Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype
Bojian Yin, Marleen Balvert, Rick van Der Spek, Bas E Dutilh, Sander Bohte, Jan Veldink, Alexander Schönhuth

To cite this version:
Bojian Yin, Marleen Balvert, Rick van Der Spek, Bas E Dutilh, Sander Bohte, et al.. Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype. Bioinformatics, 2019, 35 (14), pp.i538-i547. 10.1093/bioinformatics/btz369. hal-02344253

HAL Id: hal-02344253
https://inria.hal.science/hal-02344253
Submitted on 4 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype

Bojian Yin 1,†, Marleen Balvert 1,2,†, Rick A. A. van der Spek 3, Bas E. Dutilh 2, Sander Bohtó 1, Jan Veldink 3 and Alexander Schönhuth 1,2,*

1 Centrum Wiskunde & Informatica, Amsterdam, 1098 XG, The Netherlands
2 Theoretical Biology & Bioinformatics, Utrecht University, Utrecht, 3512 JE, The Netherlands
3 Department of Neurology, Brain Center Rudolf Magnus University Medical Center Utrecht, Utrecht, The Netherlands

† Shared first authorship. * To whom correspondence should be addressed.

Abstract

**Motivation:** Amyotrophic lateral sclerosis (ALS) is a neurodegenerative disease caused by aberrations in the genome. While several disease-causing variants have been identified, a major part of heritability remains unexplained. ALS is believed to have a complex genetic basis where non-additive combinations of variants constitute disease, which cannot be picked up using the linear models employed in classical genotype-phenotype association studies. Deep learning on the other hand is highly promising for identifying such complex relations. We therefore developed a deep-learning based approach for the classification of ALS patients versus healthy individuals from the Dutch cohort of the Project MinE dataset.

**Results:** Our approach identifies potentially ALS-associated promoter regions, and generally outperforms other classification methods. Test results support the hypothesis that non-additive combinations of variants contribute to ALS. Architectures and protocols developed are tailored towards processing population-scale, whole-genome data. We consider this a relevant first step towards deep learning assisted genotype-phenotype association in whole genome-sized data.

**Contact:** a.schoenhuth@cwi.nl

1 Introduction

Amyotrophic lateral sclerosis (ALS) is a neurodegenerative disease affecting the upper and lower motor neurons, resulting in a progressive loss of muscle strength leading to paralysis and eventually death (Goldstein and Abraham, 2013; Fuhkan et al., 2007). For many patients ALS is likely caused by genetic aberrations. While a handful of major genetic risk factors have been identified, no more than 15% of the heritability has been explained so far (Van Rheenen et al., 2016). This is because the genetic architecture of ALS has been found to be rather involved: ALS seems to be evoked through not necessarily additive combinations of genetic
aberrations that individually only have a small effect and can thus not be detected using the currently available genotype-phenotype association approaches (Van Rheenen et al., 2016).

Motivated by these findings, the application of prediction and association schemes that can capture non-additive effects is very promising. More than that, the evaluation of more complex schemes might even be an urgent necessity if one aims at further progress in predicting ALS, associate it with genetic causes, and, eventually, also treat it successfully.

In the last ten years, the identification of genotype-disease relations has been considerably enhanced by the use of large-scale genome data. Project MiAnts is an international initiative to collect genome data of tens of thousands of ALS patients and healthy control individuals. Many individuals have been sequenced at considerable depth of genome coverage (Project MiAnts ALS Sequencing Consortium and others, 2018). The current analysis demonstrates the feasibility of in-depth exploration. Clearly, it carries the potential for pinpointing ALS risk factors, guiding further research and drug development.

Genome-wide association studies (GWAS) are the current state-of-the-art in analyzing genotype-phenotype data. Statistical tests are used to determine which is the level of genetic markers between a specific phenotype and, and are therefore suitable for uncovering genotype-phenotype associations that involve single variants or variants interacting with others in additive schemes. GWAS have successfully identified disease-associated variants over a wide range of disorders (Wen et al., 2017) including ALS (van Bo et al., 2009; Nicolaus et al., 2018). However, the approach has been found to be unable to find the non-additive combinations that are associated with phenotypes (Wen et al., 2015), which limits its power as genetic variants often constitute phenotype in non-additive combinations. This could for example be caused by epistasis, where the effect of one variant on phenotype is dependent on the presence or absence of others (Frankel and Schork, 1997; Moore, 2003). As above-mentioned, the genetics underlying ALS have been found to be more involved and are therefore unlikely to be fully unravelled using basic association schemes (Van Rheenen et al., 2016). The application of novel data analysis approaches that account for complex interactions between genotype input variables and ALS are thus very promising.

Thanks to advances in the recent past, deep neural networks (DNNs) have turned into powerful classifiers in several application areas including bioinformatics (Agerermeier et al., 2016). They have been proven to map arbitrarily complex relationships between multiple input features (in our case genetic variants) and output labels (here for example binary-valued labels 'ALS' or 'no ALS'). In addition, DNNs have been pointed out to be particularly big data compatible (Schmidhuber, 2015). That is, they can handle a considerably larger number of input variables than most other machine learning methods, a prerequisite for the analysis of genome data. DNNs therefore hold the promise to successfully map complex genotype-phenotype associations.

DNNs cannot, however, be applied off-the-shelf when mapping variants to disease (ALS) status; several hurdles need to be overcome. The first is the size of genome data: the sheer number of input variables (genetic variants, which amount to usually millions) exceeds the number that these models can deal with easily (a few hundred of thousands). Second, while DNNs can achieve high classification accuracy, interpretability is insufficient: it is difficult to determine why a DNN classified a sample as a case or a control. This is a major drawback for genotype-phenotype association studies, as the main goal is to identify (combinations of) variants that associate with disease rather than obtaining a high classification accuracy. Third, DNNs have delivered their most striking successes when applied in image classification tasks. High classification accuracies were obtained with networks of great depth, employing the hierarchical nature of these images (pixels together form lines, which together form basic shapes, etc.). Thereby, the employment of convolutional filters has been crucial in delivering the breakthroughs. Such filters make use of the position invariance of local structures in images, a property that does not hold for genome data.

