A DEEP NEURAL NETWORK WITH A RESTRICTED NOISY CHANNEL FOR IDENTIFICATION OF FUNCTIONAL INTRONS

Alan Joseph Bekker[1] *Michal Chorev*[2] *Liran Carmel*[3] *Jacob Goldberger*[1]

[1]Engineering Faculty, Bar-Ilan University, Israel
[2]IBM Research - Haifa, Israel
[3]Department of Genetics, The Alexander Silberman Institute of Life Sciences,
The Hebrew University of Jerusalem, Israel

ABSTRACT

An appreciable fraction of introns is thought to be involved in cellular functions, but there is no obvious way to predict which specific intron is likely to be functional. For each intron we are given a feature representation that is based on its evolutionary patterns. For a small subsets of introns we are also given an indication that they are functional. For all other introns it is not known whether they are functional or not. Our task is to estimate what fraction of introns are functional and, how likely it is that each individual intron is functional. We define a probabilistic classification model that treats the given functionality labels as noisy versions of labels created by a Deep Neural Network model. The maximum-likelihood model parameters are found by utilizing the Expectation-Maximization algorithm. We show that roughly 80% of the functional introns are still not recognized as such, and that roughly a third of all introns are functional.

Index Terms— intron function, uncertain labels, noisy labels, Semi-supervised

1. INTRODUCTION

Most human genes are built from protein-coding segments intervened by noncoding elements known as introns [1]. Whereas many introns probably spread throughout eukaryotic genomes as slightly deleterious elements, there are many documented instances of introns that carry out critical cellular functions [2]. Many intronic functions relate to mRNA processing, but also to mRNA shuttling, quality control, and more [2]. This makes the identification of functional introns a fundamental issue in functional genomics, that may be imperative to our understanding of cellular processes and disease [3]. However, currently there is no general method to tell functional introns from non-functional ones, and all functional introns have been identified based on anecdotal studies. As a result, there is a relatively small subset of introns that are known to be functional. It is not known whether the other introns are functional or not [4].

This study was design to assist genomics researchers developing a tool that can determine which introns are likely to be functional, despite the fact that their functionality has not yet been discovered. We propose a framework that accounts for the inherent uncertainty in the functionality label of introns that have not been yet documented as functional. We view introns which are not known to be functional as training examples with an unreliable or noisy label. We thus consider the task of identification of functional introns as a task of training a classifier based on noisy labels.

Whereas noise tolerant variants have been proposed for classical machine learning classifiers [5][6], there are not many studies that have attempted to address the problem of training deep neural networks algorithms with unreliable labels [7][8][9][10]. Grandvalet et al. [11] tackled the problem of missing labels in the training set as an extreme case of noisy label data. They proposed a semi-supervised algorithm that rewards the model to predict the unsupervised data with high confidence by adding a entropy regularization term in the optimization function. Natarajan and Dhilon [12] suggested a universal unbiased estimator for binary classification with noisy labels. They developed an alternative cost function expressed by a weighted average of the original cost function, and supplied upper and lower bounds for performance criterion. Sukhbaatar at al. [10] suggested adding a regularized linear layer on the top of the softmax layer, and made strong assumptions in order to prove that the proposed noisy layer can be viewed as the transition matrix between the true and observed data labels. Larsen et al. simplified the noise model by assuming a single noise parameter that can be estimated by performing a cross validation procedure. Goldberger and Ben-Reuven [13] suggested a training procedure based on adding another softmax layer to the network, that uses the output of the last hidden layer of the network to predict the probability of the label being flipped. Bekker [7] and Mnih [8] proposed a noise model that depends on the true label. However, they did not consider the unique structure of the noisy channel defined by the biological data presented here.

978-1-5090-6341-3/17/$31.00 ©2017 IEEE

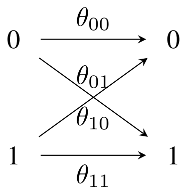

Fig. 1. Illustration of the parameters of the noisy label model.

Our approach explicitly models the intron functionality uncertainty by learning an intron functionality neural-network classifier that takes into account this uncertainty by adding an extra restricted noisy channel concatenated on top of the regular soft-max output layer.

