Exploring Regulatory Genomic Regions and Causal Variants with gkm-SVM and its Interpretive Methods

CME 216 Course Project

David Neese

1 Introduction

The aim of this project is to help show the use of the gkm-SVM classification method, along with the methods that interpret its output, in the difficult biological problem of identifying causal variants. Causal variants are alterations of "normal"[1] genome sequences (often just a single swapped out base pair or SNP[2]) which are causally linked to disease, or more broadly distinctive molecular phenotypes. They are challenging to identify, because the causal pathway from DNA sequence to phenotype is usually monstrously complex. Most disease risk factors are polygenic in origin and are further influenced by epigenomic effects - we can highlight their correlation with many variants, but labelling any variant as causal is much harder. If I get a heart attack it is likely due to much more than just one mutation in one allele.

What we can say is that most causal variants seem to be in non-coding regions, that is, the parts of RNA that regulate the production of the proteins a mature RNA transcript encodes. Rather than modifying an actual gene product (protein), these variants might lead to a gene being overly transcribed or completely suppressed. Identifying regulatory enhancer and promoter regions is unfortunately also difficult - there are no hard and fast deterministic rules, and there is much variance among difference cell types. A standard way of doing it is through ATAC-Seq and ChIP-Seq experiments, which can find regulatory regions transcription factors[3] bind to, and through this suggest where enhancers and promoters are. These methods, however, require prior knowledge of what is being looked for and are impractical to scale to all regions in all cells.

Machine learning can knock out both of our problems at once. If we train a classifier to take in a DNA sequence and tell us if it is acting as an enhancer or promoter for a given cell type, we can (1) infer the cell type specific "grammar" that marks out regulatory regions and (2) analyze the effect of a variant on a regulatory region's effectiveness (thereby determining if it is likely to be causal). It is also likely that a trained classifier might work where a more analytic method fails; we have many known examples of enhancers and promoters by cell type, and similarity to those regions tends to be a very good predictor of similar function. This machine learning task is exactly what gkm-SVM and its interpretive methods perform, which is elaborated in the next section.

[1] The idea of a normal or baseline human genome is useful for thinking about mutations, but is very shaky considering the inherent diversity in the human genome we are continuing to uncover with study

[2] Single nucleotide polymorphism

[3] Transcription factors are proteins which bind to enhancer and promoter regions in non-coding regions of RNA (introns) and in so doing regulate the transcription of the associated coding regions

2 Survey of Methods

The gapped k-mer support vector machine (gkm-SVM) is a linear classifier that computes the similarity of an input DNA sequence to known enhancer and promoter regions, and using this predicts whether the input itself acts as a regulatory region in the specified context (3). For input feature vector $\mathbf{x}$, it can be written:

$$F(\mathbf{x}) = b + \sum_i \alpha_i y^i K(Z^i, \mathbf{x}) \tag{1}$$

Where b is a bias vector, α_i is the weight associated with the ith support vector Z_i, y_i is the associated label (±1), and K is the kernel function (computing similarity between $\mathbf{x}$ and Z_i). The first two are trained (usually once for every cell type[4]), and the kernel is evaluated anew on each input sequence. To delve deeper into the Kernel, it is structured to calculate a biologically relevant measure of similarity between two strings representing DNA strands of A, T, C, and G. It outputs the normalized overlapping counts of distinct gapped k-mers[5] in two sequences (one a support vector, the other the input). It can be represented as (1):

$$K_{\text{gkm}}(S_1, S_2) = \left\langle \frac{f^{S_1}_{\text{gkm}}}{||f^{S_1}_{\text{gkm}}||}, \frac{f^{S_2}_{\text{gkm}}}{||f^{S_2}_{\text{gkm}}||} \right\rangle = \frac{\sqrt{\langle f^{S_1}_{\text{gkm}}, f^{S_2}_{\text{gkm}} \rangle}}{\sqrt{\langle f^{S_1}_{\text{gkm}}, f^{S_1}_{\text{gkm}} \rangle}\sqrt{\langle f^{S_2}_{\text{gkm}}, f^{S_2}_{\text{gkm}} \rangle}} \tag{2}$$

Where each f is a feature vector of k-mer counts for a sequence. Why gapped k-mers? This is where some biology comes in. Transcription factors bind to motifs in enhancers, but these motifs are not well defined short sequences for each TF - instead, each TF has an affinity for a given sequence, some higher and some lower. It is a soft interaction, not a key and lock one - so gaps in binding sites can still indicate enhancer regions. If a TF binds to the sequence ATACGA, it may also bind to ATACGT without much issue. However, it may refuse to bind to ATCCGA for some reason or another, making that A\C substitution a causal variant. The calculation of this kernel is simplified by the following shortcut (1):

$$\langle f^{S_1}_{\text{gkm}}, f^{S_2}_{\text{gkm}} \rangle = \sum_i \sum_j h\left(f_m(u_i^{S_1}, u_j^{S_2})\right) = \sum_i \sum_j \binom{l-m}{k} \tag{3}$$

