Article

Molecular Classification and Interpretation of Amyotrophic Lateral Sclerosis Using Deep Convolution Neural Networks and Shapley Values

Abdul Karim [1], Zheng Su [1,2], Phillip K. West [1], Matthew Keon [1,*], The NYGC ALS Consortium [3,†], Jannah Shamsani [1], Samuel Brennan [1], Ted Wong [1], Ognjen Milicevic [1], Guus Teunisse [1], Hima Nikafshan Rad [4] and Abdul Sattar [4]

[1] GenieUs Genomics, 19a Boundary St, Darlinghurst, NSW 2010, Australia; Abdul@genieus.co (A.K.); john@genieus.co (Z.S.); phillip@genieus.co (P.K.W.); jannah@genieus.co (J.S.); sam@genieus.co (S.B.); ted@genieus.co (T.W.); oggy@genieus.co (O.M.); guus@genieus.co (G.T.)
[2] School of Biotechnology and Biomolecular Sciences, Faculty of Science, The University of New South Wales, Sydney, NSW 2033, Australia
[3] The New York Genome Center, 101 Avenue of the Americas, New York, NY 10013, USA; cgnd_help@nygenome.org
[4] Institute of Integrated and Intelligent Systems, Griffith University, Nathan, QLD 4111, Australia; hima.nikafshanrad@griffithuni.edu.au (H.N.R.); a.sattar@griffith.edu.au (A.S.)
* Correspondence: mattk@genieus.co; Tel.: +61-484000656
† Principal investigators of the NYGC ALS Consortium are listed in Table S1.

Abstract: Amyotrophic lateral sclerosis (ALS) is a prototypical neurodegenerative disease characterized by progressive degeneration of motor neurons to severely effect the functionality to control voluntary muscle movement. Most of the non-additive genetic aberrations responsible for ALS make its molecular classification very challenging along with limited sample size, curse of dimensionality, class imbalance and noise in the data. Deep learning methods have been successful in many other related areas but have low minority class accuracy and suffer from the lack of explainability when used directly with RNA expression features for ALS molecular classification. In this paper, we propose a deep-learning-based molecular ALS classification and interpretation framework. Our framework is based on training a convolution neural network (CNN) on images obtained from converting RNA expression values into pixels based on DeepInsight similarity technique. Then, we employed Shapley additive explanations (SHAP) to extract pixels with higher relevance to ALS classifications. These pixels were mapped back to the genes which made them up. This enabled us to classify ALS samples with high accuracy for a minority class along with identifying genes that might be playing an important role in ALS molecular classifications. Taken together with RNA expression images classified with CNN, our preliminary analysis of the genes identified by SHAP interpretation demonstrate the value of utilizing Machine Learning to perform molecular classification of ALS and uncover disease-associated genes.

Keywords: machine learning; ALS; classification; interpretation; target identification

1. Introduction

Amyotrophic lateral sclerosis (ALS) refers to a group of rare neurological disorders in which nerve cells (neuron) functionality to control voluntary muscle movement such as chewing, walking and talking is jeopardized [1–3]. The disease results in a progressive loss of muscle strength leading to paralysis and eventually death [2]. Genetic aberration is one of the primary causes of ALS for many patients [2,4]. Most of these genetic aberrations are non-additive because of their interaction with each other which makes it challenging to be detected using classical available genotype-phenotype association approaches [2]. ALS is now recognized as a multidimensional spectrum disorder. Recently, deep learn-

ing techniques have been proven to be widely used for predicting genotype-phenotype associations and molecular ALS classifications [5–8]. The ability of deep learning models to effectively extract nonlinear relationships from a large number of samples for complex disorders has been reported in the literature [9,10].

The molecular ALS classification is also a very complex problem [11] and ideally would require thousands of samples [12] to train any deep learning algorithm. ALS is a rare disease, and it is challenging to find a large number of samples for research purpose [12,13]. Another factor that limits the availability of samples is the accessibility of affected tissues. ALS is a disease of motor neurons, which reside in the spinal cord and the brain [14]. Postmortem spinal cord, brain and cerebrospinal fluid are ideal tissue sources which either directly reflect the pathology of the disease or have interaction with central nervous system, but they are usually more difficult to access compared to tissue such as blood [15]. In a typical dataset of rare disease (ALS) samples, the number of samples is far less than the number of expressed genes [16,17], thus introducing the curse of dimensionality [18].

Another challenge with study of RNA expression, which is the gene transcription activity represented by the count of reads mapped to the gene in next generation sequencing data, is tissue heterogeneity and cell composition heterogeneity. Different human tissues have distinct RNA expression patterns [19]; furthermore, it has been found that ALS patients and healthy individuals have different cell compositions in same tissue [20], and these can be confounding factors in disease-associated gene expression identification. As we are using postmortem samples in the study, RNA quality can be easily impacted by sample collection time and storage condition, which in turn influences data quality and gene expression quantification. This is another confounding factor that makes the data analysis challenging.

Besides limited sample size, the curse of dimensionality and noise of the data, rare disorders data also suffer from a severe class imbalance problem [21,22]. In machine learning, one of the important criteria for higher classification accuracy is a balanced dataset [22]. Datasets with a large ratio between minority and majority classes face hindrance in learning using any classifier [22]. In order to cater for dimensionality curse and class imbalance for molecular classification of ALS, we propose an end-to-end machine-learning-based pipeline for ALS and control samples classification and interpretation. We used RNA expression data for control (60) and ALS (490) samples, and each sample is represented in RNA expression values. The RNA expression values for each sample are mapped to form a pixel value of an image, thus creating an image dataset. We utilized DeepInsight package [23] for image creation and a convolutional neural network (CNN) for classification. We used various subsets of RNA expression data with our classification model. Using this approach, even with a small size, highly noisy and severe class imbalance dataset, we achieved promising classification results. Our method achieves better performance for a minor class in comparison to other classical methods such as a fully connected neural network trained, random forest and support vector machines trained on RNA expression values directly.

