Experimental cleanup of loose ends in Svhip development

Christopher

November 13, 2020

1 Introduction

As Sven pointed out, there are a few points of interest remaining in the comparative performance evaluation of classifiers regarding the effects of variables in the Svhip implementation. Primarily, this concerns the suspicion that a too high fraction of viable data points are removing from training sets due to the two-fold screening process implemented for the first set of evaluation experiments. More precisely, it might be sufficient to just filter Rfam families for their number of structurally viable alignment windows and then not further reduce these by filtering against another randomized control set. To determine the validity of this approach, a direct comparison of classifiers trained on pre-screened Rfam data sets without further structural filtering will be done. On this note, moving the cutoff determining what consists a viable alignment should also be considered.
Furthermore, it has been noted multiple times that further experiments are required to determine the optimal point of control alignment simulation. The current approach uses a SISSIz based simulation at the time of initial input alignment processing. Since a generation of control alignments is in any case necessary at the time of structural conservation estimation, it might be viable to simply generate control alignments based on created alignment windows and then also proceed to use these in negative set construction. Both approaches were tested independently.

2 Results & Discussion

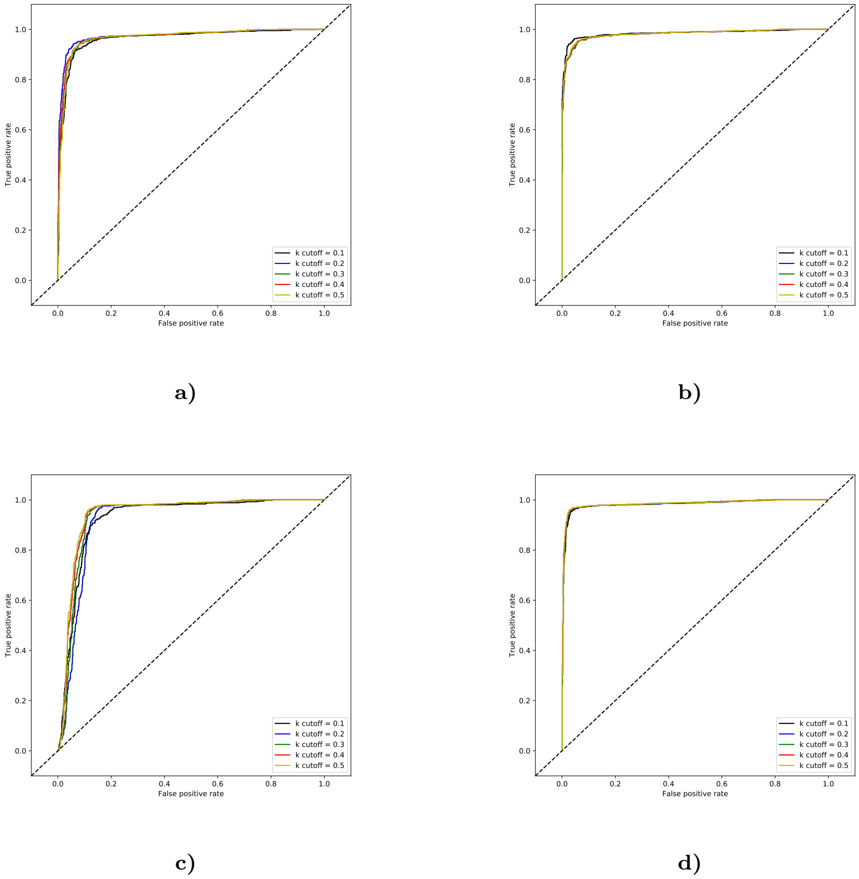
2.1 One- or two-fold structural filtering

Two structural filtering parameters were suspected to have the most significant influence on classifier accuracy. The influence of the k-cutoff value and the separate filtering of input candidates before window filtering were investigated separately. Experimental results suggest that there is no significant improvement in classifier performance when using a structural filter to pre-select input alignments while also filtering independent alignment windows. It is likely that this in fact results in a very high redundancy of filtering steps if the k value is high enough. Four classifiers originally trained on data sets assembled from pre-screened Rfam families were retrained with additional filtering of each individual alignment window. Windows were only included in the final training data, if their respective mean tree edit distance was less than the $k - percentil$ of mean tree edit distance of the control sets. It can be easily seen, that for the pre-screened set the final classifier accuracy is only sparsely affected. In fact, for certain test and training set combinations, a slight improvement in accuracy can be observed with a higher k-value (see figure 1). This can likely be attributed to the fact that a strict k-value in combination with a pre-selection of input data based on structural conservation leads to the removal of a significant amount of viable training data from the set. Table 1 shows the total number of data points in training sets used. This effect is however not consistent across all sets and is negligable in magnitude.