Which essentially reads: if $u_i^{S_1}$ and $u_j^{S_2}$ are a pair of l-mers (at positions (i, j) in sequences (S_1, S_2)) with m mismatched bases between them, then the number of gapped k-mers they share is $h(m) = \binom{l-m}{k}$. This along with the kernel trick leads to huge computational speedup - we don't need to enumerate all possible states to compute their overlap and dot product.

But now for more biology - once we have a prediction of whether a sequence is acting as an enhancer or not, can we do any more with it? This is where we recall the initial question of causal variants. It turns out there are a few methods that can use this classifier to demonstrate the effect of variants on regulatory genomic regions and thus gene expression.

Two of the more common methods are deltaSVM (4) and ISM (In-Silico Mutagenesis) (1). DeltaSVM estimates the induced change in classifier output caused by predicting on a variant instead of a reference sequence. It adds up the total change in scores of all l-mers overlapping a mutation. This works well for quantifying the effect of variants for the linear gkm-SVM kernel, but fails to consider

[4] Each cell type has its own "grammar" of motifs, which will be described a little later - essentially each cell type has certain transcription factors which makes it likely that certain sequences will act as enhancers or promoters where they may not in other cell types. In this particular experiment, we group by cell cluster (some cell types have more similar TF motif grammars than others) instead

[5] The sequence 'ATC' would have 3 distinct gapped 2-mers: AT*, A*C, *TC

non-additive relationships between l-mers (ex: 'ATA' and 'CTG' alone both mark out enhancers, but not when present together) or positional weights that are linked with non-linear kernels like wgkm or wgkmrbf (1). ISM is a computationally expensive way of dealing with these more complex relationships - it directly computes the effect of a variant on classifier output. If we want to step through all possible variants in an l-mer though, this quickly becomes expensive.

A more recent and robust method is gkmExplain, which I will principally be using. gkmExplain essentially decomposes the kernel of the gkm-SVM to infer how much weight each position or base pair in a sequence has in determining the its regulatory status (1). Once we have this matrix of positions and normalized weights, we can plot them and easily see which l-mers are the determining factors in classification. This can further support motif discovery - something the other methods do not do. When it comes to visualizing the effect of a variant, a gkmExplain PWM[6] plot can also show which enhancers are boosted or repressed by a change in the input. This makes it the best explanatory framework for gkm-SVM in our case.

3 Code Setup

For the setup of my project, I decided to use data from two cell types with variants highly linked to Alzheimer's disease (2). I trained two ls-gkm models (5) on 20,000 sequences each (evenly split between positive and negative[7]) for two different clusters of cell types - one for excitatory neurons, and one for microglia. The ls-gkm model I used had quite a few flags, most of which I didn't end up using - training ended up taking extremely long (6-7 hours for a highly reduced model), so I mostly applied prior knowledge in selecting hyperparameters rather than playing around with them. A few things to note

- Kernel: I used the standard (i.e. not center weighted) gkm kernel, or flag 2, based on prior knowledge that positional weighting would not be relevant for these cell types
- Data: I did subset the data a few times to get the model down to a manageable size - I started with 120,000 pos/neg sequences, then chopped it down to 60,000 (which still only trained halfway after 10 or so hours), and finally knocked it down to 20,000 sequences, which still seemed to be enough for my purposes in this project. For reference, toy examples I had seen had only used around 1000 or fewer carefully chosen sequences to train.
- Epsilon: this was a threshold the gradient had to dip under to finish training, set by default to 0.001. In my initial runs where the model was converging very slowly, I set this parameter to much higher (2.5), which ended up outputting a garbage model. I eventually settled on keeping it at the default value and simply using less training data (I've shown a graph of this value over each training epoch in the code).

After I trained my two models, I validated them with test sets (again cell-type specific, split into positives/negatives) from the same source, ending up with about a 90/10 train/test split. I measured a few stats to make sure the model was working well:

- Cl1 model: Precision = 0.88, Recall = 0.82, ROC AUC = 0.97
- Cl24 model: Precision = 0.88, Recall = 0.75, ROC AUC = 0.95

[6] Position Weight Matrix, which shows the contribution of each base pair at each position to a given output - here, to a sequence's prediction as being an enhancer or not. See the code for more details

[7] Positive examples are sequences that have been found to act as enhancers in these cells, negative examples ones with comparatively little TF binding - both labels are determined by the outcomes of ATAC-Seq experiments (positives are peaks)

Finally I fed a few predicted enhancer regions into gkmExplain to generate some PWMs (modifying code from (1)), run some tests and explore results.