The rest of the paper is organized as follows. First we derive the algorithm that we use to predict which intron is prone to be functional. Then we describe the data used in our experiments and then we present the obtained results and the evaluation method we applied.

2. TRAINING A DEEP NEURAL NETWORK WITH A RESTRICTED NOISY CHANNEL

Our dataset consists of introns that are known to be functional and introns which may be either functional or not. Since an intron with an unknown functionality might actually be functional, we approach this classification task as a classification model with noisy observed labels. In our model we assume that the true functionality label of the intron is not directly observed. Instead we only observe a noisy version of it. We first describe a probabilistic deep neural net model with noisy labels and then explain how the model is applied to our intron functionality discovery task.

2.1. A Deep Neural Network with a Noisy Channel

Assume we are given a binary classification problem with labels denoted by 0 and 1. In a neural net model with parameter-set w, the probability of input x being labeled as 1 is:

$$p(y = 1|x; w) = \sigma(w_o^\top h(x) + b_o) = g(x; w) \qquad (1)$$

where we denote the non-linear function applied to the input x by $h = h(x)$, $\sigma()$ is the sigmoid function and w_o, b_o are the parameters of the output soft-max layer.

We further assume that in the training process we cannot directly observe label y. Instead we can only observe a noisy version of it, denoted by z. In our case y is the correct information whether the intron is functional ($y = 1$) or not ($y = 0$). The binary label z indicates whether we know that the intron is functional ($z = 1$) or not ($z = 0$). Note that if we know that the intron is functional then, of course, it is indeed functional. However, if we have no information on the intron's functionality (i.e. $z = 0$) the intron can be either functional or not. Below we assume a simplified noise model where the noisy label is a stochastic function only of the true label. Formally, the noise model is defined by a parameter-set θ such that $\theta_{ij} = p(z = j|y = i)$ is the probability of observing label j given that the true label is i (see Figure 1).

The combined neural-net model with noisy labels, therefore, is:

$$\begin{aligned} p(z = j|x; w, \theta) &= \sum_{i=0,1} p(z = j|y = i; \theta)p(y = i|x; w) \\ &= \theta_{1j} g(x; w) + \theta_{0j}(1 - g(x; w)). \qquad (2) \end{aligned}$$

The model is illustrated in Figure 2.

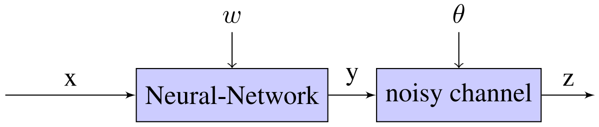

Fig. 2. The network architecture for producing estimation of the observed noisy labels.

Assume we are given n feature vectors $x_1, ..., x_n$ with corresponding noisy binary labels $z_1, ..., z_n$ which are viewed as noisy versions of the hidden labels $y_1, ..., y_n$. The log-likelihood function of the model parameters is:

$$l(w, \theta) = \sum_t \log p(z_t|x_t; \theta, w) = \qquad (3)$$

$$\sum_t \log \sum_i (p(z_t|y_t = i; \theta)p(y_t = i|x_t; w)).$$

The goal of the training procedure is to find the noise parameter θ and the neural-net set of parameters w that maximizes the likelihood function. Since the random variables $y_1, ...y_n$ are hidden, to solve this maximization problem we need to apply the Expectation-Maximization (EM) algorithm [14]. The EM auxiliary function is:

$$\begin{aligned} &Q(w, \theta, w_0, \theta_0) \qquad (4) \\ &= \sum_t E_{p(y_t|z_t, x_t; w_0, \theta_0)} \log p(z_t, y_t|x_t; w, \theta) \\ &= \sum_t \sum_{i=0,1} c_{ti}(\log p(z_t|y_t = i; \theta) + \log p(y_t = i|x_t; w)) \\ &= L_1(\theta) + L_2(w) \end{aligned}$$

where θ_0 and w_0 are the current parameter values and

$$c_{ti} = p(y_t = i|z_t, x_t; w_0, \theta_0)$$

is the posterior distribution of the true label y_t given the feature vector x_t and the noisy label z_t. Given the auxiliary function, we can easily derive the steps of the EM algorithm.

