to use a state-of-the-art transformer-based architecture to try to predict differential gene expression based on histone modification patterns. The current highest performing architecture is GPT-3, however GPT-3

is only available via an API from OpenAPI. To train from scratch, a publically available architecture must be used, such as GPT-2.

This project is implemented by first extracting histone modification pattern datasets from REMC. Then these patterns are preprocessed into a form trainable with GPT-2. In GeneBERT mentioned below, an approach was used in which numerical values were divided into values of high expression, moderate expression, and low expression, allowing the dataset to be represented in a sentence-like form where characters represented expression levels, words represented sections, and sentences represented cells. The same approach is taken here in which numerical values are converted into letters, which can form words, sentences, and paragraphs that describe the genome.

After the datasets are preprocessed into a form GPT-2 can process, the model is trained on these histone modification patterns from scratch. Finally, the word embeddings from the trained model are then extracted, and then the preprocessed histone modification patters for cell pairs under two different conditions are then tokenized using GPT-2's BPE tokenizer. These tokens are then mapped to their word embeddings, and then fed into an XGBoost Regressor model for the task of predicting differential gene expression with the goal of predicting differential gene expression based on their varying histone modification patterns. The trained model's results are then compared against the DeepDiffChrome results as a baseline.

3 Previous Work

3.1 DeepDiffChrome

DeepDiffChrome (Sekhon et al., 2018) is a deep-learning framework that seeks to understand how different histone modifications interact with each other across the genome, as well as how they directly affect differential gene expression across two different cell types for the same cell. It does this by using bi-directional LSTMs to encode the histone modification patterns and then uses an attention layer in order to provide weights for how important different types of histone modifications are to differential expression for certain genes. This project used data from the most popular database for epigenetics projects, the Roadmap Epigenomics Project (REMC), using data for ten pairs of different cell types. DeepDiffChrome significantly outperformed state-of-the-art baselines for prediction of differential gene expression.

3.2 CornBERT

Transformers are normally used for NLP applications with human language, however transformer based architectures such as BERT have been used for genomics before. Specifically, a BERT model known as CornBERT was built specifically for the task of predicting gene expression in maize. (Xu, 2020). This model predicted gene expression in ten different corn tissues given portions of DNA known as regulatory (promoter) sequences as input. The usage of transformer-based architectures for predicting gene expression in the past, suggests that usage in other areas of genomics is viable as well.

3.3 GeneBERT

Last semester for EECS 498-5 (Emotive Machine Learning), I took an approach in which BERT was trained from scratch on histone modification patterns with the task of predicting differential gene expression from histone modifications. The model was trained on the same dataset DeepDiffChrome used, and achieved a correlation of 0.3 between the predicted results and ground-truth results. While this was much lower than DeepDiffChrome (0.6), given that it was trained in highly sub-optimal conditions (trained on only 27mb worth of data, sub-optimal data preprocessing, usage of XGBoost rather than an additional fully-connected layer on top of BERT, and no fine-tuning for the specialized task), it showed that there was merit in using a transformer-based architecture in order to encode a representation of histone modification patterns on DNA to predicting differential gene expression.

4 Data

To train, validate and test our model, we used the same dataset as the authors of DeepDiff, which enables us to directly compare the performance of the model to the state-of-the-art values achieved by DeepDiff. (Sekhon et al., 2018) The data consists of two things:

- The first is an input matrix of the five core histone modification read counts from the REMC database. A row of the dataset consists of the counts of the five modifications over a span of 100 base-pairs. As a result, our data gives us the number of each of the five histone modifications per 100 base-pairs, referred to as a **bin**. Each particular type of histone modification is associated with a specific type of region on the genome, which are listed below:

Histone Modifications	
Histone Mark	Associated with Regions
H3K4me3	Promoter
H3K4me3	Enhancer
H3K36me3	Transcribed
H3K9me3	Heterochromatin
H3K27me3	Polycomb Repression

- The second are the gene expression read counts from the REMC database. (Sekhon et al., 2018)

The quantitative read counts are transformed to categorical tokens to be fed into our model.

5 Methods/Algorithms

5.1 Overview

Our approach to solving this problem is to train a GPT-2 model from scratch on histone modifications rather than the English language, such that it is able to take input histone modifications and output embeddings that take into account context.

We then feed embeddings of the same genes across two different cell types into an XGBoostRegressor model in order to try to predict differential gene expression counts as the label.

We then find the correlation between the results and the true labels, and compare to state-of-the-art results.

