Computational Systems Biology
Deep Learning in the Life Sciences

6.802 6.874 20.390 20.490 HST. 506

Kfir Schreiber
03/23/2017

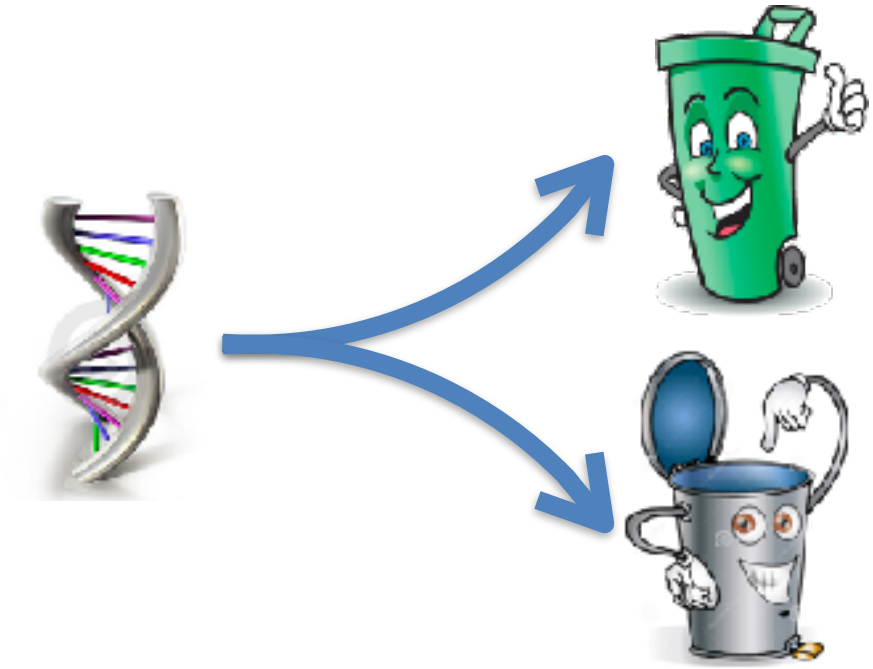
PEDLA: predicting enhancers with a deep learning-based algorithmic framework

Feng Liu, Hao Li, Chao Ren, Xiaochen Bo & Wenjie Shu

http://mit6874.github.io

The problem

Classify genomic segments to enhancer vs. non-enhancer classes



Main claims

- *PEDLA is a generalizable, state-of-the-art enhancer prediction framework, based on heterogeneous and unbalanced data*
- *PEDLA demonstrates state-of-the-art performance over a variety of cell types/tissues, in a single, relatively small model*
- *This work develops a training technique that deals with class-imbalanced data in an unbiased manner*

Data and representation

- Data was derived from the ENCODE Project and the Roadmap Epigenomics Project

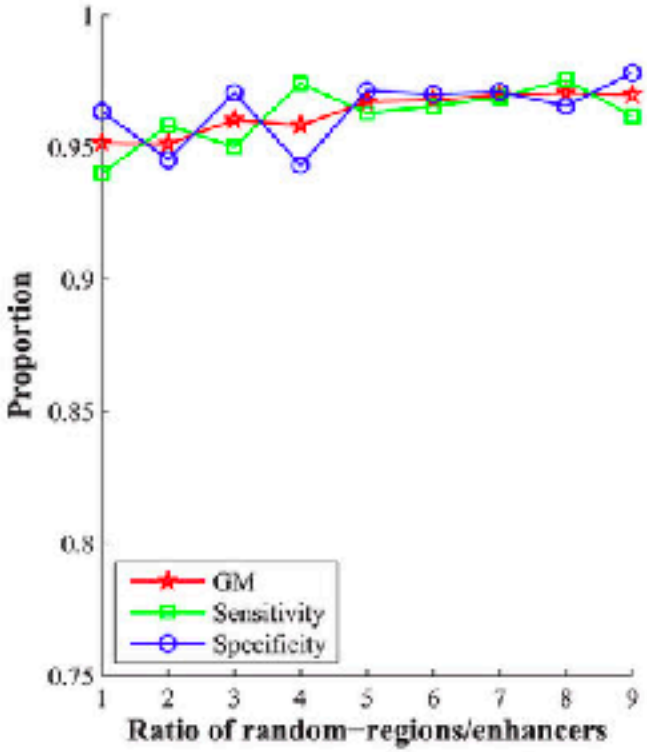
- The training set included 22 cell types/tissues, and a test set 20 other cell types/tissues