E-step:

For each $t = 1, ..., n$ and $i = 0, 1$ compute:

$$c_{ti} = p(y_t = i|z_t, x_t; w_0, \theta_0) \tag{5}$$

$$= \frac{\theta_0(i, z_t)p(y_t = i|x_t; w_0)}{\sum_{k=0,1} \theta_0(k, z_t)p(y_t = k|x_t; w_0)}.$$

The probability c_{ti} is an estimation of the hidden label y_t given the feature vectors x_t and the noisy label z_t, based on the current parameter values θ_0 and w_0.

M-step:

Based on the EM auxiliary function in equation (4), to find the updated values of the noise parameter θ we need to maximize the following function:

$$L_1(\theta) = \sum_t \sum_{i,j} 1_{\{z_t=j\}} c_{ti} \log \theta_{ij}. \tag{6}$$

such that $\theta_{ij} \geq 0$ and $\theta_{00} + \theta_{01} = \theta_{10} + \theta_{11} = 1$. This maximization problem has a closed form solution. It can be verified that the updated θ is:

$$\theta_{ij} = \frac{\sum_t c_{ti} 1_{\{z_t=j\}}}{\sum_t c_{ti}} \qquad i = 0, 1 \qquad j = 0, 1 \tag{7}$$

To find the updated parameter w we need to maximize the following objective function:

$$L_2(w) = \sum_t \sum_i c_{ti} p(y_t = i|x_t, w) \tag{8}$$

$$= \sum_t c_{t0} \log(1 - g(x_t; w)) + c_{t1} \log(g(x_t; w)).$$

This is actually a weighted version of the score function of a binary classification neural net. We can optimize this neural network using the standard back propagation algorithm. The partial derivatives of the score function with respect to the sigmoid output parameters are:

$$\frac{dL_2(w)}{dw_o} = \sum_t h(x_t)g(h(x_t); w) - c_{t1}) \tag{9}$$

$$= \sum_t h(x_t)(p(y_t = 1|x_t) - p(y_t = 1|x_t, z_t))$$

and

$$\frac{dL_2(w)}{db_o} = \sum_t (g(h(x_t); w) - c_{t1}) \tag{10}$$

$$= \sum_t (p(y_t = 1|x_t) - p(y_t = 1|x_t, z_t)).$$

Table 1. The Restricted Noisy Labels Network (RNLN) algorithm.

Input: Intron feature vectors $x_1, ..., x_n \in R^d$ with corresponding intron functionality labels $z_1, ..., z_n \in \{0, 1\}$.
Output: Neural-network parameters w and noise parameters θ.

The EM Algorithm iterates between the two steps:

E-step: Estimate true labels based on the current parameter values (5):

$$c_{ti} = p(y_t = i|x_t, z_t; w, \theta)$$

M-step: Update the noise parameter θ:

$$\theta = \theta_{1,1} = \frac{\sum_t c_{t1} 1_{\{z_t=1\}}}{\sum_t c_{t1}}$$

and train a NN to find a parameter-set w that maximizes the following objective function:

$$L_2(w) = \sum_t (c_{t0} \log(1 - g(x_t; w)) + c_{t1} \log(g(x_t; w))).$$

2.2. A Neural Network for Identification of functional Introns

The model described above is suitable for the general case of training a network based on noisy labels. In our intron dataset the feature vectors x are the evolutionary history patterns of the intron. The true unobserved binary intron label is either 1-functional or 0-nonfunctional. The noise here has a specific structure based on biological fact that for some introns their functionality has been determined and therefore we know that their label is 1. For all other introns their labels are currently unknown. Using the notation defined above, in our case we assume that $\theta_{00} = 1 - \theta_{01} = 1$. In other words, an intron whose true label is no-function is never reported as a functional intron and therefore our noisy channel is restricted since only 3 of 4 transitions are allowed. We are actually left with a single noise parameter $\theta = \theta_{11} = 1 - \theta_{10}$. The parameter θ_{11} is the probability that a functional intron has a known functionality, i.e., that is indeed reported as functional intron. The noisy channel in our case is illustrated in Figure 3. Therefore, the noise parameter update in the M-step is simplified to:

$$\theta = \theta_{11} = \frac{\sum_t c_{t1} 1_{\{z_t=1\}}}{\sum_t c_{t1}}. \tag{11}$$

Both EM and back-prorogation algorithms are iterative methods and we can alternate between them. Therefore there

Table 2. Feature description.