A few studies have considered using deep learning for genotype-phenotype association studies. Most approaches first reduced the number of variants included in the model either by selecting variants that were known to be associated with disease (Oppen and Krishna, 2017; Fries et al., 2017), or by preselecting those variants that showed a sufficiently strong correlation with phenotype in a regular GWAS (Montesano et al., 2018; Blügg et al., 2018). Two studies combine the latter strategy with the use of autoregressors for further dimensionality reduction (Montesano et al., 2018; Fergus et al., 2018). These approaches have the same drawback as a classical GWAS: already in the preselection step epistasis is overlooked, and variants that have a small effect on their own will not be included in further analysis. An alternative approach is proposed by Ramesh et al. (2016). The authors limit the computational hurdles by considering the transpose of the data matrix, which is similar to considering features as samples and vice versa, to learn the model parameters. As the number of genetic features is much larger than the number of samples in a genotype-phenotype association study with large genome data, the reduction in the size of trainable parameters and hence strongly reduces the time required for training. Trau and Blei (2017) define an implicit causal model that aims to identify relations between variants, and deals with the data dimensionality by updating the model once variant at the time. In summary, while a couple of earlier studies have used deep learning to predict phenotype from genotype, only two were able to deal with several hundreds of thousands of genetic variants. None have employed the structure inherent to genome data, and interpretation of the results has not been addressed in these studies.

This paper presents novel deep neural network architectures and a protocol by which to predict the occurrence of ALS from individual genotype data. In summary, we developed a deep learning-based method that (1) allows for the use of genome-sized data by pre-selecting parts of the genome that are most relevant for classification, (2) provides insight in which genetic regions are relevant to classification, and (3) is capable of classifying ALS patients versus healthy control individuals from genome data. The design of our approach in general and our network architecture in particular is driven by the structure of genome data.

We demonstrate in our experiments that by means of our new architectures, we achieve 77% accuracy in predicting ALS from genome data when considering chromosomes 7, 9, 17 and 22. Our results demonstrate that our ALSNet clearly outperforms other machine learning tools and protocols that we have been experimenting with, and thus demonstrates GWAS style prediction technology based on logistic regression. Our results therefore demonstrate that prior knowledge on the structure of genome data can aid in the design of a deep learning-based approach and the neural network architectures to yield improved accuracy rates in classifying genotypes with respect to occurrence of ALS. At the same time, we are aware that here we have only made the first steps towards runtime application of deep neural networks in classifying genetically involved diseases from individual genomic profiles. We will, point out where further improvements are conceivable along the way in the following, convinced that we are, at the very least, providing a very promising template for further explorations along this avenue of research.

2 Approach

We propose to make use of prior knowledge to tackle the dimensionality issue inherent to working with genome-sized data. The majority of millions of variants in genome data are irrelevant, as these are not involved
in disease. It has been found in general that most variants that relate to disease phenotypes reside in the DNAse hypersensitive sites (Manolio et al., 2012), that is, in the majority of cases they occupy the promoter regions preceding genes, where transcription is initiated. We therefore focus on the promoter regions.

An interpretable model is able to indicate which genomic regions were relevant to classification. We therefore developed a two-step approach to employ neural network architectures for mapping associations between genotypes and the occurrence of ALS. The first step consists of individual classifiers for each promoter region, i.e., individuals are classified based on their genomic information from a single promoter region only. The classification accuracy obtained with an individual promoter region is an indication for the region's predictive power, and only the eight best performing promoter regions are considered for further analysis. In the second step, the genome information of the selected promoter regions is combined and an overall classifier is trained for final classification. This is illustrated in Figure 1, where we denote the promoter region-specific neural network by Promoter-CNN (CNN for convolution neural net) and the network that classifies samples based on a combination of promoter regions by ALS-Net. We develop and validate our approach using GWAS data from the Dutch cohort of Project MiNE, which contains 4,511 cases and 7,979 controls.

As noted before, the success of DNNs for image classification heavily relies on the local structures that are present in images. Genotype data do not convey neighborhood structures that are as easy to grasp as in images and applying convolution is less straightforward. Still, genotype data does have a neighborhood structure which is due to two aspects. First, the genome consists of blocks that form functional units, such as genes and promoter regions. Second, genetic variants are passed on from ancestor to offspring in terms of blocks rather than in isolation. Although in many cases details have not been fully understood, usually combinations of neighboring variants (haplotype blocks) are responsible for the establishment of phenotypes, rather than variants in isolation. This justifies the application of DNNs that take neighborhood structures into account (Bellet et al., 2018). Note that the above does not contradict that isolated variants can be indicative of phenotypes: single variants usually are in linkage disequilibrium with other variants in their block, which establishes that basic GWAS can nevertheless be successful.

3 Methods

3.1 Project MiNE data

We use data collected by Project MiNE, a worldwide effort to collect whole-genome data from both ALS patients and unaffected individuals for the identification of ALS-causing variants (Project MiNE ALS Sequencing Consortium and others, 2018). The dataset we used contains solely the Dutch cohort, consisting of 4,511 ALS patients and 7,979 healthy individuals, including 6,127 males and 5,781 females. First SNPs were annotated according to dbSNP137 and mapped to the hg19 reference genome. Quality control (QC) was performed per cohort to remove low quality SNPs and individuals using PLINK 1.9 (Purcell and Chang, 2015; Chang et al., 2015) (--geno 0.1 and --mind 0.1). HapMap3 (Consortium et al., 2010) projected principal components were calculated and extreme CEU population outliers were removed (25 standard deviations, SD). Cohorts were merged into strata based on genotyping platform. Subsequently, more stringent SNP QC was performed (--maf 0.01, --mind 0.02, --hwe 1e-5 mail include-every, --test-miss --p<1e-8) followed by more stringent individual QC (--geno 0.02, --miss <0.2, and removed snpcheck failures and missing phenotypes). We then only kept the autosomal regions.