- 1114 dimensional representation consisted of:

  histone modifications, TFs and cofactors, chromatin accessibility, transcription, DNA methylation, CpG islands, evolutionary conservation, sequence signatures, and occupancy of TF binding sites (TFBS)

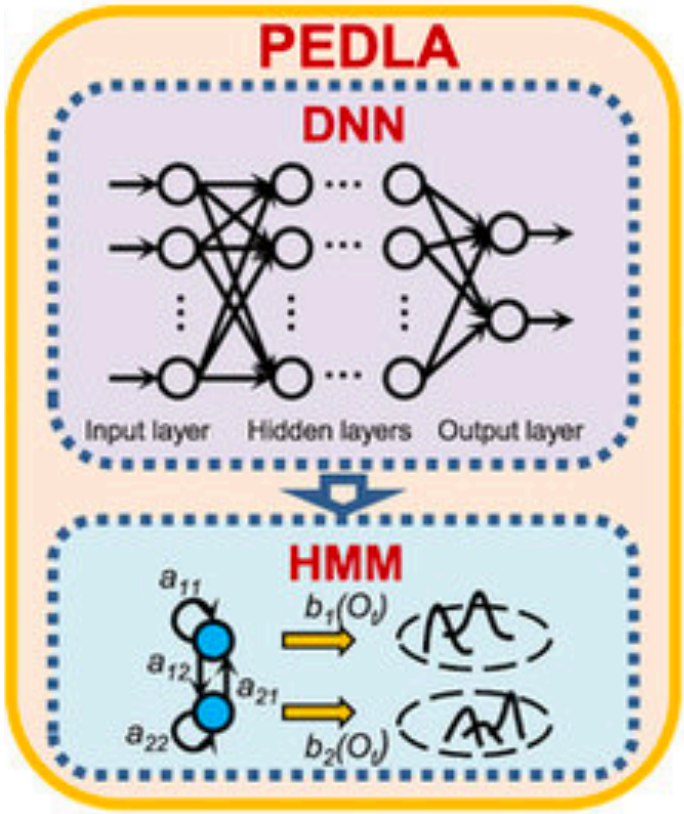
Positive & Negative datasets

- A key point in the success of PEDLA is the ability to work with highly unbalanced data
- The positive set was build using histone marker H3K27ac
- The negative set was combined from promoters and random unannotated genomic regions, and designed to maintain a ratio of 1:1:x for the enhancers, promoters and random regions respectively