In addition to molecular classification of ALS, we also employed SHAP (Shapley additive explanations) to interpret the prediction results of our classification module [24]. We identified the top 10 pixels with the highest SHAP values and investigated the genes which these pixels represented. The model found known ALS-associated genes and predicted potential new disease genes. we demonstrated the value of utilizing machine learning to perform molecular classification of ALS. In this study, we show that our image-based neural network approach is able to perform effective feature selection, learn nonlinear relationship in highly noisy data and identify biologically relevant molecular signals.

2. Materials and Methods

In this section, we first describe the dataset and the performance evaluation criteria used in this study. Then, we provide details about the developed pipeline and its major parts such as image creation module, classification module and post hoc interpretation module.

2.1. Datasets

New York Genome Center (NYGC) RNASeq data were used for this study, and RNA extraction, library preparation and sequencing were performed by NYGC under their protocol. In brief, total RNA was extracted from flash frozen postmortem tissues of ALS and control samples, using trizol/chloroform method, followed by Qiagen RNeasy minikit column purification. RNA was quantified using Nanodrop 2000 and Qubit™ 2.0 Fluorometer, and its quality was measured by RNA integrity number (RIN) scores on Agilent Bioanalyzer. Libraries were prepared using KAPA Stranded RNA-Seq Kit, then loaded to Illumina HiSeq 2500 sequencer for 2×125 bp paired-end sequencing.

After receiving the raw sequencing data in fastq format, we performed quality control using FASTQC [25], with mean quality value across each base position in the read and per-sequence quality scores as the main metrics for data quality evaluation. The sequences were pseudo aligned to reference genome of GRCh38 from Ensembl release 95 [26] by Kallisto [27] for RNA expression quantification. GTF file from Ensembl release 86 was used for gene region annotation. The transcript abundance quantified by Kallisto was used for downstream machine learning.

As sex chromosomes have different ploidies in different genders, we only use genes in autosomes, i.e., chromosomes 1–22 in our pipeline. We also excluded loci with multiple haplotypes in GRCh38 (such as MHC locus), where accurate expression quantification is challenging. As genes with low read count carry little information and can be caused by mapping errors, we conducted a case study which only used high-expression genes, which are genes with ≥ 10 reads in ≥ 10 samples, for model training. In another case study, we used only protein-coding genes for training, which are defined as protein-coding genes in Gencode Release 26 [28].

2.2. Evaluation Criteria

In order to measure the ALS classification performance of our developed pipeline, we used the following metrics: area under curve of receiver operating curve (AUC-ROC), specificity (SPE), sensitivity (SEN), negative predictive value (NPV), positive predictive value (PPV), accuracy (ACC) and Matthew's correlation coefficient (MCC). It should be noted in this paper that negative class refers to control, and positive class refers to ALS. The details of these metrics are as follows:

- Area under curve of receiver operating curve (AUC-ROC): AUC-ROC takes into account all the threshold. The higher the value of AUC-ROC, the better the model is at distinguishing between classes. It can be computed by taking area under the curve for true-positive rate (TPR) on the y-axis and false-positive rate (FPR) on the x-axis for a given dataset. TPR which is also called sensitivity (SEN) describes how good the model is at classifying a sample as ALS when the actual outcome is also ALS. FPR describes how often a ALS class is predicted when the actual outcome is control.

$$\text{SEN} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives, and SEN = Sensitivity.
- Specificity (SPE): SPE is the total number of true negatives divided by the sum of the number of true negatives and false positives. Specificity would describe what proportion of the negative class were correctly classified by our model.

$$\text{SPE} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3}$$

- Negative predictive value (NPV): NPV describes the probability of a sample predicted as negative class to be actually as negative class.

$$NPV = \frac{TN}{TN + FN} \qquad (4)$$

- Positive predictive value (PPV): PPV describes the probability of a sample predicted as positive class to be actually as positive class.

$$PPV = \frac{TP}{TP + FP} \qquad (5)$$

- Accuracy (ACC): ACC is the fraction of prediction our model was correct about , i.e., it predicted positive class and negative class correctly.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \qquad (6)$$

- Matthews correlation coefficient (MCC): MCC has a range from -1 to 1 where -1 indicates a completely wrong binary classifier while 1 indicates a completely correct binary classifier.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \qquad (7)$$

2.3. Image Creation Module

We transform RNA expression data into pixels using the DeepInsight similarity technique and then trained a convolution neural network (CNN).
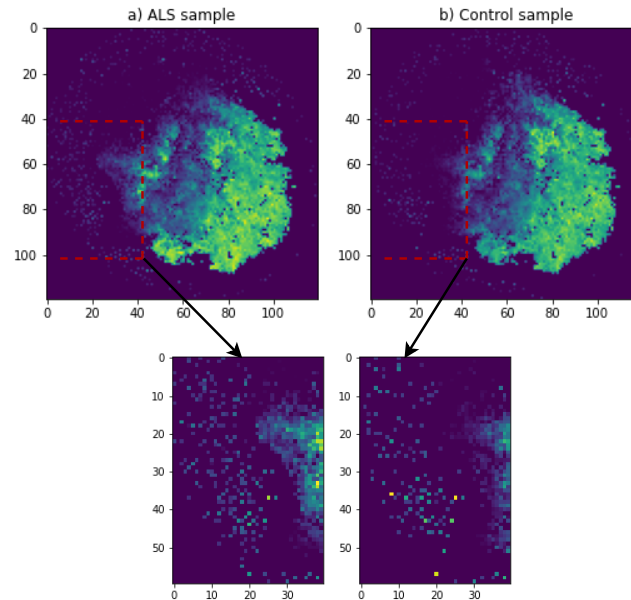

Figure 1. ALS and control sample images with 120×120 resolution obtained using DeepInsight for demonstration purposes.

