**nDNA prot:**

1. In the feature extraction step, the authors group the input dimensions into 3 bivalent classes, for each physico-chemical attribute. They first derive the frequency of each of these bivalent classes from the entire protein sequence, and then again divide the protein into 5 equal parts (why equal? why 5? and why not split based on known functional regions like N,C termini; or characteristic Amino-Acid clusters?). Furthermore, after calculating the distribution frequency of each of the 3 bivalent classes in each of these 5 parts, what is the justification for including this set of frequencies back with the class frequencies from the entire protein, for the feature space construction? How would we expect these ‘sub frequencies’ to correlate to the ‘entire protein frequencies’, and shouldn't there be weight compensation for the feature set of ‘entire protein frequencies’ in our feature space? Adding back the sub-protein frequencies and considering them at an equal weightage as the whole-protein frequencies when learning the weights for the base classifier, seems to be nudging towards redundancy and confounding the learned weights of the final ensemble classifier.
2. An ensemble classifier works well only when the individual classifiers are each highly accurate, and make errors in different data spaces (otherwise it’s just overfitting to your training data, as per my understanding). The selection of 5 classifiers from an original set of 16 that they list in the paper, does not seem to be reasonably described or justified. Is it fair to use the prior performance of a classifier (on entirely different datasets, possibly with different kernel functions) as a measure of it’s ‘quality’, and especially when said classifiers are being used in conjunction, as in ensemble learning?

*This last question is just a point of confusion I had regarding the data cleanup. I wasn’t sure if it was justified to ask the presenter to defend the choices of the authors, but thought it would be interesting to know your/other peoples’ thoughts on this.*

iii) In the data pre-processing, the authors remove proteins with sequence length > 6000. Their excuse for this is that proteins with length > 6000 AA may be too complex. However, in light of the fact that a majority of the protein structure is often for scaffolding, whereas the active domain is actually limited to a small number of Amino Acids, this does not appear to be a reasonable justification. What would be another justification for excluding proteins greater than a certain length? I figured that disparities exist in protein length between different functional classes of proteins (DNA binding proteins ought not get too bullky), but then, large proteins also have multiple functional domains - ex. zinc fingers. Excluding large proteins will be a gross oversight in such cases, ex. with CBP acetyltransferase protein (although, this one is ~2500 bp). In continuance to this question, the authors then remove similar Protein Families (PFAMs) based on similar sequences - how sound is this when you first remove a set of proteins (>6000 bp, <50 bp) already from each PFAM, and then consider which of these reduced set PFAMs are similar? I would be quite skeptical of any conclusions about protein functional classifications that they draw from this set, especially when they give no direct mention of the number of proteins removed by the protein length filter.

**iDNA-prot|dis**

1. The reference that the authors used for their reduced alphabet sets (the characteristic ones for DNA binding proteins) also used SVMs with RBF kernel. I am wondering why a two-layered SVM with RBFs, or alternatively, a neural net, might not have been employed (with the base SVM deriving from the full protein sequence, and learning weights not just of the 20 input features (Amino Acids), but also the weights for interactions between the input features. There is an example of this in image recognition, using SVMStruct. I wonder what might be the restrictions in using such an approach to answer the research question at hand.
2. If this approach were to be extended to multi-class classification of different protein classes, how effective would it be? In my mind, not so much, since there might be overlaps between the reduced alphabet sets for protein families that have different functionalities but similar physico-chemical features. A big barrier might also be finding the characteristic reduced alphabet sets for each protein family. As a side note, is it also a valid approach to say that one ought to focus solely on the active domain regions of a protein for training such a model?