Feature	Description	Feature Type
LOGLIKE	The likelihood of observing the presence-absence pattern of the intron, based on the EREM model [15]	Continuous
In AMPHIBIANS	This feature is 1 if the intron is present in at least one amphibian	Binary
IN FISH	This feature is 1 if the intron is present in at least one fish	Binary
IN BIRDS	This feature is 1 if the intron is present in G. gallus	Binary
IN FUNGI	This feature is 1 if the intron is present in at least one fungus	Binary
IN PLANTS	This feature is 1 if the intron is present in at least one plant	Binary
IN PROTIST	This feature is 1 if the intron is present in at least one protist	Binary
SANKOFF G3L1	The minimum number of intron gain and loss events required to explain the intron's presence-absence pattern, based on weighted-parsimony where gains cost three times as much as losses	Discrete
SANKOFF G1L3	The minimum number of intron gain and loss events required to explain the intron presence-absence pattern, based on weighted-parsimony where losses cost three times as much as gains	Discrete
LCA AGE	The age [in million of years] of the last common ancestor (LCA) of all intron-bearing species	Discrete
MED REL POSITION	The median distance of the exon-exon junction from the beginning of the coding sequence (CDS) divided by the CDS length	Discrete
MED POSITION	The median distance of the exon-exon junction from the beginning of the CDS [nucleotides]	Discrete
ONES RATIO KNOWN	The number of 1s divided by the total number of 1s and 0s in the intron presence-absence pattern	Discrete

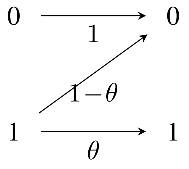

Fig. 3. A diagram of the restricted noise model in the intron functionality data in which not all transitions are allowed.

is no need to fully optimize the NN model at each M-step iteration. We can use standard methods for neural-network training and update the noise parameter θ after a few epochs over the training data. The EM algorithm is known to be a greedy optimization procedure and therefore prone to be sensitive to the starting point. Smart initialization of the model parameters is crucial in order to achieve good results. We used the following strategy to initialize our algorithm. First we trained our neural net assuming the introns with unknown functionality actually have a non-functional label, thus assuming we have clean labeled data. The obtained NN parameters set w is then used as an initial value for the first EM iteration. Then we computed the probability that each intron in the training data is functional based on the obtained model and use it as an initial value for the noise parameter set θ:

$$\theta = \theta_{11} = \frac{\sum_t 1_{\{z_t=1\}} p(y_t = 1|x_t; w)}{\sum_t p(y_t = 1|x_t; w)}.$$

The proposed method, which we dub the Restricted Noisy Labels Network (RNLN) algorithm, is summarized in Table 1.

Table 3. Data distribution

	Functional	Unknown
No. of introns	243	6180
Percent	3.7%	96.3%

3. EXPERIMENTS

3.1. The dataset

The dataset is consisted of 6423 introns, for which the labels of 243 are already known to be functional and the labels of the emaining 6180 introns are unknown since no functionality has been found for them, see Table 3. Our research objective was to estimate what fraction of the introns are functional and moreover to determine whether each intron is functional or not.

Each intron is represented by 13 features that are computed using the evolutionary history patterns of the intron [2]. These features are listed in Table 2:

3.2. Results

We first applied unsupervised data embedding into the 2D plane using linear embedding. The functional introns, labeled as blue dots, were well clustered, as shown in Figure 4. This indicates that the features are indeed relevant and informative for our intron functionality classification task.