Ultimately, we try to predict gene expression using three types of feature sets. Raw C, in which XGBoost is fed the embeddings of the histone modification levels for two different cell types, Raw D, in which XGBoost is fed the embeddings only of the difference between histone modification levels of the two cell types, and Raw, which is the concatenation of both the Raw C and Raw D features.

The first step to implement this pipeline is to transform the data.

5.2 Data transformation

5.2.1 Note

This part (data transformation) of the project has already been completed as part of prior work, and so only the essential components required to understand this project will be included.

5.2.2 The problem

The first problem is that the data itself is numerical data. An example of the data is given below:

- gene-1-bin-1: 0,0,1,4,3
- gene-1-bin-2: 0,2,3,4,5
- gene-2-bin-1: 1,2,0,0,1

Here, the counts of five histone modifications are given per bin (area of 100 base-pairs), with there being multiple bins per gene.

GPT-2 however, requires an input of sentences made up of words, of tokens. One way to tokenize the histone modification data into words, is to simply concatenate the numerical values. I.e. convert each bin into values such as 00143, 02345, 12001, then concatenate each bin into sentences representing their genes, yielding us sentences of the form:

- gene-1: 00143 02345
- gene-2: 12001 ...

The problem is however, as the information is given as numerical values, there is an infinite number of possible combinations, creating an infinite vocabulary set. This would be too large for our model to train accurately on.

As a result, we decided to threshold each histone modification value into three categories: low, mid, and high levels of modification. This would create a limited vocabulary set of 243 words, where sentences might be of the form:

- gene-1: llmhh lhhhh
- gene-2: mhllm ...

The thresholds deciding what falls into which category, as well as how many categories there are may also serve as hyperparameters for the model.

Exploring the data we see the following distribution of histone modification values, with the following percentiles.

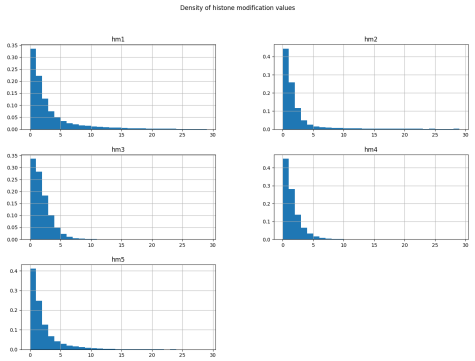

Figure 1: Density of histone modification values

We see that thresholds of 0, 1, and 1+ read counts make up roughly a third of the types of read counts each.

Read count percentiles for each mark			
Histone Mark	0	1	2+
H3K4me3	0-33.4	33.4-55.6	55.6-1
H3K4me3	0-42.9	42.9-67.8	67.8-1
H3K36me3	0-33.7	33.7-61.9	61.9-1
H3K9me3	0-45.1	45.1-73.2	73.2-1
H3K27me3	0-41.1	41.1-65.9	65.9-1

As a result, we split the thresholds into 0, 1, and 2+, transforming them into 'l', 'm', and 'h' respectively. As a result, a bin that was originally 01230 becomes 'lmhhl'.

We then concatenate each bin to make up a "sentence" per gene, where the tokens are words. For example, the rows **[gene-1-bin-1, 0,0,1,4,3] and [gene-1-bin2, 0,2,3,4,5]** become **'llmhh'** and **'lhhhh'** respectively, which then turn into **'llmhh lhhhh'** for gene 1.

These sentences then become the input to the next step of our pipeline: a GPT-2 model.

5.3 GPT-2

The success of CornBERT and GeneBERT indicated to us in the past that GPT-2 could potentially be a good model for the task at hand: differential gene expression.

We used GPT-2 to create embeddings by training it on the standard next-word prediction task. Although next-word prediction is not related to our final regression task, it creates embeddings that take into account context, i.e., the embedding for a word can change depending on what words are near it.

After training our GPT-2 model, we created embeddings for the different cell types, consisting of the same genes, as well as embeddings for the difference in histone modification counts across cell types. These embeddings were then used as input to the final step of our pipeline: an XGBoostRegressor model.