4 Results and Analysis

The main goal of this project was to demonstrate what these models were capable of at a basic level, rather than try to apply them to obtain new and significant results. In light of this, I tried to run some tests that would give insight into general principles for how gkm-SVM and gkmExplain work.

The first thing I tried was cross-applying one of the models - I used the model trained on Excitatory Neuron data to predict outcomes for the Microglia training sets. The results were fairly expected - the model's performance was worse (ROC AUC of 0.92), but not by that much. This makes sense since Microglia and Excitatory Neurons are similar cell types (both are in the brain, and probably share some common functions or are receptive to some of the same transcription factors). The transcription factor motif "grammar" is not quite the same across both, but it is close. What I am curious about is whether multiple models trained on Excitatory Neuron data would show similar variance to that between the two models I used when applied to Excitatory Neuron test data - i.e. whether the difference between the models is statistically significant or not.

The second thing I tried was using gkmExplain to show the effect of variants on regulatory function. I selected the first three samples from the positive testing data in Excitatory Neurons and modified the fasta file to contain two random variants of each. I then scored these using `gkmpredict` and decomposed and visualized their kernel outputs using gkmExplain. Both revealed little variation. The classification scores of the reference sequences differed from the variants by around 3% or less in most cases. The gkmExplain learned PWMs, when visualized (see an example below), also did not appear much different between variants and their reference.

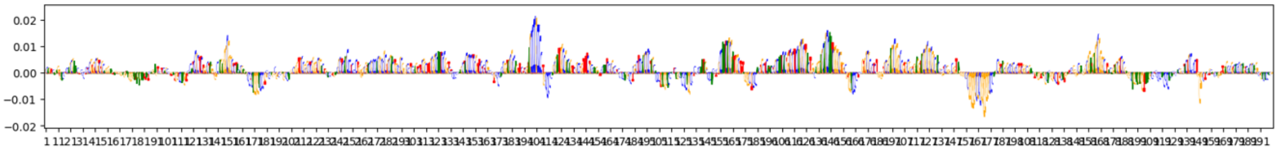

Figure 1: Visualization of the PWM for a sample of Excitatory Neuron regulatory DNA, inferred by gkmExplain

This was a good reminder of the non-triviality of identifying causal variants. Despite what one might initially assume, just any variation at any point in a regulatory region won't impede its function. Many variants just don't make much of an impact, as one can see in these tests - one can hardly tell that there's any variation in the graphs of each sample. Even if a variant were to be causal, its effect might not be so dramatic as flipping the classification of the region it was in - in any such region, there are many transcription factor binding motifs, and the inhibition of one may not lead to the inhibition of all of them.

If I were to refine my analysis (which I began but couldn't complete in time), I'd select a few known causal variant SNPs from GWAS, score the intron regions they were in, and plot them against the reference sequences over a smaller interval. This would give us a clearer view of a motif being disrupted by a variant with serious effects.

Overall, I hope to have shown gkm-SVM and its accompanying explanatory tools to be powerful, but not providing all the answers to our problems of interest. Interpretation is still not perfectly clear, and incredibly large amounts of data are still needed to make these algorithms work with any reasonable accuracy. Careful research and application is still needed to get the most from them.

References

[1] Avanti Shrikumar, Eva Prakash, Anshul Kundaje, GkmExplain: fast and accurate interpretation of nonlinear gapped k-mer SVMs, Bioinformatics, Volume 35, Issue 14, July 2019, Pages i173–i182, https://doi.org/10.1093/bioinformatics/btz322

[2] Corces, M.R., Shcherbina, A., Kundu, S. et al. Single-cell epigenomic analyses implicate candidate causal variants at inherited risk loci for Alzheimer's and Parkinson's diseases. Nat Genet 52, 1158–1168 (2020). https://doi.org/10.1038/s41588-020-00721-x

[3] Ghandi M, Lee D, Mohammad-Noori M, Beer MA (2014) Enhanced Regulatory Sequence Prediction Using Gapped k-mer Features. PLoS Comput Biol 10(7): e1003711. https://doi.org/10.1371/journal.pcbi.1003711

[4] Lee D, Gorkin DU, Baker M, Strober BJ, Asoni AL, McCallion AS, Beer MA. 2015. A method to predict the impact of regulatory variants from DNA sequence. Nat Genet. advance online publication. doi:10.1038/ng.3331

[5] Lee, D. LS-GKM: A new gkm-SVM for large-scale Datasets. Bioinformatics btw142 (2016). doi:10.1093/bioinformatics/btw142