Next we applied the proposed algorithm described in the previous section. We trained the network with 2 hidden layers with 8 and 5 neuron each and used the ReLU activation function. We applied a regular SGD learning scheme to train the network. Once we applied our algorithm we obtained the estimated (soft) labels of the introns: $p(y_t = i|z_t, w_t; \theta, w)$. By applying a threshold we converted the probabilities into a binary decision. Figure 5 shows the same 2D embedding where the introns that were classified as functional are shown as blue dots. As seen in Figures 4 and 5, the green dots in the original datasets whose 2D embedding was close to the cluster of functional introns were classified as functional, whereas the green cluster remained at 99%,suggesting that our algorithm is consistent with the unsupervised representations obtained by the PCA algorithm.

Validation-Test To validate our model we performed a 3 folds cross-validation. We divided our data into 3 folds, each one contained the same functional to non-functional ratio distribution as the complete data-set. We learned the model 3 times, each time with two-thirds of the data, and tested in the remaining third. Since in our data we only know the label of the functional-introns, we could only test the functional ones and obtained a 99$\%$ success rate.

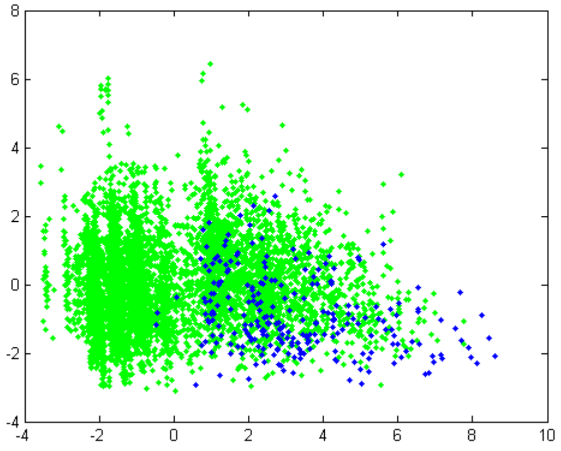

Fig. 4. 2-D PCA embedding of the (noisy) labeled data. The blue dots represent the introns known to be functional

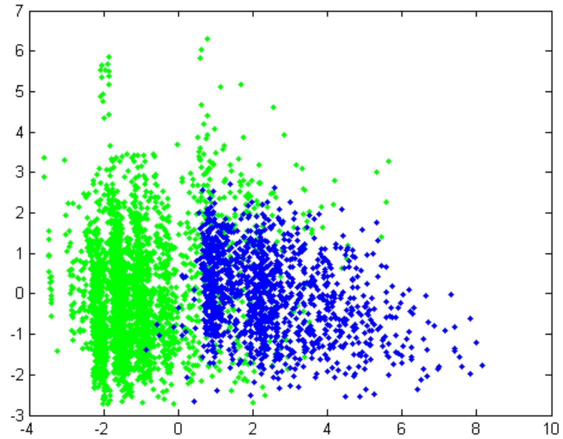

Fig. 5. 2-D PCA embedding of the classification results. The blue dots represent introns that were classified as functional by our algorithm.

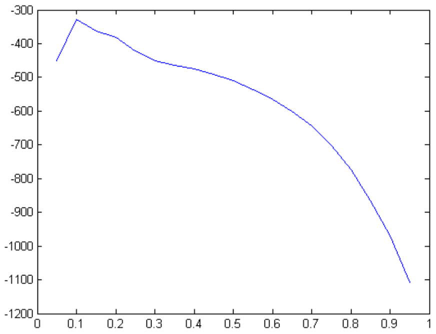

Fig. 6. The likelihood as a function of θ_{11}. It can be clearly seen that the maximum likelihood is achieved for $\theta_{11} = 0.11$, i.e. only 11% of the introns that are really functional are currently known to be functional in the dataset we used.

3.3. Parameter Analysis

We next analyse the parameters learned by the model and describe their biological interpretation. we first found the value of the single noise parameter θ_{11}. For each possible value of θ_{11} we computed the maximum likelihood value where the maximization was done over the network parameter-set w. Denote $l(\theta) = \max_w l(w, \theta)$ where $l(w, \theta)$ is the likelihood function defined in Eq. (3). In Figure 6 we plot $l(\theta_{11})$ as a function of the noise parameter θ_{11}. It can be seen the there is a clear single maximum point $\theta_{11} = 0.11$. This means that only 11% of the introns that ae really functional were currently known to be functional in the dataset we used.