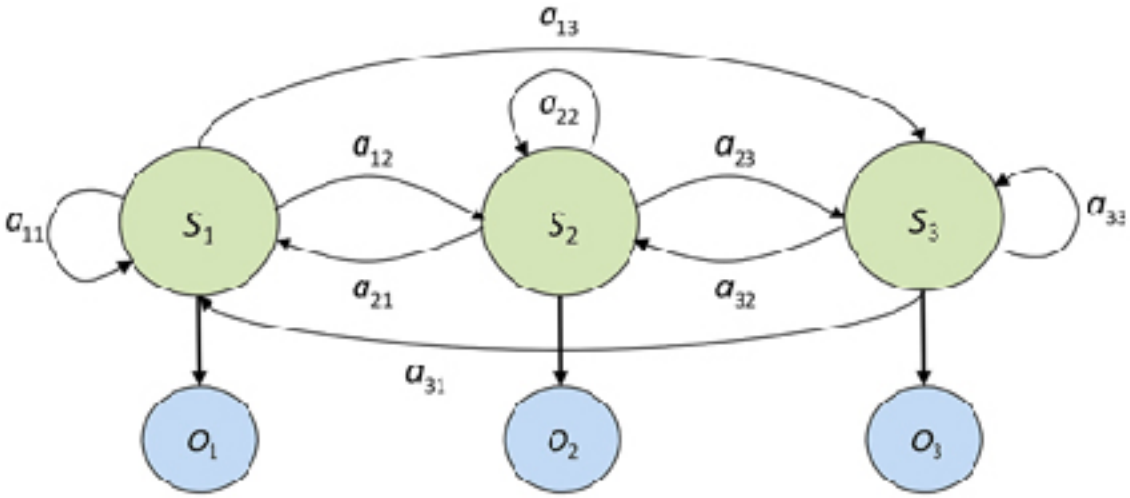


Method

- PEDLA uses a hybrid DNN-HMM model, with 2 hidden layers of size 500
- In total, 4400 models were trained - 22 cell types, 50 permutations of the cell types, and 4 permutations of the data within cell type



Hidden Markov Model



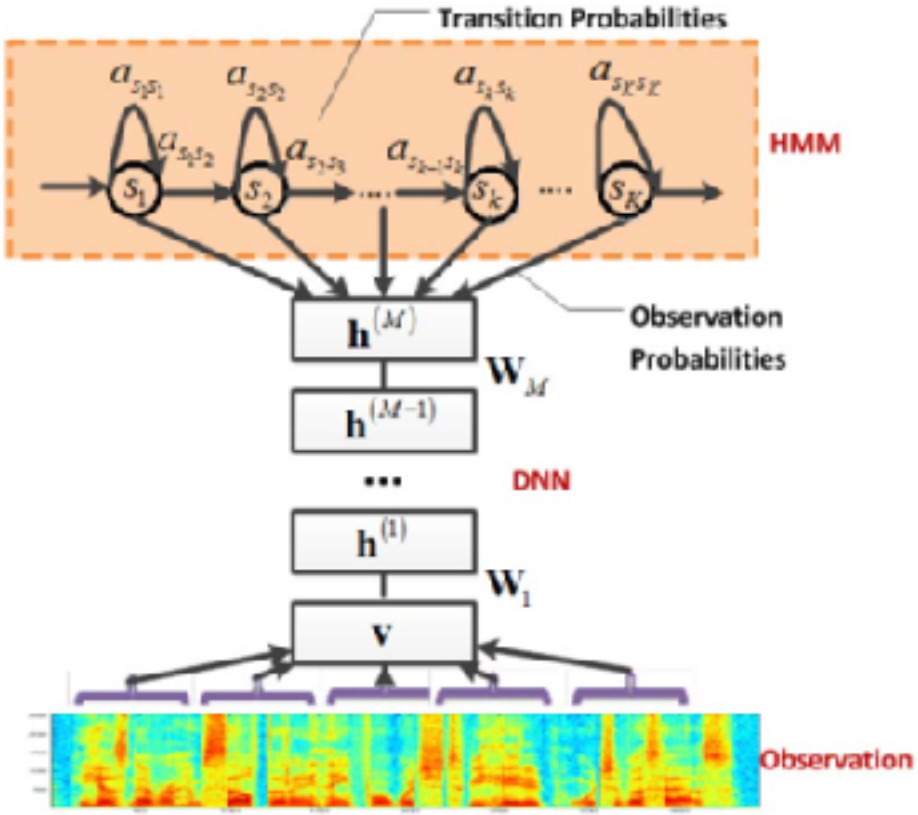
- In a hidden Markov model the observed data is assumed to arrive from a Markov process, where the next state only depends on the current state and a random seed
- HMM is parametrized by the triplet (π, A, B)
- Given the observations, the algorithm tries to predict the optimal state path that derived the observation

Deep Neural Network

- Our DNN is constructed of several stacked RBMs and a softmax layer
- At each layer the hidden layer of one RBM serves as the visible layer for the next RBM
- The softmax takes as input the output of the final RBM