It can however be observed, that the final training set with a k-value of 0.5 contains on average 22.7% more training instances than the set with k = 0.1. The sharpest increase in the total number of instances can here consistently be observed when increasing k from 0.1 to 0.2. Since a higher number of training instances is typically seen as desirable in SVM training, using at least a k-value of 0.2 seems to be preferable to 0.1. Using the F1-Score as a metric for classifier performance, then overall lower k-cutoff values are to be preferred (0.1 to 0.3). Nonetheless, the effect size is still very small when using the two-fold structural validation strategy. For an estimation of the effect on not pre-screened training data, a training set consisting of alignments expliciteley not contained in the set of pre-screened Rfam data has been assembled. Therefore, this training data was filtered using only the one-fold approach through direct comparison with the complementary negative set. A k-cutoff value of 0.2 was used, to obtain a sufficient number of training instances, while also retaining a high degree of separation between native and control set. Directly comparing the resulting classifier performance shows, that this classifier in fact slightly outperforms the two-fold filtered classifier on the given test set. The difference is with a mean value of 0.04 however small.

Table 1: Summary of the composition and performance of classifiers trained on pre-screened Rfam alignment data further filtered by using a structural threshold for generated alignment windows. The k-value refers to the fraction of the control set mean tree edit distance the edit distance value of a given alignment window has to be below. AUC and F1-Score were used to examine performance on a representative test set with 2468 data points. Used data sets were built upon the best performing four classifiers trained in the course of the master thesis.

data set	k	instances	AUC	F1-Score
1	0.1	6477	0.964	0.882
	0.2	7248	0.974	0.889
	0.3	7588	0.968	0.879
	0.4	7799	0.971	0.880
	0.5	7953	0.968	0.879
2	0.1	8739	0.984	0.896
	0.2	9826	0.981	0.866
	0.3	10191	0.981	0.868
	0.4	10390	0.982	0.865
	0.5	10553	0.983	0.868
3	0.1	7773	0.926	0.861
	0.2	10178	0.920	0.872
	0.3	11603	0.933	0.866
	0.4	11865	0.940	0.865
	0.5	12004	0.943	0.865
4	0.1	11400	0.983	0.905
	0.2	12859	0.983	0.896
	0.3	13144	0.984	0.895
	0.4	13220	0.984	0.899
	0.5	13307	0.985	0.901



a) **b)**

c) **d)**

Figure 1: Performance comparison of classifiers trained on data sets assembled from Rfam families selected for high structural conservation. Generated alignment windows were further filtered by only including those in the final training set that had a mean tree edit distance lower than the k-th fraction of the control group tree edit distances. The k-values used were [0.1, 0.2, 0.3, 0.4, 0.5]. The test set used contained 2468 positive and negative instances of each class (RNA and OTHER) and was assembled from structurally conserved Rfam alignment data.

2.2 Strategy of control alignment assembly

Two different approaches to the automatic assembly of negative training instances have been investigated. Control sets are generated using SISSIz to create alignments with identical decision dinucleotide composition and gap patterns. Control alignments can be generated either directly from the input alignment or

Table 2: Performance comparison of classifiers trained using two different strategies of control set generation. ”Input control set” refers to the generation based on the original input alignment, otherwise control alignments are generated for each individual alignment window. Total training instances in the resulting sets are also listed.