In Figure 1, DeepInsight for RNA expression data is used to create images of ALS and a control group sample. DeepInsight image creation process is shown in Figure 2. The RNA expression values were mapped to a 2D matrix such that the features which are similar to each other occupy nearby positions in the matrix. As shown in Figure 2a, a transformation T was applied on the genes feature vectors for each sample which creates a 2D matrix M. Features g1, g3, g6 and gd are closer to each other, which brings them into each other's vicinity after applying the transformation T. On the other hand, gene feature g7 is different

than the others and mapped to a very different location in 2D matrix. Figure 2b shows each step of the transformation T.

The first step is to find the location of each gene feature. For that purpose, similarity measuring technique or dimensionality reduction technique like t-SNE or kernel principal component analysis (kPCA) is applied sample-wise on the gene features data. This results in feature locations in 2D plane. Once the location of the gene features of samples is determined, then convex hull algorithm is used to find the smallest rectangle that contains all the points. Rotation is performed to obtain images in horizontal or vertical orientation only. Then the gene features are mapped to their respective positions obtained in the previous step. Thus a new image data is created for each sample where each pixel correspond to one or more gene features. In case of multiple genes very similar to each, their values are averaged out to obtain a pixel.
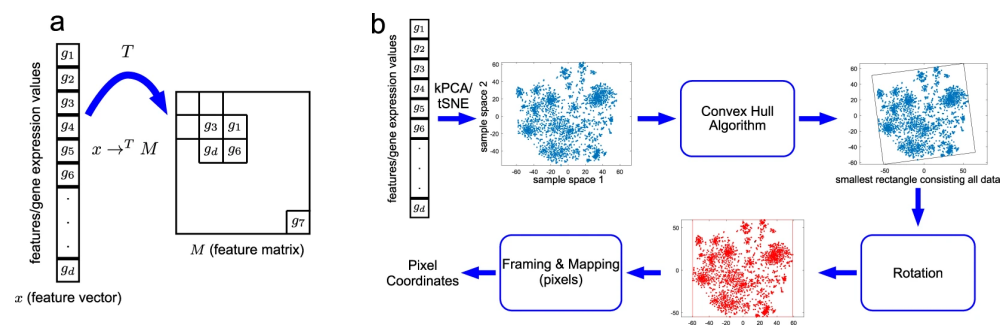

Figure 2. DeepInsight pipeline. (**a**) An illustration of transformation from feature vector to feature matrix. (**b**) An illustration of the DeepInsight methodology to transform a feature vector to image pixels. Image taken from DeepInsight [23].

In specific implementation for RNA expression data as gene features in our case, we first normalized the expression values to a range of [0, 1]. We used normalization method named as norm-2 from DeepInsight [23]. In this normalization, the topology of features was reserved to some extent of DeepInsight [23]. After normalization, we used a python package of DeepInsight [29] with t-SNE method to generate single-channel images with a dimension of 380×380 in our final model. For the demonstration purposes, we show an image of 120×120 in Figure 1. Major portion of both ALS and control was similar, yet subtle changes can be observed, one of which is highlighted in Figure 1.

2.4. Classification Module

After obtaining image data from RNA expression data, we used convolutional neural network (CNN) to classify images into ALS and control. A convolutional neural network is a special type of neural network for the image data. CNNs can extract low-level features from images and compute more complex features as we go deeper in the networks [30,31]. Variants of CNN such as Inception, Alexnet and Resnet have been developed and employed as highly accurate image classification models [32]. In our particular case, as shown in Figure 3, there was an input, 2 conv blocks, then fully connected block followed by an output block. Input contained image data of 490 ALS and 60 control samples. Each sample was 380×380 single-channel image. There were two conv blocks concatenated after the input. Each of the conv block consisted of one 2D convolution layer followed by ReLU activation, max pooling and drop out as shown in Figure 3. The depth d of 2D convolution was 32 in first conv block and 64 in the second conv block. In fully connected block, a dense layer with 256 units along with ReLU activation was used after flattening the output of the conv block. A single dense unit followed by sigmoid function was applied at the end of the fully connected block. At the output, the sample was considered as ALS for prediction probability greater than 0.5 and control for prediction probability less than 0.5.

CNN Training

We used Keras deep learning framework [33] for developing and training the CNN model. We trained it for 500 epochs with an early stopping criteria. During the training, we used class weights for computing the loss function to cater for class imbalance in our data. ALS class was assigned with a weight of 0.56, and control was assigned with a weight of 4.57 using sklearn class_weight function [34]. We used ADAM optimizer [35] with a binary cross entropy loss function from Keras [33] for training our CNN model.
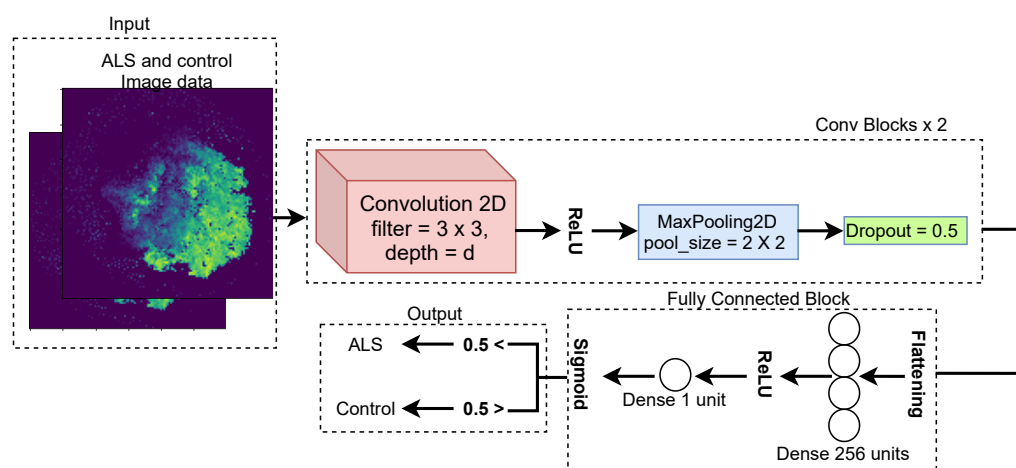

Figure 3. CNN architecture for classifying ALS and control images.

