

Innovation Track Report

Introduction

I propose a model (neural network) that uses only plasmid DNA sequences as predictors (inputs) to produce (perform classification) a listing of most probable lab origins (output). This is beneficial for the investigation process since the knowledge about biological design functionality (like being drug-resistant or growing characteristics) might not be obvious/available/reported in the first place, it can also be intentionally left out, or even deceptively reported (for example, samples smuggling, or even stealing).

If these inputs are not known but needed in the attribution model then the investigator will have to spend more time and much more resources (translating as costs, and in some cases as growing risk of unacceptable failure as time runs out) to get the job than - they will need access to real sample (and a lab) and not just the sequence record. Also, if we want to be socially responsible with attribution, the model itself must be useful like scientific skepticism (not trusting everything you hear or read), in other words, not depending on anybody's verbal inputs or claims about the sample. If there is something than it is in the sample and will be found in the sample.

The model is a convolutional neural network (CNN), relatively simple in design - it is a serial neural network. The network is designed with attribution in mind. The design goal is a network that learns to identify or respond to the origin-revealing-features where ever they are in a sequence and whatever the composition of the sequence might be.

The trained model has high accuracy for a single prediction (TOP-1) and not just for TOP-10. And that is a good thing and not just because the first suspect is most likely the right one, simplifying the investigation, or because is obvious how the assemble of models (combining results of different models to improve accuracy) improves that TOP-10 prediction, or because you now know where is best to start if you don't have enough resources to engage all TOP-10 simultaneously (it is not a same to write the warning letter to some companies, or send disqualification notice to some students in the competition, or make a call to the insurance company, or to trigger something that is closer to a national emergency). This is in first place desirable because the model or assemble is not just telling how highly sure it is about the target but is also able to rapport why such classification is made, and that is the real power of the model.

Besides the needed classifications (lab listing) proposed model also reveals a part (or parts) of a sequence that is the reason for assigned classification. It can find that little piece of a needle in a haystack (using network explainability and interpretability techniques), showing what is the evidence, and where it is in the sequence. This also improves human trust in the received result (lab listing) besides giving the investigators needed evidence for the origin of a sample. An investigator can trust a result since they can check out (seeing is believing) how much of discovered about the sample using the model is grounded in reality (for example, is the discovered part of sequence really a property of that lab).

Here investigators can be university, laboratory, a pharmaceutical company, government, insurance companies, patent offices, military (biochemical warfare), and security organizations (biological terrorism), anybody that is interested to know more about what they have or somebody else has. All that is needed to perform this is a sequence, that AGGGT... string, or even a part of a sequence. Model is trained on sequences of different lengths sacrificing somewhat of classification accuracy but improving model robustness against variations (natural/intentional/deceptive) in sequence length and composition and even against sample presentation resolution variations which are analog to the presence of small sequencing errors (like missing parts of a sample or erroneous reading). This is achieved through a network structure and using sequence feature engineering inspired with signal processing techniques (wavelets).

I have approached sequences as signals or instructions characterized not just with discrete values, or frequencies, and decompositions, but characterized with the meaningful grouping of values (patterns) that network needs to find inside the sequence. Using the scalograms came as the obvious choice for "signal" representation since able to capture the inner arrangements and not just how much of A, T, G, C, N is there and how often.

Scalogram is a meaningful sequence representation for the machine, and for humans not so much, you will have to look quite a number of this first. The scalogram is the absolute value of the continuous wavelet transform (CWT) of a signal, plotted as a function of time/samples and frequency. If you are not coming from signal processing it is not easy to recognize what statistical dependencies are captured only by looking. The good thing is that we can know to what part of sequence the observed/reveled scalogram pattern belongs to. We can even use these visual patterns to classify elements in the sequence and possibly find new ways to describe sequences (like sort of statistically recognizable n-mers or codons). But here this scalogram revelation of patterns

is what we call the feature extraction, or more correctly, the sequence feature extraction and the feature composing in a form of an image (scalogram).