Hybrid DNN-HMM

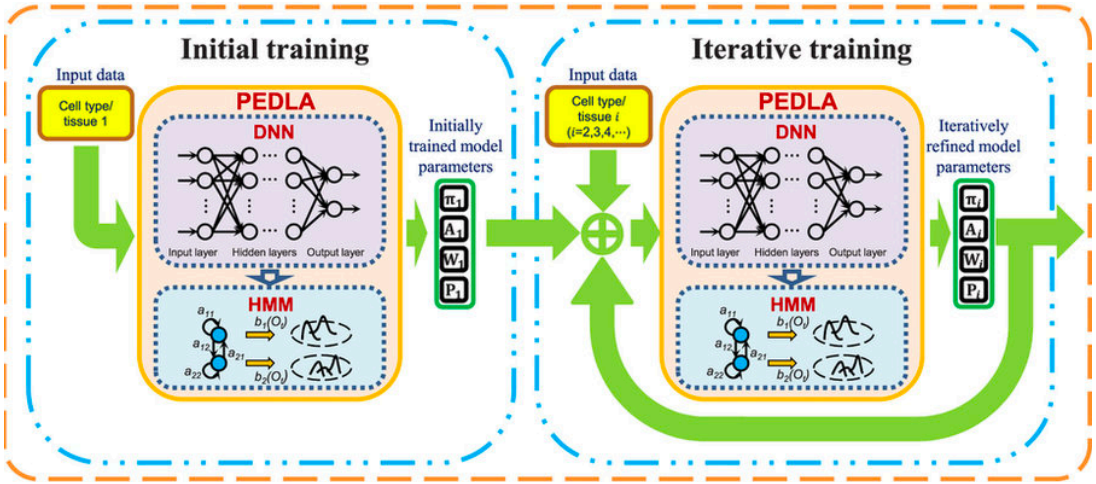
- Observations are fed into the DNN
- The DNN output is viewed as the posterior probability of each state
- Using Bayes' law is used to calculate B
- Finally, the Viterbi algorithm predicts the final state



Training

- Training PEDLA was done in two parts - initial training and iterative training

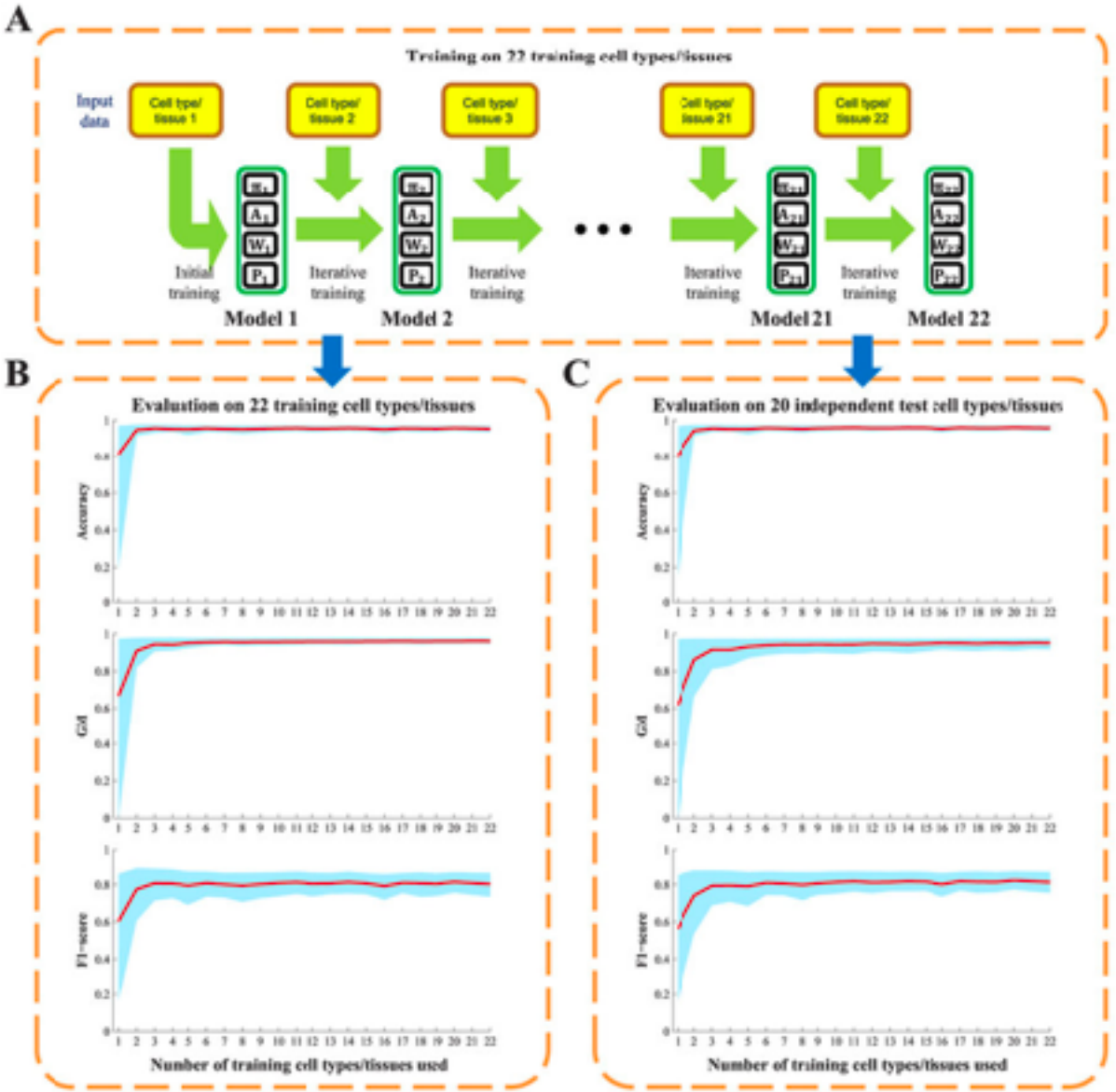
- At each step the model was trained on a single cell type/tissue, and the output (posterior parameters) of that step was set as the initialization for the next cell type