2.5. Classical Machine Learning Methods

We used random forest (RF), support vector machines (SVM) and a fully connected neural network (FCNN) with 33,153 RNA expression features related to autosomal genes directly to compare their results with our method.

2.5.1. Random Forest (RF)

Random forests are a combination of tree predictors, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [36]. We used RF from scikit-learn machine learning library with its default parameters [37].

2.5.2. Support Vector Machines (SVM)

Support vector machines (SVM) belongs to a class of supervised machine learning methods. It attempts to find a line/hyper-plane (in multidimensional space) that separates classes of data under observation or ranges for regression [38,39]. We used SVM from scikit-learn machine learning library with its default parameters [37,40].

2.5.3. Fully Connected Neural Networks (FCNN)

We also compared our method with a fully connected neural network (FCNN). FCNN can be viewed as a complex mapping function, where the fundamental unit of a FCNN is called a neuron. It takes input and computes the output after applying non-linearity and gradient descent based back-propagation algorithm [41]. In specific implementation for this study, FCNN consisted of two fully connected layers with 200 neurons in each, a second last layer with 10 neurons and an output layer with one neuron. We placed a dropout rate of 0.5 after each hidden layer. We used Keras deep learning framework [33] for training. We trained it for 100 epochs with an early stopping criteria. We used ADAM optimizer [35] with a binary cross entropy loss function from Keras [33] and a batch size of 32 with a learning rate of 0.001.

2.6. Post Hoc Interpretation Module

Deep learning methods such as CNN are black-box in nature and extremely difficult to interpret [42,43]. These methods are capable to answer the "what" question about certain prediction but fails to give an answer to the "why" question [43,44]. Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications [24]. In this study, we used SHAP (Shapley additive explanations) to interpret the prediction results of our classification module. SHAP assigns each feature an importance value for a particular prediction. It connects game theory with local explanations and uniting several previous methods [44–47]. SHAP represents the only possible consistent and locally accurate additive feature attribution method based on expectations [24].

In this study, once we have trained and tested our model, we employed Deep SHAP package [48] to interpret the prediction outcome of our classification module. In the first step, we selected a distribution of background 200 random samples out of the input image data to take expectations over. Then, we used the selected background distribution along with the trained model to obtain SHAP values for each pixel in a sample as shown in Figure 4. Red pixels represent positive SHAP values that increase the probability of the class, while blue pixels represent negative SHAP values that reduce the probability of the class. The sum of the SHAP values equals the difference between the expected model output (averaged over the background dataset) and the current model output. After obtaining SHAP values of each pixel, we selected the top ten pixels with the highest SHAP values and mapped them back to specific genes forming those pixels.
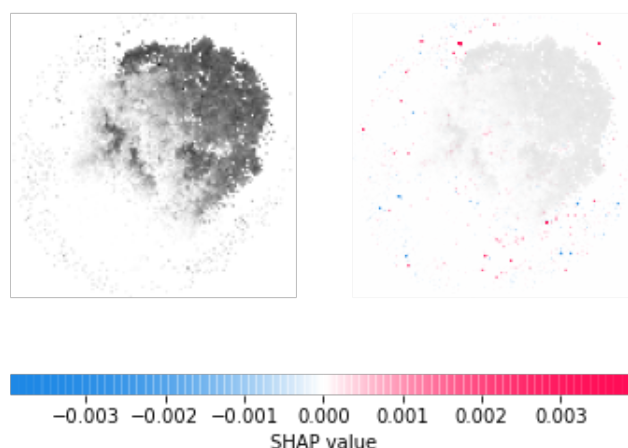

Figure 4. Left side is gray-scale image of an ALS sample. Right side shows highlighted pixels in the image with SHAP values.

3. Results and Discussion

In this section, we present the classification performance of our method with different image resolutions and RNA expression features related to various sets of genes. We also show our method's performance compared with classical machine learning models and the top 10 gene extracted using SHAP interpretation of our method. Finally, we investigate the functional and disease association of the extracted genes and then discuss some potentially identified new genes.

3.1. Samples and Quality Controls

Fastq files of 550 samples (490 ALSs and 60 controls) were downloaded from NYGC database; all of them passed the default quality control criteria of mean quality value in the read and per sequence quality scores of FASTQC; and all samples were used for downstream analysis. The samples are from 114 ALS patients and 19 control individuals. The majority of patients and control individuals donate multiple tissues, including cerebellum (n = 98), frontal cortex (96), motor cortex (118), occipital cortex (58), and spinal cord (163)

(Table S2), and the composition of tissues is similar between ALSs and controls (Table S4). We quantified the expression of 39,429 genes (33,153 autosomal and 6276 non-autosomal genes), and the expression data of autosomal genes were used for model training.

RIN is one of most commonly used metrics for RNA quality control. It has been shown that RIN values impact RNA sequencing data quality and gene expression quantification [49]. The suggested threshold of RIN value for sample inclusion varied in different studies, and it can be as high as 8 [50] and as low as 3.95 [51]. The samples in this study have RIN values ranging from 2.2 to 9.9 (Figure S1), and we did not filter out samples with low RNA quality using a RIN threshold, as it can reduce the power of the analysis [49]; by contrast, we just used all samples for model training and left the task of distinguishing biological signals from RNA quality confounding effects to the model. It should be noted that even though we did not apply any filter on the RIN values, there should not be any bias in RIN values between two groups; otherwise, it can confound the prediction. There is no significant difference in the distribution of RIN values in two groups (*p*-value = 0.70, *t*-test) as shown in Figure S2 of the Supplementary Material.