Another interesting quantity is the probability that an unlabeled intron has a functionality. This can be computed using

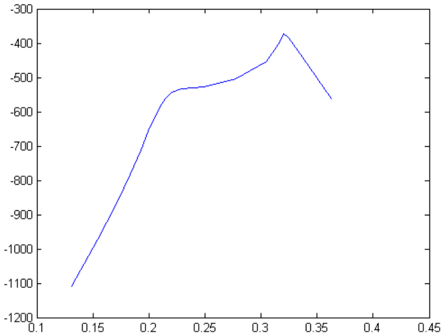

Fig. 7. The likelihood as a function of $p(y = 1|z = 0)$. It can be seen the maximum likelihood is achieved in $p(y = 1|z = 0) = 0.33$, i.e., 33% of the unlabeled introns actually have a functionality that have yet to be discovered.

Bayes rule:

$$P(y = i|z = j) = \frac{p(z = j|y = i)p(y = i)}{p(z = j)} = \frac{\theta_{ij}p(y = i)}{p(z = j)}$$

where $p(z = j)$ can be obtained directly from the data and $p(y = i)$ can be estimated as follows:

$$\hat{p}(y = i) = \frac{1}{n}\sum_t p(y_t = i|x_t, z_t).$$

Using the maximum likelihood parameters, we found that $p(y = 1|z = 0) = 0.33$, i.e., 33% of the unlabeled introns actually have a functionality that we didn't discovered yet. In Figure 7 we show the model likelihood function as a function of the probability $p(y = 1|z = 0)$. For each value of θ_{11} we trained the network and then computed both the likelihood and $p(y = 1|z = 0)$. This provided points in the 2D plane that are shown in Figure 7.

4. CONCLUSION

In this study we developed a method to identify which introns are functional based on 13 features that represent the evolutionary history of each intron. We proposed a Deep Learning Restricted Noisy labels model and applied it to solving this semi-supervised classification problem. Based on our model we predicted that a major part of all functional introns have yet to be discovered and we supported this hypothesis by successfully predicting 99% of the known functional introns in a validation set.

5. REFERENCES

[1] I. B. Rogozin, L. Carmel, M. Csuros, and E. V. Koonin, "Origin and evolution of spliceosomal introns," *Biology Direct*, vol. 7, pp. 11, 2012.

[2] M. Chorev and L. Carmel, "The function of introns," *Front Genet*, vol. 3, pp. 55, 2012.

[3] F. Hube and C. Francastel, "Mammalian introns: when the junk generates molecular diversity," *Int J Mol Sci*, vol. 16, no. 3, pp. 4429–4452, 2015.

[4] M. Chorev and L. Carmel, "Computational identification of functional introns: high positional conservation of introns that harbor rna genes," *Nucleic acids research*, vol. 41, no. 11, pp. 5604–5613, 2013.

[5] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[6] B. Jakramate and A. Kabán, "Label-noise robust logistic regression and its applications," in *Machine Learning and Knowledge Discovery in Databases*, pp. 143–158. 2012.

[7] A. J. Bekker and J. Goldberger, "Training deep neural-networks based on unreliable labels," in *Int Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[8] V. Minh and G. Hinton, "Learning to label aerial images from noisy data," in *Int. Conf. on Machine Learning (ICML)*, 2012.

[9] J. Larsen, L. Nonboe, M. Hintz-Madsen, and K. L. Hansen, "Design of robust neural network classifiers," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 1998.

[10] S. Sukhbaatar and R. Fergus, "Learning from noisy labels with deep neural networks," in *arXiv preprint arXiv:1406.2080*, 2014.

[11] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[12] N. Natarajan, I. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[13] J. Goldberger and E. Ben-Reuven, "Training deep neural networks using a noise adaptation layer," in *Int. Conference on Learning Representations (ICLR)*, 2017.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[15] L. Carmel, Y. Wolf, I. B. Rogozin, and E. V. Koonin, "Erem: Parameter estimation and ancestral reconstruction by expectation-maximization algorithm for a probabilistic model of genomic binary characters evolution," *Advances in Bioinformatics*, vol. 2010, pp. 167408, 2010.