Initial training

- The DNN was pre-trained in a greedy, layer-by-layer, un-supervised way, and then refined in a supervised way
- The HMM was initialized using an estimate of π and A
- The complete model was trained using data from the first cell type/tissue

Iterative training



Results

- *Generally, PEDLA achieved state of the art results on most metrics throughout many different tests and cell types*

- *Performance metrics - accuracy, sensitivity, specificity, GM, F1-score, validation rates of distal DHSs, P300, and TFs, and misclassification rate*

Metrics

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

$$Sensitivity = TP/(TP + FN)$$

$$Specificity = TN/(TN + FP) = Recall$$

$$GM = sqrt(Sensitivity * Specificity)$$

$$Precision = TP/(TP + FP)$$

$$F1score = 2/(1/Recall + 1/Precision)$$

Results

			PEDLA	RFECS	CSI-ANN	DELTA	ChromHMM	Segway	PEDLA (all features)
Number of prediction			22691	75084	30173	112044	26869	131698	20689
Performance metrics	Accuracy		96.30%	93.67%	95.58%	87.78%	94.03%	91.01%	97.65%
	Sensitivity		95.72%	64.19%	65.50%	73.56%	37.67%	12.89%	96.16%
	Specificity		96.37%	97.89%	98.63%	89.84%	99.75%	98.94%	97.80%
	GM		96.02%	79.26%	80.34%	81.29%	61.30%	35.71%	96.97%
	F1-score		83.01%	71.71%	73.06%	60.40%	53.74%	20.90%	88.31%
	Validation rate	DHS	40.68%	31.85%	30.65%	12.25%	38.86%	40.61%	42.29%
		P300	15.25%	7.26%	10.83%	1.57%	9.89%	3.52%	16.82%
		TFs	28.89%	17.71%	19.72%	5.75%	19.14%	6.42%	32.37%
	Misclassification rate		7.53%	3.09%	16.46%	3.01%	6.42%	14.53%	6.59%

Comparison of the performance of PEDLA with that of existing methods

Key Claims

- This work presents a deep-learning framework that incorporates heterogeneous, imbalanced, multi-dimensional data and achieves state-of-the-art results across cell types and metrics

- PEDLA has the ability to generalize to new cell types and prediction of many functional elements

Analysis

- The paper provides plenty of support to the main claims - comprehensive comparison to other methods, verity of statistical metrics and different datasets

- A significant weak point of the work (and all the other methods) is the difference between the good performance on most metrics and the validation rate

- CNN / RNN?

Summary

- PEDLA is an interesting piece of work that handles some profound problems in machine learning
- The paper shows great results with massive statistical support
- NLP influenced RNNs and CNNs might prove to be beneficial to the problem

Questions?

Thank you!