3.2. Classification Performance for Various Image Resolutions

We present the effects of different image resolutions on the classification performance of our method. For this purpose, we used RNA expression features of 33,153 autosomal genes. We performed 12-fold cross-validation experiments with image resolutions starting from 50 × 50 till 380 × 380 as shown in Figure 5. AUC and ACC improves with higher image resolutions. The highest value of AUC: 0.927 is obtained for the resolution of 320 × 320, whereas for that of ACC: 0.827 is obtained for the resolution of 380 × 380. For F1 and MCC, we observe an initial improvement with a dip in between around 220 and 250 followed by an improvement until 380 × 380. The highest values for both F1: 0.813 and MCC: 0.639 are obtained for the resolution of 380 × 380. As the data are highly imbalanced with ALS as major class and control as minor class, we see higher sensitivity with little variations as compared to specificity for all image resolutions. SPE increases continuously with higher image resolution with a highest value of 0.706 for 380 × 380. The ability of our model to correctly identify minor class which is control in our case improves significantly with higher resolution images. For PPV on the other hand, there is a slight improvement with a highest value of 0.963 for 380 × 380. The highest value for NPV however occurs at a resolution of 180 × 180, which is 0.790. Moreover, with higher resolutions, the standard deviation for 12 folds increases, as shown by the black error bars in Figure 5.
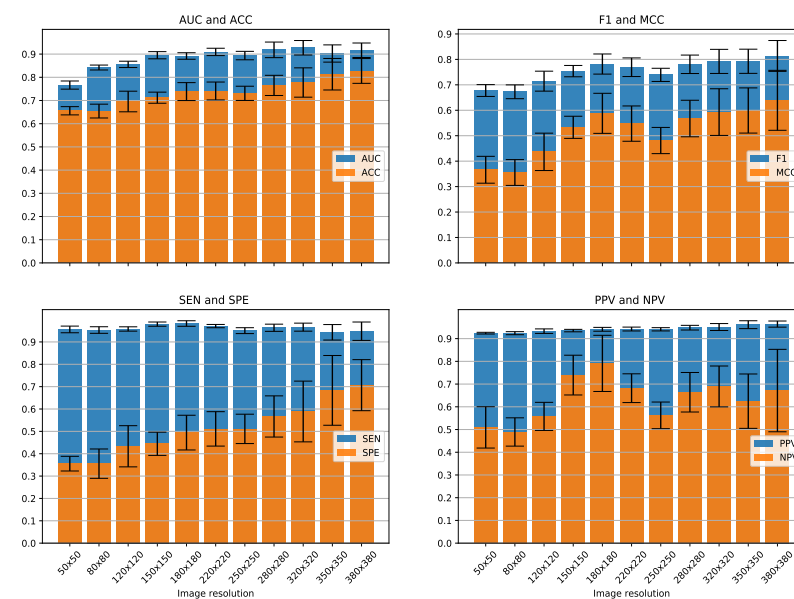

Figure 5. 12-fold cross-validation performance for creating images of various resolutions.

3.3. Comparison with Classical Models

In order to investigate the effectiveness of our method, we compared its results with classical machine learning methods such as random forest, support vector machine and fully connected neural networks. As shown in Table 1, our method performs significantly better in most of the classification metrics. Specifically in classifying the control class correctly, our method proves to be very robust as shown by SPE value in Table 1. Both RF and SVM achieves very poor performance for classifying the minor class correctly. FCNN performs relatively well though as compared to RF and SVM for SPE. A relatively higher value of AUC for RF and SVM is strongly influenced by the major class, which is most of the time predicted correctly. AUC for our method is the highest as compared other classical methods. Nearly all the methods are performing well for classifying the major class as shown by the SEN value in Table 1. SVM achieves the highest NPV followed by RF, FCNN and our method, respectively. For PPV, ACC, MCC and F1, our methods achieves 3.54%, 20.37%, 33.68% and 12.44% improvement over the second-best FCNN method.

Table 1. 12-fold cross-validation performance comparison with classical machine learning methods such as random forest (RF), support vector machines (SVM) and fully connected neural network (FCNN). For our method while comparing with classical methods, we used images with resolution of 380×380 and all 33,153 RNA expression features of autosomal genes.

Method	AUC	SPE	SEN	NPV	PPV	ACC	MCC	F1
Our method	**0.917** $\pm$ 0.03	**0.707** $\pm$ 0.11	0.947 $\pm$ 0.04	0.671 $\pm$ 0.18	**0.963** $\pm$ 0.01	**0.827** $\pm$ 0.05	**0.639** $\pm$ 0.11	**0.813** $\pm$ 0.06
RF	0.831 $\pm$ 0.04	0.155 $\pm$ 0.05	0.994 $\pm$ 0.00	0.798 $\pm$ 0.19	0.906 $\pm$ 0.00	0.575 $\pm$ 0.02	0.319 $\pm$ 0.09	0.602 $\pm$ 0.04
SVM	0.866 $\pm$ 0.05	0.083 $\pm$ 0.03	**1** $\pm$ 0	**1** $\pm$ 0	0.899 $\pm$ 0.00	0.541 $\pm$ 0.01	0.270 $\pm$ 0.04	0.549 $\pm$ 0.02
FCNN	0.805 $\pm$ 0.04	0.4 $\pm$ 0.12	0.974 $\pm$ 0.02	0.692 $\pm$ 0.20	0.930 $\pm$ 0.01	0.687 $\pm$ 0.06	0.478 $\pm$ 0.15	0.723 $\pm$ 0.07

3.4. Case Study of Classification Performance with High Count and Protein-Coding Genes

In the initial phase of this study, we used expression values of all available 33,153 autosomal genes to train our pipeline with 380×380 image resolutions. We observed that even though higher resolution of images give better classification performance, it also increases the computational complexity and run-time of the pipeline. Therefore, we chose the resolution of 350×350 for two case studies to investigate the effectiveness of high-expression genes and protein-coding genes.

