The predictive power of entropy, control set selection and ensemble learning: An addendum

Christopher

December 17, 2020

1 The discriminatory power and information content of the Shannon entropy in RNA alignments

As was already discussed in the report 'A short statistical investigation of RNA features used in classification with RNAz' (November 2020), evidence suggests, that the discriminatory contribution of the Shannon entropy feature in ncRNA classification is in fact marginal when compared to structural conservation and z-score of minimum free energy.
To lend further credibility to this claim, investigations of two more aspects of this features properties were conducted.

1.1 Random forest feature importance

First, it was already established in November, that a random forest classifier is also feasible in structural RNA classification (¿0.95 accuracy on an unoptimized, experimental classifier). Random forests, to recapitulate, derive a classification by majority vote from a randomized subset of the decision tree set making up the forest. The individual decision trees in the used models are themselves binary trees, i.e. each node is represented as exactly one threshold value corresponding to one feature. This now allows for an estimation of the relative statistical significance of each of the three features used, a property otherwise known as 'feature importance'. A visualization of feature importance of z-score, SCI and entropy can be reviewed in figure 1.

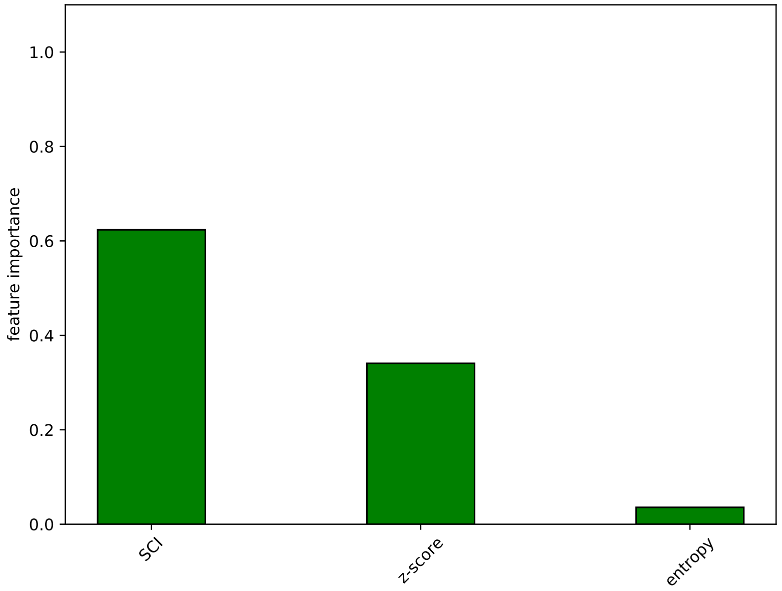


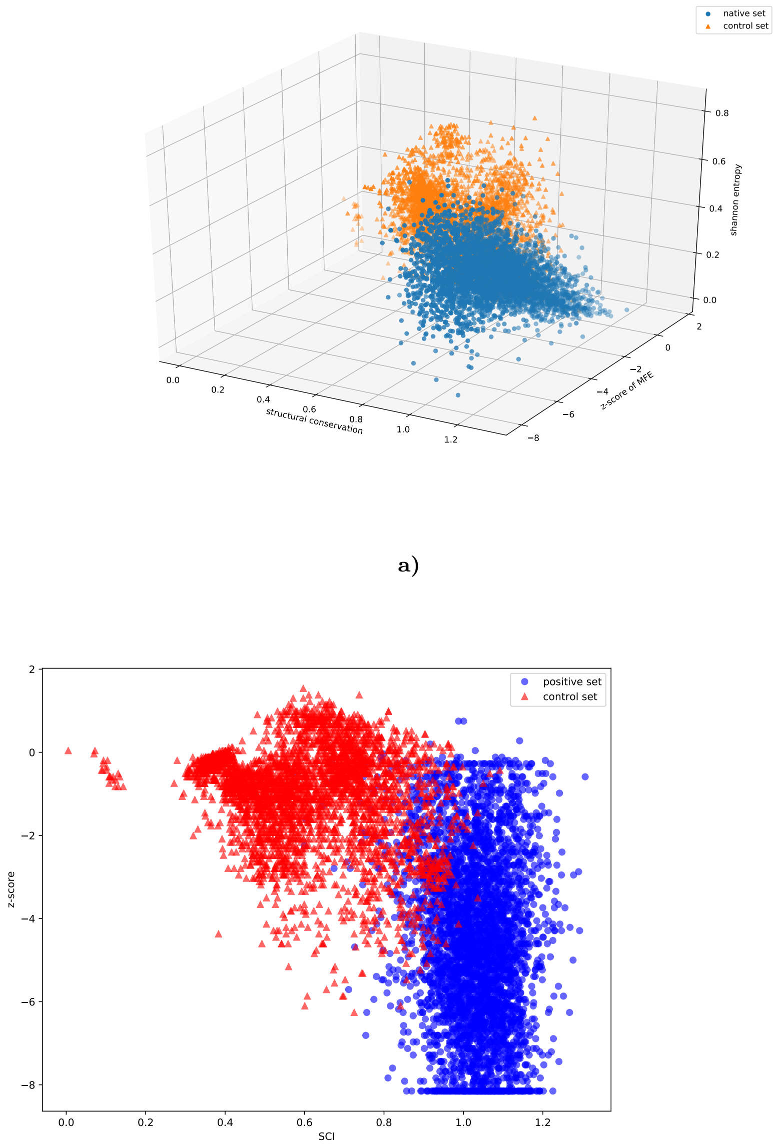
Figure 1: Normalized significance of each feature in ncRNA classification, using a Random Forest binary classifier. The forest used was composed of 500 individual decision trees using bootstrapping. Furthermore, the following restraints were established to reduce overfitting of the training set: For each new split at an existing node or each new leaf to be added, the new path had to at least represent 1% of the underlying data points sampled to grow each tree. The forest was trained on a data set consisting of 4303 positive and negative instances each.