5.4 XGBoostRegressor

Gradient boosting is a technique to produce a strong prediction model using an ensemble of weak prediction models. EXtreme Gradient Boosting, or XGBoost, is an implementation of gradient boosted decision trees specifically designed for speed and performance.(Chen and Guestrin, 2016)

The embedding output of RoBERTa is fed into an XGBoostRegressor model as features, with the log-fold change of a gene's expression across two different cell conditions as the label, i.e., $log(c1 + 1) - log(c2 + 1)$, where $c1$ and $c2$ are the expressions of cell 1 and cell 2 respectively. We add 1 to all read counts to avoid $log(0)$ which is undefined. This is a technique also used in the DeepDiff paper as well as GeneBERT. We used the Pearson coefficient value (r) to compare our model to the ones created by DeepDiff.

6 Results and Discussion

6.1 Raw tokens vs. Embeddings

First, we compared the results given by the raw word tokenizations to those given by the word embeddings. This gives a baseline within the model itself to see if the GPT-2 model embeddings are actually improving on the tokenization itself.

For tokenization, each of the processed sentences (e.g. hmmlm mllml ...) are tokenized using a ByteLevelBPETokenizer into a vocabulary of 752 words such as "hmm" or "mmlhm", identified by IDs. For the baseline with just the tokens, the IDs are fed raw. That is, as input into the XGBoost model, it is simply fed sentences of numerical IDs (112, 532, 752, 0, ...).

For embedding inputs, the word embeddings are extracted from the trained GPT-2 model, where each word in the vocabulary is given a size-N embedding that is optimized as the model is trained. Each of the numerical IDs from tokenization are mapped to their associated embedding, yielding our final input. I.e. "hmmlm mmlmh" is mapped to ["hm", "ml", "m m", "lmlh"] which is then converted to it's numerical id within the vocabulary [101, 672, 752, 2] where each token is then mapped to its size-N embedding, yielding a 4xN embedding matrix.

- **Raw D** uses the difference of corresponding histone modification signals as input $X = X_A - X_B$ for cell types A and B.
- **Raw C** uses the concatenation of histone modification signals as input $X = [X_A, X_B]$ for cell types A and B.
- **Raw** uses the concatenation of histone modification signals in addition to the difference of the histone modification signals as input $X = [X_A, X_B, X_A - X_B]$ for cell types A and B.

Input	Raw C	Raw D	Raw
Raw Tokens	0.403	0.190	0.479
Embeddings (base)	0.415	0.382	0.528

Comparing these results we see that using the word embeddings from the model sees an increase in score over the raw tokens (which do not use the GPT-2 model). This shows that at least the GPT-2 model is helpful.

6.2 Fine-tuning: embedding length

There are various hyperparameters within the GPT-2 model that we can fine-tune to try to achieve better results. One of these is the embedding length produces. The default model used earlier, produced an embedding size of 32, in other words, for each word in the tokenizer vocabulary, they would be mapped to a word embedding of size 32. Here we compare lengths of 32, 128, and 256.

Embedding size	Raw C	Raw D	Raw
32	0.415	0.382	0.528
128	0.421	0.365	0.454
256	0.553	0.493	0.528

Interestingly enough, comparing between an embedding size of 32 and 128, there doesn't seem to be a significant difference in performance. For the Raw C dataset, the R score for Raw C and Raw D is roughly the same, with that for the Raw dataset being slightly higher for the 32 embedding than the 128 embedding. This might mean that the model simply is able to train faster on the 32 embedding and as a result achieve better performance, however this idea seems to be discounted by the 256 embedding which performs better on all measures, except for the Raw dataset, in which it performs equally to the 32 embedding length model.

Given that the Raw dataset is the concatenation of both the Raw C and Raw D dataset, this suggests that with only the Raw C or Raw D dataset, a larger embedding size is helpful to create a more unique representation for each word, however when combined together, the total information is roughly the same, and so embedding length no longer helpful.

6.3 Fine-tuning: context window size

Another feature we can fine-tune is the context window size. This takes into account how large the context considered should be for any particular token.

Context window size	Raw C	Raw D	Raw
32	0.446	0.566	0.479
64	0.415	0.382	0.528
128	0.419	0.427	0.430

For the Raw C data set we see a slight decrease in performance, with the Raw D data set seeing a dramatic decrease than a slight increase, and the Raw dataset seeing an increase then decrease.

The higher performance of a window size of 32 in general suggests that at lower sizes, the model is able to train faster and as a result achieve better performance. But at an embedding size of 64, there is more information that can be used, contributing to a higher performance for the Raw dataset. At 128, there are too many parameters for the model to have trained properly to compete with the 32 and 64 window sizes.

6.4 Sources of variability

6.4.1 Training time

As mentioned earlier, some of the variability likely has to do with training time. Given the time required to train a GPT-2 model, it it's infeasible to train across so many hyperparameters for a full model, and so each model was trained on the same set number of iterations.