- High-expression genes: Many of the 33,153 autosomal genes only express in very few samples and carry little information; thus, we filtered out those genes with low expression and only included genes with a high read count in a certain number of samples. For that purpose, we used a threshold of 10, i.e., at least 10 samples across our training data have a read count of 10 or higher. Through this filtering strategy, we obtained a total of 18,194 high expression genes. The RNA expression values of these high read count genes were converted into 350×350 images and subsequently used in CNN training for classification.
- Protein-coding genes: As including non-protein-coding genes in the training data may only increase the model complexity but bring little benefit to the model, thus in the second case study, we also selected RNA expression data of 19,724 protein-coding genes, converted them into images with resolution of 350×350 and evaluated the performance of CNN model trained with those images.

As shown in Table 2, for all the metrics, RNA features for high-expression genes substantially improve the classification performance as compared to 33,153 autosomal gene expression results shown in Table 1. Protein-coding genes however show a slight decrease in the classification performance.

Table 2. 12-fold crossvalidation classification performance with a resolution of 350×350 high-expression and protein-coding genes RNA features.

RNA Features	AUC	SPE	SEN	NPV	PPV	ACC	MCC	F1
High count genes	**0.964** $\pm$ 0.04	**0.776** $\pm$ 0.12	**0.978** $\pm$ 0.00	**0.809** $\pm$ 0.00	**0.973** $\pm$ 0.01	**0.877** $\pm$ 0.06	**0.767** $\pm$ 0.10	**0.882** $\pm$ 0.05
Protein-coding genes	0.910 $\pm$ 0.04	0.646 $\pm$ 0.13	0.968 $\pm$ 0.02	0.720 $\pm$ 0.12	0.957 $\pm$ 0.01	0.807 $\pm$ 0.07	0.643 $\pm$ 0.13	0.819 $\pm$ 0.06

3.5. SHAP Interpretation

We used SHAP interpretation [24] to investigate the role of each gene in classifying ALS samples using our developed model. It should be noted that each pixel of the image may contain one or more gene expression values. It should be noted that for SHAP interpretation, we used the model obtained with high-count gene RNA expression features with image resolution of 350×350. For each prediction of our model discussed previously, we identified top 10 pixels having highest SHAP values. We identified 12 genes (Table S3) which appeared in the top 10 pixels in more than 200 samples. We are currently investigating 10 of the genes further; however, two genes in our dataset, Survival of motor neuron-1 (SMN1) and SMN2, have been previously classified as associated with disease in ALS [52]. SMN1 and SMN2 are paralogous genes as the result of an inverted duplication [53]. The SMN protein has a myriad of roles in motor neuron function and is critical for the regulation of transcription and RNA maturation, axonal trafficking of RNA transcripts, facilitating the binding of RNA-binding proteins to mRNA transcripts and regulating cytoskeletal dynamics [54]. In addition, SMN is involved in protein degradation pathways by its actions in autophagy and the ubiquitin-proteasome system (UPS) and may contribute to mitochondrial function through regulation of splicing, translation or mRNA transport of genes crucial for mitochondrial homeostasis [54]. Therefore, the SMN protein encoded by SMN1 and SMN2 contributes to many distinct pathways implicated in the pathology of ALS.

In addition, SMN interacts with, and its properties are modulated by, other proteins known to contribute to ALS pathogenesis, such as FUS, SOD1 and TDP-43 [55]. The interaction between ALS and FUS is enhanced by ALS-associated mutations in FUS, causing these proteins to form a stable complex [56]. As a result, SMN is mislocalized and sequestered into cytoplasmic FUS aggregates, leading to a decrease in SMN in the axons of neurons and subsequent axonal defects [56]. These aberrations also result in the loss of small nuclear bodies, dysregulated small nuclear ribonucleoprotein (snRNP) assembly and defects in downstream RNA processing [57,58]. Similarly, SOD1 variants cause the mislocalization of SMN and disrupt the formation of nuclear bodies [59,60]. In vitro over-expression of SMN enhances chaperone activity and protects cells from mutant SOD1 toxicity [61]. Furthermore, over-expression of SMN in SOD1 or TDP-43 mutant mice ameliorates disease [62,63], while SOD1 mutant mice exhibit accelerated disease severity when SMN is depleted [64]. Therefore, the currently available evidence indicates that, in addition to its function in relevant biological pathways affected during ALS, the SMN protein may cooperate with other ALS-associated genes to coordinate and modify the disease phenotype of ALS. It should be noted that there are changes in cell-type composition in ALS as found in previous studies [20,65]; therefore, in the interpretation of top genes found in our model, we should be cautious that they can arise from this confounding factor. Taken together, our preliminary analysis of the genes identified by SHAP interpretation demonstrate the value of utilizing Machine Learning to perform molecular classification of ALS and uncover potential disease-associated genes.

4. Conclusions

In conclusion, we developed a deep learning framework, which took full advantage of image recognition ability of convolutional neural network by transforming the gene expression data of ALS into images and then used them for neural network training. We showed that the model effectively extracted disease associated features and learned the

nonlinear gene-disease relationship in highly noisy, heterogeneous and imbalanced data, and we also showed its superior performance in disease clarification over other machine learning algorithms. We interpreted the model with SHAP and successfully identified known disease associated genes, and some potential new disease genes, which demonstrated the potential of our model in the new biomarker and drug target identification in complex disease research.