As can be observed, the Shannon entropy has by a large margin the least discriminatory power from the set of used features. This, as mentioned, was already suggested before, but it further backs the hypothesis that aside from being a statistical curiosity, the feature does in fact not contribute much in terms of binary classification.

1.2 Principal component decomposition

Principal component analysis (PCA), sometimes also referred to as main component analysis, is a statistical tool to estimate the total contribution of variance in a data set broken down by dimension/degree of freedom (or axis, or in this case features). In machine learning, this is obviously typically used for dimension reduction in feature-rich problems. Even though the total number of features here is small, PCA can still be used to estimate the overall information contribution of a given feature.

The (three-dimensional, i.e. SCI, z-score and entropy) data set from the previous section was used and 'compressed' to a two-dimensional shape by omitting the entropy-axis (see figure 2). The overall loss in variance associated with the removal of this dimension was calculated using PCA. It was found, that the entropy feature can be removed from the data set completely, while retaining a fraction of $\tilde{0}.96$ of the original variance. This further implies, that the total contribution of the Shannon entropy to the information contained in the feature vectors is marginal when compared to the remaining features.



a)

b)

Figure 2: The data set used in figure 1 plotted in 3D space (a) and the same data set reduced to two dimensions by removing the entropy axis. Principal component analysis shows, that 96% of the numeric variance can be retained when removing the Shannon entropy as feature.

However, this short experiment also shows, that while the variance contribution of the Shannon entropy is small, it is not nonexistent. The two-dimensional feature vector still contains information that is measurably lower than the three-dimensional one. Therefore, this short analysis serves more to bring this discussion to a preliminary conclusion than to make the case that this feature is completely useless and should be dismissed.

2 Negative set generation: The selection of pseudo-random genomic loci

The limitations of the SISSIz-based approach to control set generations indicate a need to establish viable alternatives. In the RNAz 2.0 release paper, part of the used data sets were built using randomly sampled genomic sequences for negative instances with a fixed dinucleotide content. This approach has the obvious advantage of reducing the danger of including sequence artifacts caused by the shuffling process. However, the selection of subsequences tailored for specific sequence contraints poses a challenge. For this reason, I propose the following experimental pipeline setup to extract specific sequence subalignments from whole genome alignments with specific sequence and structural properties.

A core tool in the RNAz 2.0 framework used in whole genome screens is the rnazWindow.pl script, which serves to break down the full alignment in overlapping windows of a given length. Now, in a standard screening procedure to identify ncRNA loci, RNA features (SCI, z-score, Shannon-entropy) are calculated for each window and then passed to the classifier. However, I suggest using these features to identify viable negative training candidates from the full pool of generated genomic alignment windows.
Analyzing the data sets included in the RNAz 2.0 release, it becomes quite obvious, that the feature tuples of the negative sets are not randomly distributed, but clustered around a numeric center of mass (which lies, in the data set in figure 1 and 2, around [0.6, -1, 0.35]). Since we are in essence proposing, that non-ncRNA sequences all share a distinctive property identifiable by these features, it is assumed that this center is more or less the same in all RNA alignments that are of comparable length, dinucleotide composition and not identified as ncRNA. Note, however, that the above is presumed to be true only because the negative set taken as an example was built from sampled genomic loci and not shuffled or otherwise synthetic sequences.

How would one now best identify alignment windows that correspond to the cluster outlined above? As the name given indicates, I would suggest using k-means or another efficient clustering-algorithm to cluster the alignment windows beforehand and then categorize prospective candidates by their numeric distance. However, using k-means clustering on millions of alignment windows becomes computationally expensive very fast. This is the reason, why I suggest using the mentioned 'center of mass' as an artificial starting point for clustering: It can be trivially shown that using an entry point known to be close to the presumed cluster center speeds up computation dramatically when compared with a random start.