		Input control set			Window control set		
data set	k	instances	AUC	F1-Score	instances	AUC	F1-Score
1	0.2	7248	0.974	0.889	7035	0.979	0.923
	0.5	7953	0.968	0.879	7980	0.984	0.924
2	0.2	9826	0.981	0.866	9675	0.989	0.957
	0.5	10553	0.983	0.868	10582	0.991	0.955
3	0.2	10178	0.920	0.872	10183	0.938	0.859
	0.5	12004	0.943	0.865	12102	0.951	0.841
4	0.2	12859	0.983	0.896	12563	0.972	0.869
	0.5	13307	0.985	0.901	13498	0.972	0.869

after the generation of alignment windows. For a direct comparison, four classifiers were retrained using both methods. The training data used was identical to the one in the previous experiment. A direct comparison of performance can be reviewed in table 2.

It can be observed that there is only a minor difference in AUC between both methods. However, for data sets 1 and 2, there is a significant increase in the F1-Score when switching to the alignment window generation method. The reverse can be observed for data set 4, but with a less strong impact. Comparing the number of remaining training instances after filtering, no significant difference can be noted. This alone suggests, that the overall tree edit distance offset of the control set is very similar between both methods, i.e. both approaches result in a control set that is similar in distance to the native set. Figure 2 also directly compares the tree edit distances of negative sets using both approaches. Given that the AUC does not significantly change, it can be assumed that for current practical purposes, both methods of negative set generation are sufficient. However, the on average higher F-Score as well as the overall better computational efficiency suggest that a negative set creation based on already established alignment windows is more efficient in the long run. Combining this with the results of the previous section, the most efficient overall approach to training set preparation would be the pre-selection of input alignments according to external criteria, the generation of negative instances from alignment windows and the structural conservation filtering of these windows by directly comparing them with the negative set. This way, the highest distance between negative and positive training sets is achieved with minimal computation time, thus reducing overlap between both classes. For the k-cutoff, a value of 0.2 or higher should be preferred, to reduce the number of unnecessarily removed training instances.

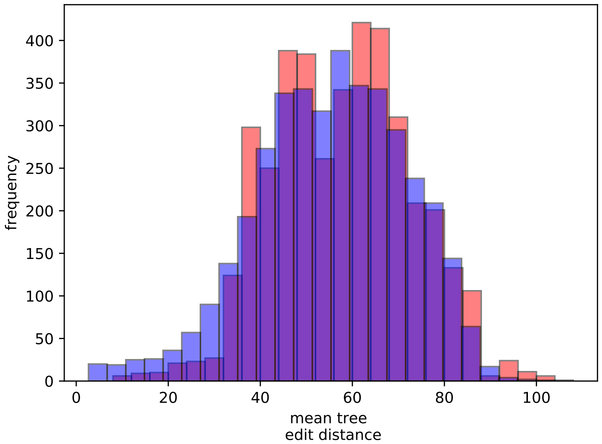


Figure 2: Distribution of mean tree edit distances of all alignment windows comparing two strategies of control alignment generation. Control alignments were generated using SISSIz either from (1) the unprocessed input alignment and then processed through the full Svhip pipeline or (2) directly from generated alignment windows.

2.3 Addendum: An experimental random forest classifier

The question has been discussed, if the data preparation pipeline implemented by Svhip is also a viable tool to train other binary classification tools. For this reason, data prepared with Svhipm has been used to train an experimental Random Forest binary classifier. As training data, the data set used to train the best performing custom SVM classifier has been reused. As a test set, the original RNAz 2.0 test set has been used for best comparability. Without further parameter tuning, an AUC score of 0.979 and an f1-Score of 0.934 has been obtained (see figure 3), which already moves the classifier into the better performing league of classifiers. In direct comparison, the original RNAz 2.0 SVM classifier achieves an AUC of approximately 0.99 on this set. It must however be noted that the overall false positive rate appears to be slightly higher with this classifier.

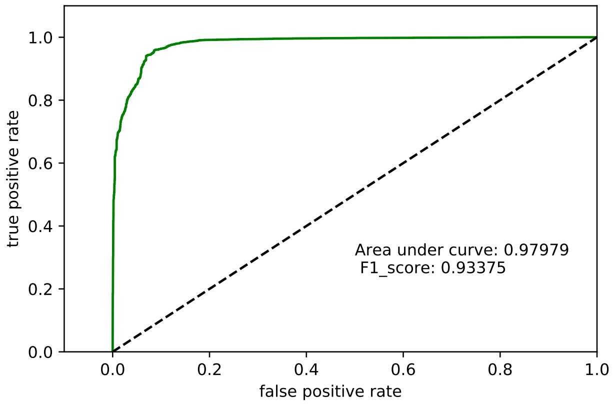


Figure 3: Performance of an experimental random forest classifier trained on alignment data prepared using the Svhip pipeline. The test set used is the unmodified RNAz 2.0 test set containing a set of well-established structurally conserved RNAs.