Even if the true origin of the sample (laboratory) is not known to a model, the model will deliver the most interesting part of the sequence, for which believes that is of interest, and the most likely candidates with one distinct variation in the score for a class that I have included in the model as “unknown” or simply class “0” (model have $1314 + 1 = 1315$ classes). Scores for class zero in the general cases are more like Femto-level noise than something that really receives network activations in the output layer. This is a way that network could signal that something is different.

Higher scores for class “0” can indicate that lab is not known to a model-this is just a possibility that requires more investigation since is tested on a small number of samples (I simply excluded the labs that appear only once from training set and tested on them after, but this is not really something to take as proven to hold as a general case). On the other side, activations for the “unknown” class will reveal what part of the sequence is interesting to a network. In other words, you can ask, if the lab is unknown, what is (according to the model) the most interesting part of the sequence that we should look at first. This is the same procedure as asking the model what part of the sequence contributed to a particular class score.

In the next sections, all of this is given in more detail. Improvements are also discussed in this proposal since none of this are used in the actual competition- due to limited resources and time that I have. Models, functions, code including code for rapport figures are also given in appendix section, and you can find them in attachment.

Neural network description

Model is CNN with 24 layers, spatial input is of $128 \times 128 \times 3$ (scalograms size), it uses large convolution filters (5x5) to capture a larger amount of color information from scalogram, following max-pooling layer that reduces spatial input to a next layer to half. This is repeated several times through the same structures until a spatial input reaches 8×8 . The global average pooling layer averages this to $1 \times 1 \times 768$. Using the global average pooling layer, the final classification output is only sensitive to the total amount of each feature captured in the scalogram image but insensitive to spatial positions of the features. And finally, there is a fully connected layer and classification layer (with extra class “0”).

Finding the evidence. Sequence of interest. Result interpretation

The technique is based on class activation mapping (CAM). This is used to investigate and explain the predictions of a deep convolutional neural network for a given image (here scalogram image as the sequence representation). This is a known technique for the image classification networks, and since sequences classification problem is solved by transforming sequences into images (problems are often solved easy once mapped into different space) we can use this technique to identify what part of the scalogram contributed the most to prediction and from this “hot” zones what sequence sections are of interest.

Deep learning networks are often considered to be “black boxes” that offer no way of figuring out what a network has learned or which part of an input to the network was responsible for the prediction of the network. And when models fail and give incorrect predictions this is often spectacular and without warning or explanation. Class activation mapping [1] is one technique that can be used to get visual explanations of the predictions of convolutional neural networks. Incorrect, seemingly unreasonable predictions can have reasonable explanations.

Using class activation mapping, you can check if something specific as part of a sequence “confused” the network and led it to make an incorrect classification. For example, competition developed model really “likes” confusing I7FXTVDP with RYUA3GVO class so much, that one improvement for TOP-10 accuracy would be to include RYUA3GVO on the list every time when I7FXTVDP is listed as output and opposite (this is not used for the competition submissions, only pure predictions). Class activation mapping identifies bias in the training set and increases model accuracy.

In case the network bases predictions on the wrong features, then you can make the network more robust by changing the image resolution and size in order to better capture the data in the images form. For example, during model development it is observed that network precision increases for larger size images, meaning that less distortion to statistical dependencies (that we can see as colorful scalogram patterns) will be present if the larger image is used. Image of 128×128 used here is a compromise between available resources (out of memory trouble) and accuracy. (Yes, this can be solved using the data-store approaches, however this is significantly slower than reading from memory and I could not spare that much time. Model definitely have room for improvement).

This example class activation map shows which regions of the input image contribute the most to the predicted class mouse (see Figure.1). Red regions contribute the most.