6.4.2 Data size

Similarly, the dataset used is extremely small-roughly 23.1 mb. Such a small dataset is not generally viable for transformer architectures, however even for GeneBERT, a RoBERTa model, 23.1 mb did yield considerable results of 0.3 correlation compared to support vector machines which yielded correlations of 0.1. Similarly the state-of-the-art baseline used (DeepDiff) is a Bi-Directional LSTM architecture that used the same amount of data to produce correlations of around 0.6. While it is preferable to have far more data, it does seem that this data size does its job as showing GPT-2 to be effective as a proof of concept, with correlations as high as 0.5. However, as the dataset is extremely small, this may contribute to variability, as based on the fact that training runs are randomized, it's likely that with such a small amount of training random factors may play a larger role than if the model could train over far more data, providing more normalization.

6.5 Comparisons with other models

Our best models trained on the embeddings created by a GPT-2 model gave us Pearson coefficient (R) values of **0.566**, **0.553**, and **0.528**, for the Raw D, Raw C, and Raw feature sets which were above the values for a single support vector machine around **0.1**, as well as GeneBERT around **0.3**, but below that for DeepDiffChrome which were **0.65**, **0.68**, and **0.67**, after being trained and tested on the exact same data.

It is surprising however, that simply using the BPE word tokenization without the GPT-2 model, produced results as high as **0.403**, **0.190**, & **0.479**, which beats both the SVM and GeneBERT results. This suggests that compared to the bare words, a higher dimensional (roughly 207 token) representation for each word simply provides a more detailed and unique representation that might take into account smaller scale relationships that aren't accounted for when using just the full words.

Although our GPT-2 model was trained on an extremely small dataset, totaling around 23.1 mb in size, combined with a minimal XGBoostRegressor model it was able to produce better results than models such as single support vector machines, as well as the RoBERTa model (GeneBERT) but it was not able to beat out the state-of-the-art DeepDiffChrome. This is unsurprising however, as we found that it is in fact extremely difficult to train a viable GPT-2 model on a small corpus, and that GPT-2 by itself is only able to be applied for tasks such as regression or classification through the addition of an additional layer above the model.

Despite this, our results with a fairly minimal model are promising and warrant further investigation of this technique.

The table below shows how the best versions of our model performs compared to different variations of DeepDiff using the Pearson coefficient as the metric. The variations correspond to how the input histone modification data is manipulated before being fed into the model.

Model	Raw D	Raw C	Raw
RoBERTa + XGBoost	0.31	0.38	0.30
GPT-2 + XGBoost	0.566	0.553	0.528
DeepDiff	0.65	0.68	0.67

7 Conclusion

Based on the high correlation values from our GPT-2 model (around 0.5) in comparison with the state-of-the-art DeepDiff model (around 0.6), we find that the use of GPT-2 is promising and warrant further investigation into the methods used. CornBERT showed that there is potential to use transformer architectures towards predicting gene expression, GeneBERT showed that even with such a small corpus, such a model can produce high results, and our model GeneGPT-2 has shown that the GPT-2 strongly outperforms the BERT architecture.

We are optimistic that by gathering a larger corpus, we can improve the embeddings further, and that there is also room to improve the regression model used to generate predictions. This motivates further exploration of this technique to potentially achieve results that are closer to, or even better than, the state-of-the-art results obtained by DeepDiff.

Future work might involve training the GPT-2 model for longer, as well as exploring the integration of a fully-connected layer at the end for the task of regression rather then the extraction of word embeddings to be fed as input to an XGBoost model. This may yield better results, as now the GPT-2 model is directly trained for the regression task.

The public GitHub Repo showcasing the ipynb file and processed data folder to get the project working is shown here: https://github.com/ZovcIfzm/GeneGPT2

References

Tianqi Chen and Carlos Guestrin. 2016. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Arshdeep Sekhon, Ritambhara Singh, and Yanjun Qi. 2018. Deepdiff: Deep-learning for predicting differential gene expression from histone modifications. *Bioinformatics*, 34(17):i891–i900.

Han Xu, Chia-Lin Wei, Feng Lin, and Wing-Kin Sung. 2008. An HMM approach to genome-wide identification of differential histone modification sites from ChIP-seq data. *Bioinformatics*, 24(20):2344–2349.

Zihao Xu. 2020. Bringing bert to the field: How to predict gene expression from corn dna.