Motivated by the fact that chromosomes 7, 9 and 17 all have been found to carry elevated amounts of missing heritability (Van Rheenen et al., 2016), we focus on those chromosomes. Additionally, we included chromosome 22 that was reported to have a low level of heritability. The genome data of the four chromosomes contains 833,104 positions of variation.

Note that all chromosomes occur in pairs: one maternal and one paternal copy. We convert the data, which is in VCF format, to isolate allele frequency data. Hence the data of each individual is a list of values in {0, 1, 2}, indicating the number of occurrences of the minor allele at each position on the genome. In some cases information for one of the chromosome copies was missing. In such cases, we assume this to be the frequent allele here. This step can be improved in future work by eliminating missing values through high quality imputation.

We focus on the promoter regions. As the position of a promoter region on the genome is generally not as well defined as the transcription start sites of a gene, and because a deep neural network requires the data representation for each promoter region to be of the same size, we used the following approach for determining the variants that are in the promoter regions. We used the transcription start sites as reported in the RefSeq database (O’Lear et al., 2015). The 56 variant positions upstream and the 8 variant positions downstream of the transcription start site were then included in our representation of the promoter region. Hence, each promoter region is represented by a list of 64 values from the set {0, 1, 2}.

Note that a gene can have multiple transcription start sites, and hence multiple promoter regions.

![Fig. 1. An overview of the workflow. CV: cross-validation, acc: accuracy.](image-url)
In summary, the input data to our model for one individual is a list of vectors \((0, 1, 2)^{256}\), where each vector resembles the occurrences of the minor allele at the positions in a promoter region.

### 3.2 Neural network architectures

Promote-CNN uses two convolution layers followed by two dense layers. As such (unlike ALS-Net in the following), Promote-CNN is not deep, which is justified by the small input. Details of the architecture of Promote-CNN are presented in Table 1. Batch normalization is applied after each layer, followed by the softplus activation function.

| Layer type | Description | Output shape |
|------------|-------------|--------------|
| Input      |             | (64, 1)      |
| Convolution, BN and Act | 1 x 1 filter, 4 output channels | (64, 4) |
| Convolution, BN and Act | 4 x 4 filter, 32 output channels | (91, 32) |
| Reshape    | Flatten     | (1056, 1)    |
| Dense, BN and Act |             | (148, 1)    |
| Dense, BN and Act |             | (16, 1)     |
| Output     | Softmax     | (2, 1)       |

ALS-Net is a more involved neural network, where the design of the architecture is based on intuition guided by the structure of genome data. This will be further explained below. The full network architecture is shown in Figure 3 with further details in Section 4 of the supplementary materials. Note that the network contains seven blocks of layers. These are recurring stacks of convolution and pooling layers, of which details are provided in Figures 7 up to 10 in Section A of the supplementary materials.

The input is formed by concatenating the vectors with genome information from the individual selected promoter regions to obtain one vector of length \(64 \times 8 \times c = 512c\), where \(64\) is the number of variants in a promoter region, \(8\) is the number of selected promoter regions per chromosome, and \(c\) is the number of chromosomes included in the analysis. When dealing with all autosomes of a genome \(c\) reaches a maximum of 22, which results in a vector of length \(512 \times 22 = 11,204\). By order of magnitude this scales just right with the number of training data available (Project MicaE: several tens of thousand individuals), providing evidence of the potential to deal with whole genome-sized data.

In the first block of layers each promoter region is considered separately, that is, the information from different promoter regions is not yet combined. This allows the model to focus on obtaining a good representation of the individual promoter regions before combining their information. The first layer of Block 1 is a convolution layer with stride 64, which ensures that information from separate promoter regions is not combined, and kernel size 64, which implies that the information from each promoter region is processed as a whole (no convolution within the promoter region). The layer has 256 output channels, hence 256 functions of the input values of a single promoter are trained and the information from a promoter region is new represented by 256 values (see convolution step in Figure 2). The second layer is a convolution layer with kernel length 1, stride 1 and 256 output channels, as proposed by Howard et al. (2017).

This layer takes a linear recombination of the information from each single promoter region. It does so 256 times with different weights and biases, once for each output channel. The first two convolution layers do not have an activation function. The block is concluded with a batch normalization layer and a rectified linear unit activation function.

Next, the tensor is reshaped into a 3 dimensional tensor, where the information of each promoter region is reshaped from a vector of length 256 to a 16 by 16 matrix (see reshape step in Figure 2). This three dimensional tensor can be viewed as an image of 16 by 16 pixels with 8c channels, where each channel corresponds to a promoter region.

In block 2 the promoter regions are combined, hence from this point onwards the information from the promoter regions is considered together. The main building blocks of the network are convolution layers, which allow for learning from large input data without using an excessive number of trainable parameters. We often employ three consecutive (separable) convolution layers, which we represent by block 2 (convolution layers, Figure 8 in the supplement) and block 4 (separable convolution layers, Figure 10 in the supplement). In block 3 (Figure 9 in the supplement) convolution layers are alternated by pooling layers to prevent the model from over-fitting.

Since the underlying classification task requires the model to identify complex patterns we employ parallel computation blocks (after block 3) as well as residual connections followed by an "add" operation (inputs to the "+" operator, dashed arrows in Figure 6) to prevent the loss of information in future layers (Szegedy et al., 2015; He et al., 2016). The model concludes with two dense layers to combine all information into a single classification. Within dense layers are usually preceded by a flattening of the output of the previous layer, see make use of a global average pooling layer instead to allow for a stronger dimensionality reduction.

The architecture of ALS-Net is optimized using cross validation, see the next section for further details.