Abbreviations

The following abbreviations are used in this manuscript:

ALS	Amyotrophic Lateral Sclerosis
FUS	Fused In Sarcoma
MND	Motor Neuron Disease
SMN	Survival of Motor Neuron
SOD1	Superoxide Dismutase 1
TDP-43	TAR DNA-binding protein 43

References

1. Phukan, J.; Pender, N.P.; Hardiman, O. Cognitive impairment in amyotrophic lateral sclerosis. *Lancet Neurol.* **2007**, *6*, 994–1003. [CrossRef]
2. Yin, B.; Balvert, M.; van der Spek, R.A.; Dutilh, B.E.; Bohté, S.; Veldink, J.; Schönhuth, A. Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype. *Bioinformatics* **2019**, *35*, i538–i547. [CrossRef] [PubMed]
3. Amyotrophic Lateral Sclerosis (ALS) Fact Sheet | National Institute of Neurological Disorders and Stroke. Available online: https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Amyotrophic-Lateral-Sclerosis-ALS-Fact-Sheet (accessed on 15 February 2021).
4. Van Rheenen, W.; Shatunov, A.; Dekker, A.M.; McLaughlin, R.L.; Diekstra, F.P.; Pulit, S.L.; Van Der Spek, R.A.; Võsa, U.; De Jong, S.; Robinson, M.R.; et al. Genome-wide association analyses identify new risk variants and the genetic architecture of amyotrophic lateral sclerosis. *Nat. Genet.* **2016**, *48*, 1043–1048. [CrossRef] [PubMed]

5.   Arloth, J.; Eraslan, G.; Andlauer, T.F.; Martins, J.; Iurato, S.; Kühnel, B.; Waldenberger, M.; Frank, J.; Gold, R.; Hemmer, B.; et al. DeepWAS: Multivariate genotype-phenotype associations by directly integrating regulatory information using deep learning. *PLoS Comput. Biol.* **2020**, *16*, e1007616. [CrossRef] [PubMed]

6.   Liu, Y.; Wang, D.; He, F.; Wang, J.; Joshi, T.; Xu, D. Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. *Front. Genet.* **2019**, *10*, 1091. [CrossRef]

7.   Drouin, A.; Letarte, G.; Raymond, F.; Marchand, M.; Corbeil, J.; Laviolette, F. Interpretable genotype-to-phenotype classifiers with performance guarantees. *Sci. Rep.* **2019**, *9*, 1–13. [CrossRef] [PubMed]

8.   Aronica, E.; Baas, F.; Iyer, A.; ten Asbroek, A.L.; Morello, G.; Cavallaro, S. Molecular classification of amyotrophic lateral sclerosis by unsupervised clustering of gene expression in motor cortex. *Neurobiol. Dis.* **2015**, *74*, 359–376. [CrossRef]

9.   Baloch, Z.Q.; Raza, S.A.; Pathak, R.; Marone, L.; Ali, A. Machine Learning Confirms Nonlinear Relationship between Severity of Peripheral Arterial Disease, Functional Limitation and Symptom Severity. *Diagnostics* **2020**, *10*, 515. [CrossRef] [PubMed]

10.   Nicholls, H.L.; John, C.R.; Watson, D.S.; Munroe, P.B.; Barnes, M.R.; Cabrera, C.P. Reaching the end-game for GWAS: Machine learning approaches for the prioritization of complex disease loci. *Front. Genet.* **2020**, *11*, 350. [CrossRef] [PubMed]

11.   Zarei, S.; Carr, K.; Reiley, L.; Diaz, K.; Guerra, O.; Altamirano, P.F.; Pagani, W.; Lodin, D.; Orozco, G.; Chinea, A. A comprehensive review of amyotrophic lateral sclerosis. *Surg. Neurol. Int.* **2015**, *6*. [CrossRef]

12.   Grollemund, V.; Pradat, P.F.; Querin, G.; Delbot, F.; Le Chat, G.; Pradat-Peyre, J.F.; Bede, P. Machine learning in amyotrophic lateral sclerosis: Achievements, pitfalls, and future directions. *Front. Neurosci.* **2019**, *13*, 135. [CrossRef]

13.   Mitani, A.A.; Haneuse, S. Small data challenges of studying rare diseases. *JAMA Netw. Open* **2020**, *3*, e201965. [CrossRef]

14.   Rowland, L.P.; Shneider, N.A. Amyotrophic lateral sclerosis. *N. Engl. J. Med.* **2001**, *344*, 1688–1700. [CrossRef]

15.   Agah, E.; Saleh, F.; Moghaddam, H.S.; Saghazadeh, A.; Tafakhori, A.; Rezaei, N. CSF and blood biomarkers in amyotrophic lateral sclerosis: Protocol for a systematic review and meta-analysis. *Syst. Rev.* **2018**, *7*, 1–5. [CrossRef]

16.   Barbour, D.L. Precision medicine and the cursed dimensions. *NPJ Digit. Med.* **2019**, *2*, 1–2. [CrossRef]

17.   Chattopadhyay, A.; Lu, T.P. Gene-gene interaction: The curse of dimensionality. *Ann. Transl. Med.* **2019**, *7*, 24. [CrossRef]

18.   Köppen, M. The curse of dimensionality. In Proceedings of the 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), Online, 4–8 September 2000; Volume 1, pp. 4–8.

19.   Consortium, G.T. The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science* **2015**, *348*, 648–660. [CrossRef] [PubMed]

20.   Dols-Icardo, O.; Montal, V.; Sirisi, S.; López-Pernas, G.; Cervera-Carles, L.; Querol-Vilaseca, M.; Muñoz, L.; Belbin, O.; Alcolea, D.; Molina-Porcel, L.; et al. Motor cortex transcriptome reveals microglial key events in amyotrophic lateral sclerosis. *Neurol.-Neuroimmunol. Neuroinflamm.* **2020**, *7*. [CrossRef] [PubMed]

21.   Li, D.C.; Liu, C.W.; Hu, S.C. A learning method for the class imbalance problem with medical data sets. *Comput. Biol. Med.* **2010**, *40*, 509–518. [CrossRef] [PubMed]

22.   Haque, M.M.; Skinner, M.K.; Holder, L.B. Imbalanced class learning in epigenetics. *J. Comput. Biol.* **2014**, *21*, 492–507. [CrossRef] [PubMed]

23.   Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K.A.; Tsunoda, T. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Sci. Rep.* **2019**, *9*, 1–7. [CrossRef] [PubMed]

24.   Lundberg, S.; Lee, S.I. A unified approach to interpreting model predictions. *arXiv* **2017**, arXiv:1705.07874.

25.   Andrews, S. FastQC: A Quality Control Tool for High throughput Sequence Data. Available online: http://www.bioinformatics. babraham.ac.uk/projects/fastqc (accessed on 27 September 2021).