Suppose now, that using a clustering approach we have successfully identified a sufficient number of prospective negative set candidates. A piece of information is still missing here: The negative set is supposed to mirror the positive set in sequence composition, gap patterns and overall length as closely as possible.

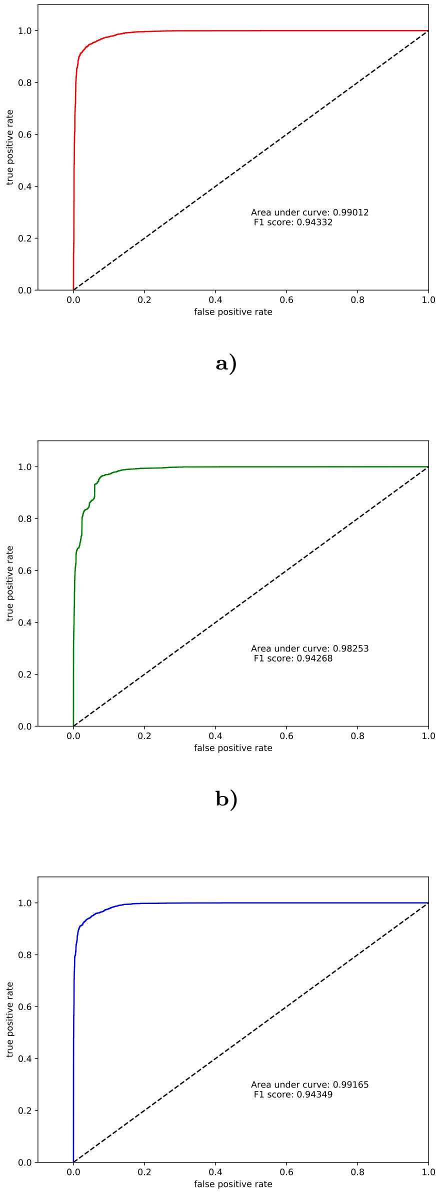
Dinucleotide frequency per sequence can be calculated in a reasonable time frame for each alignment window, and a simplified gap score can be used to approximate gap patterns (note here, that SISSIz also does not exactly mirror gap occurence in simulated alignments). Using these values, a grouping of each alignment window by sequence composition using threshold values is possible. The remaining task is then to draw a number of alignment windows (or their respective features tuples) from each group corresponding to the composition of the positive training set.

3 An experimental classifier-ensemble

As of now, alternative approaches to classifier implementation are still investigated. One additional angle not experimentally looked at until now, is the usage of Ensemble classifiers. Ensemble classifiers, as the name would imply, are obviously conglomerates of classifiers with different mathematical and algorithmic approaches linked in parallel, which reach a consensus classification by majority vote. A simple example of this approach is the Random Forest, which consists of a set of individual decision trees with varying decision thresholds.

An important distinction in ensemble classification is the usage of a hard and soft margin. While a hard margin classifiers reaches a consensus by merely 'counting the votes', a soft margin classifier weights the votes based on their relative class probability values, i.e. a vote with an attached probability of 0.6 will be weighted lower than one with a probability of 0.95. In practice, the soft margin approach is used less often, as it hardly improves accuracy in edge cases (i.e. all votes are close to 0.5 in class probability), which are one of the major sources of misclassification.

As a proof of concept, a simple Ensemble classifier linking three separately trained classifiers was created: An unmodified SVM classifier, a Random Forest classifier presented in November and a new Logistic Regression binary classifier, trained specifically as an example for this use case. It is important to note, that each of these classifiers can be swapped out or the list extended. This would allow, for example, the usage of multiple SVM classifiers trained on smaller and more specialized data sets to be linked.



a)

b)

c)

Figure 3: Visualization of classifier performance of (a) SVM classifier, (b) Random Forest classifier and (c) Logistic Regression classifier. All three were used in parallel to form a classifier ensemble. Training set was identical for each classifier and consisted of 3886 curated positive instances and an identical number of negative instances generated with SISSIz. Test set was the data set used in the visualization of figure 1 and 2.

In practice the linked classifier performs marginally better than each individual with an F1-score of 0.945, if using a hard voting margin. Somewhat surprisingly, the Logistic Regression classifier actually performs the best given the data set at hand. A drawback is the inability to generate a ROC-curve for the full Ensemble, as calculating false positive rates for different p-value cutoffs obviously necessitates a class probability. However, an overall linked class probability can only be calculated if using a soft margin system.
Note further, that all new classifiers are very much unoptimized at this point and this addendum only serves to explore the possibilities of implementation.