### 3.3 Training and testing procedure

The dataset was split into a train-validate set (90% of samples) and a test set (10% of samples). The train-validate set was used for model development and selection of the promoter regions, the test dataset was used only for final testing. To test the model fairly, the ratio of cases and controls is 1:1 in the test dataset.

A nine-fold cross-validation on the train-validate data was used to train Promote-CNN. For each chromosome the eight promoter regions that achieved the best prediction accuracy averaged over the nine folds were selected for further analysis. This small network is trained using stochastic gradient descent on 50 epochs where the batch size is 64, and with a learning rate of 0.01.

The architecture and other hyperparameters for ALS-Net are optimized using a nine-fold cross-validation of the train-validate data. The network architecture was optimized based on the learning curve and performance measures such as accuracy, precision and recall. For examples of
the performance of networks that slightly deviate from others as well as a simple multi-layer perceptron, see Table 5 in Section B of the supplementary materials. Additionally, we present the performance of a much more shallow neural network, namely a three-layer MLP, in the last row of Table 5. These results show the necessity of using a deep network to achieve high recall. Network parameters are optimized using the AdaGrad algorithm (Duchi et al., 2011) with an initial learning rate of 0.02 and a decay of $2e^{-4}$. Optimization was performed over 300 epochs with a batch size of 32.

The model's network architecture is optimized based on chromosome 7 only, and used for all four chromosomes individually as well as for the combination of the four chromosomes. Parameters are optimized separately for each chromosome as well as for the combined model.

The performance of our approach was tested by applying ALS-Net to the test data. Hence for these samples we only use the selected promoter regions.

3.4 Comparison with other machine learning approaches

The performance of ALS-Net is assessed using the test data, and is compared with the performance of logistic regression – this corresponds with the approach for calculating a basic polygenic risk score (PRS) (Duthridge, 2013) –, support vector machines (SVM, Vapnik (1998), Joachims (1999)), random forest (Breiman, 2001) and AdaBoost (Freund et al., 2000; Freund et al., 1999). For each of these we used the same promoter regions as for the large neural network. Hyperparameters were optimized using a cross-validation approach, and performance on the test dataset is reported.

In logistic regression a linear function is used to estimate the disease risk score from genotype, followed by a classification where samples with a predicted risk score above a predetermined threshold are considered as positives (in our case "ALS") and the others as negatives (in our case "no ALS"). While GWAS uses a single genetic variant as explanatory variable, we base our prediction of disease status on multiple variants, as is common for the calculation of the PRS. We apply logistic regression to the full set of promoter regions as well as to the variants that reside in the promoter regions selected by Promoter-CNN. Note that the choice of threshold determines the balance between precision and recall. In order to allow for comparison with the other methods, we chose the threshold such that accuracy on the training set is maximized.

SVM is a popular binary classification method designed to find a non-linear boundary (determined by the kernel function) to maximize the margin between two clusters. Here we used a radial basis function SVM with kernel coefficient 0.001.

Random forest is a widely used machine learning algorithm that creates multiple decision trees and combines their individual classifications to obtain a final classification. Using a large number of decision trees is required for higher accuracy, but also results in slow training. We implement a random forest consisting of 100 trees with maximum depth 5, and at most 100 features will be considered when looking for the best split.

The core idea of AdaBoost is to train several decision trees, assign weights to samples and classifiers to force the algorithm to focus on hard-to-classify samples, and combine the weighted classifications to form a stronger final classifier. While the model is powerful and yields explainable results, it is sensitive to outliers. We used AdaBoost with 1000 decision trees of depth 3.
4 Results

4.1 Single promoter classifiers select known ALS-associated genes as well as potential novel risk factors

Figure 4 shows histograms for the classification accuracy for the single promoter classifiers, organized per chromosome. While most promoter regions lead to an accuracy around 0.5 - the same as random - the distribution has a tail on the right with a few promoter regions achieving higher accuracy. Hence only a few promoter regions have the potential to aid in classification of cases versus controls.

The genes that the selected promoter regions correspond to are listed in Table 2 together with the accuracy, precision, recall and F1-score obtained with Promoter-CNN. Some of these genes have been associated with ALS or other neurological disorders before, while others can be viewed as potential ALS-related genes. The ALS-Net outperforms all other methods in terms of precision - regions obtained by running a logistic regression are presented as well (Acc LR). The results show that using logistic regression would have resulted in a partially different selection of promoter regions. Recall that multiple promoter regions can correspond to a single gene, as a gene can have multiple transcription start sites. See also Table 7, section D in the supplementary materials for some annotations (known gene ontology classes) for genes selected in chromosome 7. While the polymeric loci in the selected promoters are important as input for successful deep learning based classification, we do not yet provide clear evidence whether, and if so how, the selected genes are associated with ALS, which is important to keep in mind. Please also see (Biedrzycki et al., 2019) for a (wholesome) discussion.

Several genes that were associated with ALS by earlier studies are not among the top eight performing promoter regions from Promoter-CNN. The classification accuracies from Promoter-CNN for the ALS-associated genes reported by Abet al. (2013) are listed in Table 6 in Section C of the supplementary materials.

4.2 ALS-Net outperforms other classifiers in terms of accuracy and recall

The selected promoters regions were included in a final overall classifier. We compared the performance of ALS-Net with logistic regression, SVM, random forest and AdaBoost (indicated by Promoter-CNN + classifier, Table 3). Additionally, we compare the results of Promoter-CNN with the five classifiers in logistic regression on all promoter regions, so without the help of Promoter-CNN. This was only possible for the individual chromosomes, as a logistic regression on all promoter regions from the four chromosomes combined required too much RAM. SVM, random forest and AdaBoost could not deal with the full chromosome data of even a single chromosome. The methods are compared based on classification accuracy, precision, recall and the F1 statistic for each chromosome separately as well as for their combination, and results are presented in Table 3.