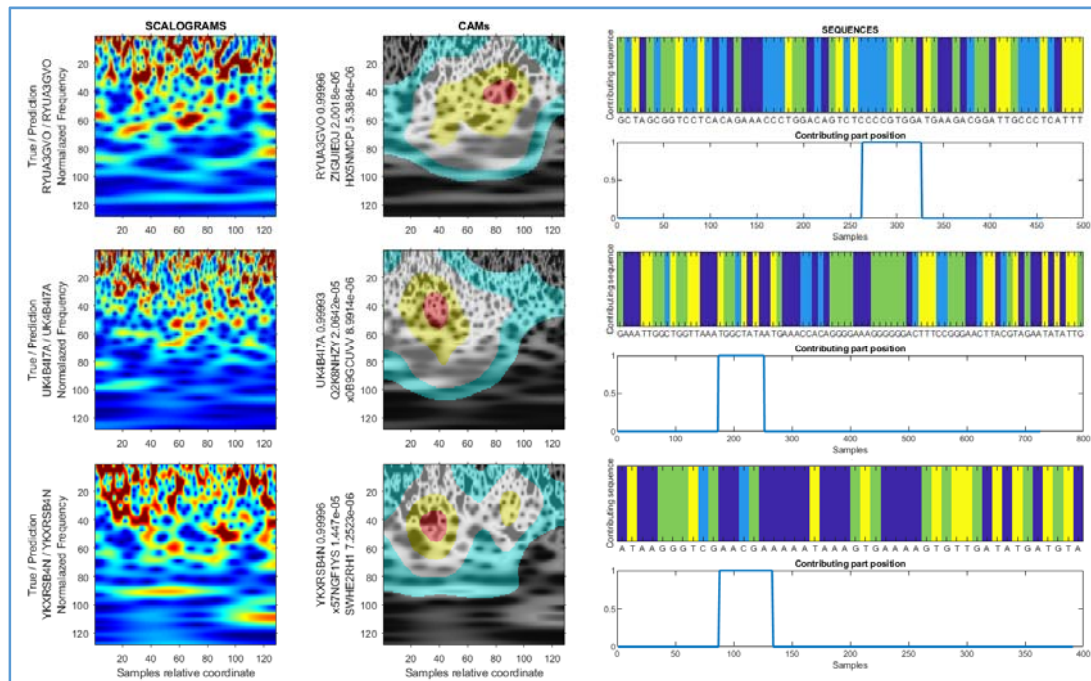


Figure 1. Scalograms are shown on the left side. True/prediction class is given as their titles. In middle are CAM maps, shown as masks burned over black and white scalogram images. Prediction and activation scores for TOP-3 labs are given. And on the right are extracted contributing sequences shown as sequences of color and GCTA ticks on the x-axis, and their position within-sample is shown with a contributing part position plot.

The class activation map for a specific class is the activation map of the ReLU layer above the final convolutional layer weighted by how much each activation contributes to the final score of that class. Those weights equal the weights of the final fully connected layer of the network for that class.

These class activation maps can be generated for any output class (including "unknown" or zero class). For example, if the network makes an incorrect classification, you can compare the class activation maps for the true and predicted classes. Figure.1 shows the class activation map for the predicted class with the highest score. The class activation map in the image shows the contribution of each region of the input image to the predicted class. Red regions contribute the most. The network bases its classification on the longer part of the sequence, but the strongest input comes from the red areas – that is, this could be the “smoking gun”. Since this corresponds directly to the sequence (x-axis is representing samples on the known scale) we can identify a part of the sequence that this region belongs to.

That the network correctly identifies the sequence of interest we can verify by comparing the identified sequence with known examples. Since is not revealed what sequence part (or a pattern) belongs to a particular laboratory, I demonstrate this by measuring cross-correlation between identified sequence parts for the same laboratories. As shown in Figure.2 we can see that network really is capturing features that are specific to the laboratory.

