Classifying circRNA

Eric Wolf （种伟神）
Project Report

Features

Kmers

My classifier primarily relies on kmers, i.e. the relative frequency of base pair sequences of length up to k. Due to limitations of computational resources, I have chosen k=5, resulting in 1.364 features. Longer lengths could become computationally feasible when using gapped kmers[1]. The features were scaled by the length of the gene, and the length was not included among the features. This was done on purpose to avoid classification purely by length when using the exons as negative samples (see below).

ALU repeats

As ALU repeats occur more frequently near circRNA boundaries[2], ALU repeats before and after circRNA boundaries were counted. Inspired by Jeck et al., 50, 100, and 200 bps up- and downstream were considered, as well as the content of the gene itself. The features used measured the ratio of base pairs in that area covered by any ALU repeats.

Chromosome

As circRNAs are not evenly distributed across all chromosomes, I also attempted to add the chromosome as a feature with a one-hot encoding. However, this did not improve results, and was therefore dropped from the final solution.

Negative samples

Using individual exons as negative samples seems unwise, as these can be trivially distinguished from circRNAs in many cases by their length. Also, circRNAs commonly consist of multiple exons and introns, implying that a classifier trained on exons might actually be trained to recognize introns or the boundaries between exons and introns. Other classifiers sidestep this problem by using other lncRNA as negatives and thus use a narrower problem scope[3]. The classification of exons and circRNA seems a lot easier than the classification of other lncRNA and circRNA.

In order to avoid classification by length, the length was expressly excluded from the features. To construct better negative samples, I have generated genes spanning multiple exons, starting and ending within exons, from the exact length distribution of the given circRNAs. This should give rise to a harder, but more realistic classification problem.

In any case, the number of negative samples was always chosen to be equal to the number of positive samples to prevent the issues arising from imbalanced data.

Algorithm

I have evaluated the usage of an SVM with Gaussian, linear and polynomial kernels, k-Nearest Neighbor, linear regression as well as Random Forests. Even though considerable time was spent on finding the best hyperparameters for an SVM using grid search, Random Forests consistently exhibited the best performance across all metrics used. Additionally, training on Random Forests was faster than on an SVM and allowed for the extraction of a feature ranking to better judge the usefulness of individual features.

Fast access to the reference gene was implemented using pyfasta, which allows memory-mapped access to FASTA files. Quick lookup of genes for the calculation of the ALU-based features is using an interval tree. For kmer

[1] Ghandi M, Lee D, Mohammad-Noori M, et al. Enhanced regulatory sequence prediction using gapped k-mer features[J]. PLoS Comput Biol, 2014, 10(7): e1003711.

[2] Jeck W R, Sorrentino J A, Wang K, et al. Circular RNAs are abundant, conserved, and associated with ALU repeats[J]. Rna, 2013, 19(2): 141-157.

[3] Pan X, Xiong K. PredcircRNA: computational classification of circular RNA from other long non-coding RNA using hybrid features.[J]. Molecular Biosystems, 2015, 11(8):2219-2226.

counting, there are some highly optimized libraries such as khmer. However, their lack of documentation made it hard to integrate them for this application.

The parallelization of feature calculation and Random Forest training did, however, greatly improve the performance of the algorithm. On a compute-optimized AWS instance, feature calculation and cross validation of the entire data set take about 25 minutes.

I also tried to use the large-scale gapped kmers SVM implemented by Ghandi et al. A test run on a subset of the data with the default hyperparameters showed promising results, however, training of lsgkm took too long to further pursue this.

I also briefly attempted to implement an RNN, but soon ran into performance issues as the sequence length, even on a small training set, leads to very long training times and the need for high-performance hardware. Others seem to have encountered similar problems and have resorted to solutions such as truncating the sequence length at 100bp[4]. As some circRNAs have a sequence length of up to 10.000bp, however, that seems to severely restrict the flexibility of the model.

Performance

Negative samples	Features used	AUC	AUPR	F1
Exons	kmers	0.9660	0.9711	0.8962
Exons	kmers & ALU repeats	0.9661	0.9712	0.8967
Generated genes	kmers	**0.9809**	**0.9798**	**0.9301**
Generated genes	kmers & ALU repeats	0.9653	0.9711	0.8986

All results are mean values after 10-fold cross validation

See the attachment for the feature ranking.

Discussion

Surprisingly, the performance exhibited in all of the tested configurations was very similar, even though earlier evidence suggested that when using exons as negatives, the classifier would significantly outperform the generated negative samples. Certainly, this would be different if the length were included in the features. Since most of the hyperparameter tweaking was done when using the generated negative samples, it is possible that the model is better-suited for the generated negative samples, thus making up for the higher difficulty.

Using the ALU repeats also did not contribute to better results – in fact, when using the generated negatives, the model not using ALU repeats slightly outperformed the other models. This suggests that either those features have to be encoded differently, for example as binary features, or that the training method should be changed to better accomodate heterogeneous features, such as multiple kernel learning[5], or that the ALU repeats are already recognized through the kmers. In the Random Forest's feature ranking, three of the ALU repeat features appear in the top 50 features, which suggests that they do contain some meaningful information.

Acknowledgements

I discussed the parsing of the data as well as the feature extraction with my classmates Russlan Ramdowar and Noah Hollmann.

[4] Zhang J M, Kmath G M Learning the Language of the Genome using RNNs http://cs224d.stanford.edu/reports/jessesz.pdf

[5] Pan X, Xiong K. PredcircRNA: computational classification of circular RNA from other long non-coding RNA using hybrid features.[J]. Molecular Biosystems, 2015, 11(8):2219-2226.