First note that the classification accuracy of logistic regression is improved by Promoter-CNN. Second, Promoter-CNN + ALS-Net outperforms all other methods in terms of accuracy, closely followed by Promoter-CNN + logistic regression. Both methods largely outperform SVM, random forest and AdaBoost. Third, Promoter-CNN + ALS-Net almost always yields the highest recall, but is almost never outperformed by logistic regression or random forest in terms of precision - i.e., ALS-Net is better at identifying ALS patients (lower number of false negatives) but classifies healthy controls more often as patients than the other methods (higher number of false positives). Thus, each of the methods provides a different trade-off of precision versus recall. We therefore also consider the F1 statistic, a combined measure of precision and recall. Our deep neural network outperforms the other methods in terms of the F1 statistic for three out of the four individual chromosomes as well as the combination of chromosomes.

4.3 Including more genomic information improves classification

For most models the highest accuracy is obtained when the four chromosomes are combined rather than considering each chromosome individually. This does not hold for Promoter-CNN + Random Forest, which is likely due to the fact that a larger forest would be required to be able to deal with the larger dataset that is obtained when combining the four chromosomes. Since this slows down training times considerably, the applicability of random forests on genome-sized data remains (more than) questionable.

4.4 Potential identification of disease-associated variants with ALS-Net

In order to identify which input features (in our case, genetic variants) were relevant for a neural network's classification of a single sample one can make use of saliency maps. These are heatmaps that show the gradient of the objective function with respect to each input feature. A large absolute value indicates a strong influence of this feature on the final classification. We have constructed saliency maps for 100 randomly sampled ALS patients and 100 randomly sampled healthy controls. The
Table 3: Classification results obtained with four classification methods applied to chromosomes 7, 9, 17 and 22 independently and combined. The result of best performing model for the given (set of) chromosome(s) is denoted in italic, while the overall best score is indicated in bold. Chr = chromosome.

| Classifier | Chr | Accuracy | Precision | Recall | F1-Score |
|-----------|-----|----------|-----------|--------|----------|
| Logistic  | 9   | 0.625    | 0.642     | 0.586  | 0.602    |
| Regression| 17  | 0.546    | 0.574     | 0.535  | 0.535    |
|           | 22  | 0.590    | 0.619     | 0.567  | 0.588    |
| Promoter-CNN | 7 | 0.675 | 0.667 | 0.695 | 0.661 |
| + ALS-Net | 17 | 0.688 | 0.725 | 0.696 | 0.663 |
|           | 22 | 0.617 | 0.601 | 0.410 | 0.517 |
| All       | 7   | 0.769    | 0.711     | 0.908  | 0.797    |
|           | 9   | 0.635    | 0.728     | 0.485  | 0.555    |
| Promoter-CNN | 17 | 0.683    | 0.734     | 0.560  | 0.685    |
| + Logistic | 22  | 0.580    | 0.714     | 0.299  | 0.422    |
| Regression| 17  | 0.739    | 0.759     | 0.699  | 0.728    |
|           | 22  | 0.550    | 0.750     | 0.151  | 0.252    |
| Promoter-CNN | 17 | 0.558    | 0.729     | 0.263  | 0.399    |
| + SVM     | 17   | 0.577    | 0.788     | 0.212  | 0.334    |
|           | 22   | 0.521    | 0.743     | 0.267  | 0.393    |
| All       | 7    | 0.725    | 0.783     | 0.624  | 0.694    |
|           | 9    | 0.579    | 0.759     | 0.239  | 0.351    |
| Promoter-CNN | 17 | 0.645    | 0.762     | 0.420  | 0.542    |
| + Random  | 22   | 0.587    | 0.743     | 0.265  | 0.391    |
| Forest    | 22   | 0.596    | 0.813     | 0.249  | 0.381    |
|           | 7    | 0.562    | 0.776     | 0.175  | 0.285    |
|           | 9    | 0.604    | 0.642     | 0.462  | 0.541    |
| Promoter-CNN | 17 | 0.621    | 0.668     | 0.481  | 0.559    |
| + AdaBoost| 17   | 0.599    | 0.633     | 0.472  | 0.401    |
|           | 22   | 0.561    | 0.591     | 0.398  | 0.475    |
| All       | 7    | 0.661    | 0.780     | 0.565  | 0.625    |

Table 4: Training accuracy, precision and recall for each method, obtained with Promoter-CNN + Logistic regression and Promoter-CNN + ALS-Net. In C5 there are no cases, implying that TP and FN counts are zero, which renders precision (= 0) and recall (= undefined) statistics meaningless in the frame of a comparison.

| Classifier | Batch | Accuracy | Precision | Recall |
|-----------|-------|----------|-----------|--------|
| Promoter-CNN + | 71    | 0.648    | 0.510     | 0.793  |
| ALS-Net   | 71    | 0.711    | 0.829     | 0.573  |
| + Logistic | 54    | 0.934    | 0.000     | N/A    |
| + C44     | 71    | 0.780    | 0.753     | 0.967  |
| + C3      | 60    | 0.626    | 0.480     | 0.373  |
| + C5      | 60    | 0.434    | 0.765     | 0.313  |
| Logistic  | 54    | 0.990    | 0.000     | N/A    |
| + C44     | 54    | 0.657    | 0.740     | 0.768  |

Together C44 and C5, two highly imbalanced batches, cover approximately 95% of the individuals. A classifier may thus achieve good accuracy on predicting disease status by picking up batch-related data structures rather than disease-associated genetic characteristics. If a classifier picks up differences between C5 and C6 instead of between case (ALS) and control (no ALS), the classifier will fail to make reasonable predictions in C1 and C3, which still cover 786 individuals. Since both C1 and C3 are fairly balanced in terms of case-control labels, batch labels cannot be confounded with true case / control labels as easily as in C5 and C44. To check whether Promoter-CNN + ALS-Net (our approach) and Promoter-CNN + Logistic Regression (as the second best classifier evaluated) pick up on disease status rather than batch effects during training we evaluated the performance of these two within the individual batches on the training data. As can be seen in Table 4 both classifiers achieve good accuracy within batches C5 and C44, for which it remains unclear whether the classifiers predict batch labels rather than true labels. On C1 and C3 Promoter-CNN + Logistic Regression fails to bring up competitive performance rates (in particular: precision/recall logistic regression: 0.48 / 0.37 on C1 and 0.77 / 0.31 on C3), while Promoter-CNN + ALS-Net keeps significantly better performance rates (precision/recall: 0.51 / 0.79 on C1, 0.83 / 0.76 on C3), a clear indication that ALS-Net picks up truly ALS related effects to a substantial amount. For logistic regression however, it is likely that batch effects have been picked up, which lead to random classification in batches C1 and C3.