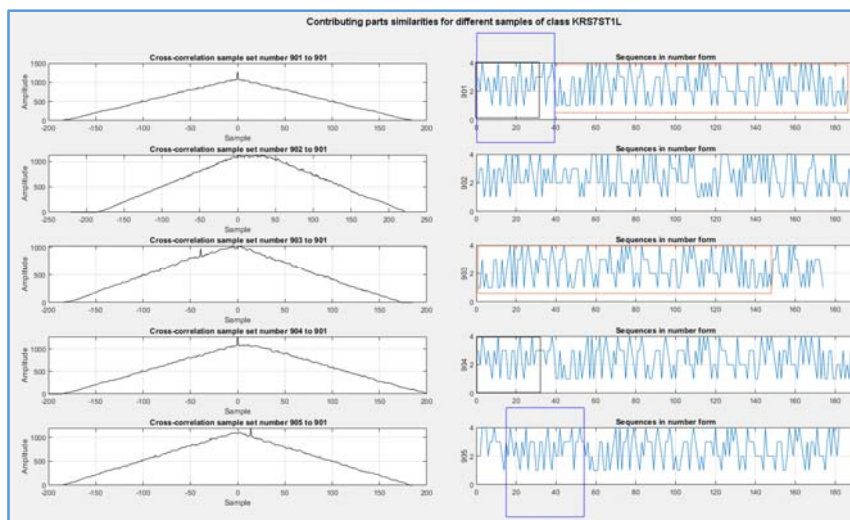


Figure 2. On the left are cross-correlation scores and on the right side are extracted contributing sequences. The first sample (901 in the training set) is compared to himself and the rest (902-905) also with extraction for sample 901. They all belong to the same class KRS7ST1L. Spikes indicate that signals have similarities, and we can identify the positions of these sections. Other techniques for this also apply. I have placed several window annotations to the point where extracted sections have identical sequences. The longest identical sequence is for 903, shortest for 902 (902 don't have a clear spike, and have several identical "signals" made of up to 4 matching ACGT interrupted with one or couple of elements in-between, this also indicates that the network could benefit from improving scalogram image resolution), and 904 and 905 have identical parts as the first part of 901. All sample's model classifies correctly.

Model improvement

As mentioned above, the model is using low image resolution and will benefit significantly from the use of larger images (jumping from 96x96 to 128x128 improved validation accuracy by additional ~6% points, pushing the model above the BLAST benchmark). A further improvement is to implement padding (extending sequence with zeroes so that all of them have the same length) so that CWT is performed on the same scale.

Padding is not used for the competition model, because the sequence can be really large, and the larger the sequence more time is required to produce a scalogram. Also, I believe that this stretching and compressing with different scales to a small image contributed to the model robustness against variations in sequence length and composition as sources of model uncertainty-serving as sort of augmentation. In the padding approach, the network will also benefit from sequence scalogram augmentation (actually in both, with padding or without padding), which should be performed using the number of voices per octave in cwt rather than the usual stretching and shifting of image. This changes the signal resolution and the number of details that will be captured on scalogram forcing the network to discard features potentially leading to network overtraining. This augmentation through cwt resolution is not used in the competition developed model but is verified that network accuracy changes if the input image has a different cwt resolution than those on which is trained. However the accuracy is quite robust against this, and surpassingly considering the accuracy of the TOP-10 can be even better for a resolution (number of octaves) for which network is not trained. Because of this I also consider using this as a replacement for the assemble approach, using different predictions from different resolution images (like their scaled averages) to improve the model TOP-10 accuracy. The developed neural network is not large (comparing to some other image classification networks) and assemble is of acceptable size even if more networks are added, but having a single network in the model means even a smaller model, and that is a good thing for implementation (unfortunately I have tested this only on validation set, and not through the test submission).

Another thing to implement is to use classes balancing for the output layer and to test additional network structures and hyper-parameters (to optimize the network for speed and precision) since I have not performed any of this. I simply found a configuration that learns and trained a couple of models with different hyperparameters to use them in assemble to improve predictions.

Conclusion. Simple, clear, useful, implementable, informative, investigation neural network.

REFERENCES

[1] Zhou, Bolei, A.Khosla, A.Lapedriza, A.Oliva, and A.Torralba. "Learning deep features for discriminative localization." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921-2929. 2016.