As a point of interest, the overall required run time using this setup is significantly smaller than with an SVM, both for training and classification. This might categorize a potential optimized random forest classifier as a viable alternative if either run time becomes an issue or a pre-screening of large amounts of data is required.

3 Methods

3.1 Screening and selection of training data

3.1.1 Preliminary data screening as only structural conservation filter

Null hypothesis: There is no observable difference in classifier performance when
(a) training data is directly generated from Rfam families showing sufficient structural conservation signals with no further filtering of alignment windows regarding structure or
(b) training data is filtered two-fold for structure by first pre-selecting Rfam families with strong structural conservation signals and then from these only selecting generated alignment windows that show significantly higher conservation when directly compared to a new set of control alignments

To test the hypothesis, a set of 4 new classifiers was be trained, mirroring the Rfam family composition of the 4 best performing classifiers trained for the master thesis. As families have already been investigated for their overall degree of structural conservation for this thesis, the pre-screened Rfam data base were used for this experiment. For the first part of the experiment, structural conservation filtering for individual alignment windows was be deactivated in data preparation while otherwise retaining training set composition. Even though a correlation between a significant increase in training set size and lower overall accuracy was suggested in the thesis, the effect here should be small enough to not falsify results. The trained classifiers were be evaluated against their counterparts trained with the default pipeline by comparing performance on a test set (10 families per set) sampled from the screened Rfam alignment data. If a significant offset in accuracy can be observed, then it had to be assumed that contrary to the null hypothesis the selection of a one- or two-step structural filtering approach has indeed an impact on overall accuracy.

As a secondary evaluation of training data selection, it has been suggested to reevaluate the cutoff used to determine if a given alignment window shows high structural conservation. Individual alignment windows are evaluated by using the mean tree edit distance as a metric to approximate structural conservation between sequences in comparison to a control alignment. Currently, alignments are considered highly conserved if their mean tree edit distance is $\leq$ the 10-percentil of the entirety of all control mean edit tree distances. In testing, this has been showing to drastically trim the space of potential training candidates. For this reason, it has to be tested how severe the impact of loosening this cutoff is in praxis.

Null hypothesis: There is no correlation between cutoff for structural conservation estimation and classifier performance.

The best performing classifiers were retrained, with the selection process changed to also include alignment windows with mean tree edit distances less than or equal to the [10, 20, 30, 40, 50]-percentile of the control set. Performance was be evaluated on the test set used in the previous experiment. It should be noted that a 50-percentile inclusion of alignment windows implies, given that the distribution of tree edit distances seems to approximate a normal distribution, that even with a near total overlap of mean edit distance distributions still around 50% of alignment windows would be included in training.

3.1.2 Generation of control alignments

Null hypothesis: There is no observable difference in classifier performance if
(a) control alignments are generated directly from the input alignment and then subsequently screened and
(b) control alignments are generated independently from alignment windows.

Again, the best performing classifiers were be retrained, using two different approaches for control set generation. Other parameters of the pipeline were not changed. Exactly one control alignment based on each individual alignment windows was be generated where applicable, i.e. 1 for each valid, non-empty window. This approach results in the set of control alignments and alignment windows generated from input being equal in size. Performance was evaluated by comparing accuracy on the same test sets as before.

3.1.3 Run time analysis - not yet finished

Five classifiers with a training set size of 10, 20, 30, 40, 50 Rfam families will be chosen. Then, run-time will be measured individually for the following segments of the data processing pipeline:
(1) Alignment data processing, (2) Structural conservation validation, (3) Feature vector calculation, (4) Hyper parameter optimization, (5) SVM training.
Run time will be directly compared and it will be attempted to estimate the correlation between input size and run time.