While ALS-Net comes with the clear promise to be (considerably) less prone to picking up batch effects, we conclude to say that correcting for confounding effects for CNN based methods still requires (most interesting!) further research.

4.6 Runtimes of ALS-Net are acceptable

Promoter-CNN was run on a CPU cluster. The training process for a single promoter region takes around 200s. Note that training of the promoter regions can be done in parallel on a multi-processor system: ALS-Net was trained on a GPU (Nvidia TitanX). For the largest model, which classified individuals based on information from the four chromosomes combined, the model needed 50h and 10GB of RAM for training.

5 Discussion

In this work we presented a novel deep learning-based approach for genotype-phenotype association studies on genome-sized data that has the potential to identify phenotype-associated genomic regions. By making use of earlier evidence that regulatory elements harbor the majority of disease-associated variants we developed a two-step approach, and designed a neural network where we made use of the structure of genome
data by first considering each promoter region separately, and then combining their information in later layers. The combination of Promoter-CNN with ALS-Net has several advantages: (1) it can easily be extended to handle genome-sized data, (2) identifies regions of the genome that are relevant to classification of ALS patients versus healthy controls, and (3) yields good classification results.

ALS-Net outperforms other methods in terms of classification accuracy, followed by Promoter-CNN aided logistic regression. As for prior related work, note that Bell et al. (2018) observed small improvements of CNN based methods over logistic regression in several, but not all cases. Here we observe some marked improvements of Promoter-CNN + ALS-Net over logistic regressions. An explanation might be that Bell et al. (2018) use a (substantially) simpler (less deep) network architecture and make a pre-selection of genetic features based on linear models, and hence overlook non-additive interactions already in the pre-selection step.

ALS-Net outperforms all other methods in terms of recall, also called power. This indicates that our approach might point out ways to overcome the (notoriously complained) lack of power that arises from the use of linear models when associating genotypes with phenotypes that underlie more involved genetic mechanisms. Additionally, further examination of what caused the increase in true case predictions might yield novel insight in the genomic mechanisms underlying ALS. Overall, ALS-Net provides a better trade-off between precision and recall as measured by the F1 statistic, which finally documents its value as a predictor in general.

Note that all methods have been helped by our two-step approach: the classification performance of logistic regression goes up when combined with Promoter-CNN, while none of the other methods evaluated were able to process chromosome-sized genome data without pre-selecting features.

Our results support the belief that ALS is caused by non-linear combinations of variants, which was hypothesized before by Van Rheenen et al. (2016). Table 3 shows a low recall for each of the individual promoter regions. Combining these (so a single classifier improves recall for Promoter-CNN + ALS-Net up to a level that far exceeds the recall obtained by Promoter-CNN logistic regression. This implies that the promoter regions on the different chromosomes interact in a non-additive way.

The two-step approach allows for the identification of potential ALS-associated genetic regions: Promoter-CNN selects promoter regions that are potentially associated with ALS. This information is then used by ALS-Net for classification. Our analysis has identified several promoter regions that potentially contribute to ALS prevalence, some of which are known to be associated with ALS. On the other hand, several ALS-associated genes (Abel et al., 2013) were not selected by Promoter-CNN. This does not necessarily imply that Promoter-CNN gave low prediction accuracies for these promoter regions: they simply were not among the eight most predictive promoter regions. Four out of the nine ALS-associated promoter regions that were not selected by Promoter-CNN were among the 5% best performing promoter regions (for these promoter regions, Promoter-CNN achieved an accuracy above 0.518, 0.513 and 0.520 for chromosomes 7, 9 and 17 respectively, see Figure 4). Despite missing some of the known ALS-associated genes, our final classification, which did not use any information from these genes, was able to classify at high accuracy. Further research is required to understand this.

The architecture of ALS-Net was optimized for chromosome 7. When applying this architecture to classify samples from the test set based on genotype data from chromosomes 9, 17 and 22 as well as their combination, the model performed very well and there was no need for further adjustments of the network architecture. These results show that our network architecture is flexible to unseen data, even to data from a different chromosome or set of chromosomes. Since beyond generally applicable genetics principles we have not made use of particular ALS-related knowledge, we believe that our architectures hold the potential to be applicable more universally. Further such experiments, however, predominantly depend on the availability of cohorts of sizes equal to the rather large cohorts we have been investigating here—which will be possible for ever more diseases in the mid-term future.

A major issue for genotype-phenotype association studies has been the large number of input variables, which causes issues for most machine learning approaches. To the best of our knowledge, there has so far been no method that can deal with more than half a million of genetic variants other than GWAS (where one tests for the association of a single variant with genotype) or approaches where GWAS or prior knowledge was used for pre-selecting relevant variants. This makes our approach the first that accounts for non-additive interactions between genomic features right from the start.

We view our work as a first step towards biologically-informed deep learning for association studies. We would like to emphasize that the current work is not a ready-to-use method that identifies relations between SNPs and phenotype, as one does in a GWAS. In fact, we do not envision that deep learning will lead to the identification of associations between individual SNPs and ALS. Rather, this method provides a framework for exploring which genomic characteristics are associated with disease. For example, this method allows us to better understand what makes a disease susceptible or resistant to a specific treatment.

While the results are promising, several improvements can still be made. By analyzing promoter regions individually with Promoter-CNN, the approach is capable of detecting non-linear interactions within a promoter region. While non-linear interactions across promoter regions cannot be detected at this point, ALS-Net will pick up interactions across the Promoter-CNN selected promoters. Note that ALS-Net cannot take all promoters as input, because the input would be too large. One can thus consider Promoter-CNN as a CNN-based feature selection. Interesting future work therefore is to increase the number of promoters that ALS-Net can cover. Also the number of included promoter regions may be chosen to be dependent on the length or the expected contribution to heritability of the chromosome under consideration. Additionally, an even deeper model may improve performance as well. We plan to further develop these methods in the future.

While this work presents a methodology for the analysis of genotype-phenotype data, refinements are required before practical implementation. For example, in our analysis we did not account for population stratification. As we first focus on the development of the neural network-based approach, we leave such improvements for future research.

The framework of our approach allows for analyzing full genome data. The pre-selection step of promoter regions is very fast and highly parallelizable, as Promoter-CNN is run on the promoter regions separately. Only the selected promoter regions are used as an input to ALS-Net. This input contains 237,649 to 1,408 variables (where 22 reflects the number of alleles, and k is the number of promoter regions selected per chromosome). This means 11,264 variables when k=8, an input size that is well manageable for a deep neural network. We may need to re-optimize the network architecture to achieve an optimal level of accuracy.

Even though our analysis was limited to the genomic information of only four chromosomes, we obtained a high level of classification accuracy. We plan to extend our work by including all chromosomes, which we expect to result in a strong increase in classification accuracy, as well as the identification of more potential ALS-associated promoter regions. Additionally, Project: Mining is a worldwide ongoing effort, and we plan to
apply our methods to the full dataset to strengthen our results once this data becomes fully available.

6 Conclusion

In this paper we presented ALS-Net, a convolutional neural network approach to predict ALS prevalence from genotype data. In order to employ the strengths of convolution we have developed a two-level approach where we focus on promoter regions, which are known sensitive sites for disease-causing variants. The architecture of the final classification network employs the strength of convolution and the structure of genome data by applying convolution filters to individual promoter regions. The results of our methods are promising, and are expected to generalize to genome regions that were unexplored in this work. Additionally, this work shows that deep learning is a highly promising approach for the identification of complex genotype-disease relations. We view our approach as a first step towards deep learning for genotype-phenotype association analysis guided by regulatory principles.

Funding

AS. BY and MB are supported by the Netherlands Organization for Scientific Research (NWO) Vidi grant 639.072.309, BED and MB are supported by NWO Vidi grant 864.18.004.

References

Abd, O. et al (2017). Development of a smartphone app for a genetics website: the myogenic-lateral-sclerosis online genetics database (myos). J Med Informatics and Allied Sciences, 1(2).
Andersen, M. et al (2016). Deep learning for computational biology. Molecular Systems Biology, 12, 878.
Bellido, F., de los Camiones, G. and Perea-Jimenez, M. (2018). Can deep learning improve genomic prediction of complex human traits? Genetics, 208(3), 809-819.
Boroditskiy, R. et al (2019). Spiking convolutional neural net for both one and two association signals. Genetic epidemiology.
Breiman, L. (2001). Random forest. Machine learning, 45(1), 5-32.
Chang, C-C. et al (2015). Second-generation phk: rising to the challenge of larger and richer datasets. Gigascience, 4(1), 1.
Cassendorf, M. et al (2016). Integrating coassociation and gene variance in diverse human populations. Nature, 407(6811), 52.
Dai, S. et al (2016). Next-generation genome imputation service and methods. Nature genetics, 48(6), 1264.
Duchi, J., Hazan, E. and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(6), 2121-2159.
Dudbridge, F. (2013). Power and predictive accuracy of polygenic risk scores. PLoS genetics, 9(5), e1003548.
Fergus, P. et al (2016). Utilizing deep learning and genome-wide association studies for epistasis-driven protein isoform classification in african-american women. arXiv preprint arXiv:1603.02977.
Freitags, W. and Schork, N. (1998). Who’s afraid of epistasis? Nature genetics, 14(6), 371.
Funada, Y., Schork, N. and Abe, N. (1990). A brief introduction to boosting. Journal of Japan Society For Artificial Intelligence, 14(7-8), 1612.
Funke, J. et al (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 28(2), 357-408.
Gao, R., Wang, Z. and Ji, S. (2014). Channelnet: Compact and efficient convolutional neural networks via channel-wise convolutions. In Advances in Neural Information Processing Systems, pages 5205-5213.
Goldstein, E. and Abrahams, S. (2013). Changes in cognition and behaviour in amyotrophic lateral sclerosis: nature of impairment and implications for assessment. The Lancet Neurology, 12(4), 368-381.
He, K. et al (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770-778.
Hero, M. et al (2017). Statistical learning of deep belief networks for x-ray data. Bioinformatics, 33(20), 3175-3183.
Howard, A. et al (2017). Multichore: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1706.04965.
Jawhari, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning, pages 137-142. Spilagen.
Mastropietro, M. et al (2012). Systematic visualization of common disease-associated variation in regulatory data. Science, 323(5919). 273-279.
Matsui, H. et al (2014). Analysis of causally obese individuals using deep learning stacked autoencoders and genome-wide genetic data. arXiv preprint arXiv:1504.05262.
Monteluz, C. et al (2018). Deep learning classification of polygenic obesity using genome-wide association study data. arXiv preprint arXiv:1804.01958.
Muren, J. (2003). The allelic nature of epistasis in determining susceptibility to common human diseases. Human Heredity, 56(1-3), 73-82.
Nomura, A. et al (2016). Genome-wide analyses identify KLF4 as a novel ALS gene. Neurology, 86(7), 1266-1268.
O’Leary, N. et al (2015). Reference sequence (refseq) database at ncbi current status, taxonomic expansion, and functional annotations. Nucleic acids research, 44(D1), D737-D745.
Pfeifer, J., Pender, N.P. and Harding, O. (2007). Cognitive impairment in amyotrophic lateral sclerosis. The Lancet Neurology, 6(11), 994-1003.
Ponpipit, M. et al (2018). Project scite: study design and pilot analysis of a large-scale whole-genome sequencing study in amyotrophic lateral sclerosis. European Journal of Human Genetics, 26(10), 1537.
Purcell, S. and Chang, C. (2015). PLINK 1.9. www.cog-genomics.org/plink1.9.
Romero, A. et al (2014). Dart networks: these parameters for fat genomics. arXiv preprint arXiv:1411.0154.
Scheele, D. (2015). Deep learning in neural networks: An overview. Neurocomputing, 181, 85-117.
Sengupta, C. et al (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1-9.
Sohn, D. and Hoi, D. (2017). Impact causal models for genome-wide association studies. arXiv preprint arXiv:1710.10742.
Uppal, S. and Kolluri, A. (2017). Tuning hyperparameters for gene interaction models in genome-wide association studies. In International Conference on Neural Information Processing, pages 791-800. Springer.
van Es, M. et al (2009). Genome-wide association study identifies 1p31.3 (unc15a) and 9q34.5 as susceptibility loci for sporadic amyotrophic lateral sclerosis. Nature genetics, 41(9), 1063.
van Rietbergen, W. et al (2016). Genome-wide association analysis identifies novel risk variants and the genetic architecture of amyotrophic lateral sclerosis. Nature genetics, 48(9), 1041.
Venkat, V. (1990). The support vector method of function estimation. In Handbook of Neural Computing, pages 59-65. Springer.
Visscher, P. et al (2017). 10 years of pegas discovery: biology, function, and translation. The American Journal of Human Genetics, 101(1), 5-22.
Wang, N. et al (2013). Potentials of predicting complex traits from snps. Nature Reviews Genetics, 14(7), 507.
A Details of the network architecture

Fig. 7. Network architecture of Block 1. c is the number of chromosomes included in the model.

Fig. 8. Network architecture of Block 2. Inputs are given by “Genes” (c1, k, d) and “Channels” (d1, d2, d3) in Figure 6, and the shape of the input tensor is (HxWx1).

Fig. 9. Network architecture of Block 3. Inputs are given by “Channels” (d1, d2, d3) in Figure 6, and the shape of the input tensor is (HxWx1).

Fig. 10. Network architecture of Block 4. Inputs are given by “Genes” (c1, k, d) and “Channels” (d1, d2, d3) in Figure 6, and the shape of the input tensor is (HxWx1).

B Performance of alternative network architectures

Table 5. The effect of small changes in network architecture on the classification performance for chromosome 7. Acc - accuracy, Prec - precision, Rec - recall, F1 - F1 score. The layers have 1024 nodes, 512 nodes, and 256 nodes, respectively, with a SELU, a ReLU, and a SELU activation function, respectively.

| Model change | Acc | Prec | Rec | F1 score |
|--------------|-----|------|-----|----------|
| Block 1: change output size from | 0.556 | 0.720 | 0.510 | 0.597 |
| (96, 256) to (96, 128) | | | | |
| Block 2: change Sep. Conv. into Conv. | 0.638 | 0.683 | 0.515 | 0.588 |
| Final dense layer: 128 nodes | 0.662 | 0.704 | 0.539 | 0.623 |
| Set learning rate = 0.0005 | 0.656 | 0.699 | 0.583 | 0.636 |
| Select 4 best parameter regions | 0.672 | 0.670 | 0.629 | 0.674 |
| Select 12 best parameter regions | 0.681 | 0.677 | 0.691 | 0.684 |
| Three-layer MLP | 0.628 | 0.711 | 0.430 | 0.535 |

"output" — 2019/11/4 — page 10 — #10
## C Classification accuracies of known ALS genes

Table 6. Accuracies for ALS-associated genes of chromosomes 7, 9 and 17 as listed on http://ghr.nlm.nih.gov/gene/ (Amo et al., 2013). The ALS-associated genes from Abel et al. (2013) for chromosome 22 were not kept in our dataset after QC, and hence no results are reported.

| Chr | Gene | Accuracy |
|-----|------|----------|
| Chr3 | GARS | 0.5159 |
|     | RAMP3 | 0.503 |
|     | ZNF746 | 0.524 |
|     | DPP6 | 0.532 |
| Chr9 | SUSD1 | 0.504 |
|     | ALAD | 0.513 |
|     | STEX | 0.549 |
| Chr17 | MAPT | 0.514 |
|     | SLC39A11 | 0.511 |

## D Gene ontology of selected chromosome 7 promoter regions

Table 7. Gene ontology terms of the promoter regions that were selected by Promoter-CNN for chromosome 7.

| Gene                  | Gene ontology class(es)                                                                 |
|-----------------------|----------------------------------------------------------------------------------------|
| LAMB4                 | Basement membrane, cell adhesion                                                       |
| LOC105375113          | None - RNA Gene, affiliated with the miRNA class                                         |
| LOC105375507          | None - RNA Gene, affiliated with the miRNA class                                         |
| TRY2P                 | None - trypsin-like pseudogene                                                         |
| TYW1B                 | rRNA processing, FMN binding, metal ion binding, 4 ion 4 sulfur cluster binding, oxidation-reduction process, tRNA-4-demethylcytosine synthase activity |
| LOC101829756          | None - RNA Gene, affiliated with the miRNA class                                         |
| DTX2                  | None for Homo sapiens. For other organisms: metal ion binding, zinc ion binding, protein ubiquitination, notch signalling pathway, cellular component, nucleoplasm |

"output" — 2019/11/4 — page 11 — 011