26.   Yates, A.D.; Achuthan, P.; Akanni, W.; Allen, J.; Allen, J.; Alvarez-Jarreta, J.; Amode, M.R.; Armean, I.M.; Azov, A.G.; Bennett, R. Ensembl 2020. *Nucleic Acids Res.* **2020**, *48*, D682–D688. [CrossRef]

27.   Bray, N.L.; Pimentel, H.; Melsted, P.; Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **2016**, *34*, 525–527. [CrossRef] [PubMed]

28.   Frankish, A.; Diekhans, M.; Ferreira, A.M.; Johnson, R.; Jungreis, I.; Loveland, J.; Mudge, J.M.; Sisu, C.; Wright, J.; Armstrong, J.; et al. GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res.* **2019**, *47*, D766–D773. [CrossRef] [PubMed]

29.   Sharma, A. GitHub-alok-ai-lab/DeepInsight: A Methodology to Transform a Non-Image Data to an Image for Convolution Neural Network Architecture. Available online: https://github.com/alok-ai-lab/DeepInsight (accessed on 25 February 2021).

30.   Karim, A.; Singh, J.; Mishra, A.; Dehzangi, A.; Newton, M.H.; Sattar, A. Toxicity prediction by multimodal deep learning. In *Pacific Rim Knowledge Acquisition Workshop*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 142–152.

31.   Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.

32.   He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

33.   Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 15 February 2021).

34.   Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; et al. API design for machine learning software: Experiences from the scikit-learn project. *arXiv* **2013**, arXiv:1309.0238.

35.   Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

36. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
37. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
38. Andrews, S.; Tsochantaridis, I.; Hofmann, T. Support vector machines for multiple-instance learning. *Adv. Neural Inf. Process. Syst.* **2003**, *15*, 577–584.
39. Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.* **1999**, *10*, 61–74.
40. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. (TIST)* **2011**, *2*, 1–27. [CrossRef]
41. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Neurocomputing: Foundations of Research*; MIT Press: Cambridge, MA, USA, 1988.
42. Karim, A.; Mishra, A.; Newton, M.H.; Sattar, A. Efficient toxicity prediction via simple features using shallow neural networks and decision trees. *ACS Omega* **2019**, *4*, 1874–1888. [CrossRef]
43. Karim, A.; Mishra, A.; Newton, M.; Sattar, A. Machine Learning Interpretability: A Science rather than a tool. *arXiv* **2018**, arXiv:1807.06722.
44. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
45. Štrumbelj, E.; Kononenko, I. Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **2014**, *41*, 647–665. [CrossRef]
46. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning important features through propagating activation differences. In *International Conference on Machine Learning*; PMLR: Sydney, Australia, 2017; pp. 3145–3153.
47. Datta, A.; Sen, S.; Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 598–617.
48. GitHub—Shaoshanglqy/Shap-Shapley. Available online: https://github.com/shaoshanglqy/shap-shapley (accessed on 28 February 2021).
49. Romero, I.G.; Pai, A.A.; Tung, J.; Gilad, Y. RNA-seq: Impact of RNA degradation on transcript quantification. *BMC Biol.* **2014**, *12*, 1–13.
50. Imbeaud, S.; Graudens, E.; Boulanger, V.; Barlet, X.; Zaborski, P.; Eveno, E.; Mueller, O.; Schroeder, A.; Auffray, C. Towards standardization of RNA quality assessment using user-independent classifiers of microcapillary electrophoresis traces. *Nucleic Acids Res.* **2005**, *33*, e56. [CrossRef] [PubMed]
51. Weis, S.; Llenos, I.C.; Dulay, J.R.; Elashoff, M.; Martinez-Murillo, F.; Miller, C.L. Quality control for microarray analysis of human brain samples: The impact of postmortem factors, RNA characteristics, and histopathology. *J. Neurosci. Methods* **2007**, *165*, 198–209. [CrossRef] [PubMed]
52. Abel, O.; Powell, J.F.; Andersen, P.M.; Al-Chalabi, A. ALSoD: A user-friendly online bioinformatics tool for amyotrophic lateral sclerosis genetics. *Hum. Mutat.* **2012**, *33*, 1345–1351. [CrossRef]
53. Miccio, A.; Antoniou, P.; Ciura, S.; Kabashi, E. Novel genome-editing-based approaches to treat motor neuron diseases: Promises and challenges. *Mol. Ther.* **2021**, in press,
54. Chaytow, H.; Huang, Y.T.; Gillingwater, T.H.; Faller, K.M.E. The role of survival motor neuron protein (SMN) in protein homeostasis. *Cell Mol. Life Sci.* **2018**, *75*, 3877–3894. [CrossRef]
55. Bowerman, M.; Murray, L.M.; Scamps, F.; Schneider, B.L.; Kothary, R.; Raoul, C. Pathogenic commonalities between spinal muscular atrophy and amyotrophic lateral sclerosis: Converging roads to therapeutic development. *Eur. J. Med. Genet.* **2018**, *61*, 685–698. [CrossRef] [PubMed]
56. Groen, E.J.; Fumoto, K.; Blokhuis, A.M.; Engelen-Lee, J.; Zhou, Y.; van den Heuvel, D.M.; Koppers, M.; van Diggelen, F.; van Heest, J.; Demmers, J.A.; et al. ALS-associated mutations in FUS disrupt the axonal distribution and function of SMN. *Hum. Mol. Genet.* **2013**, *22*, 3690–3704. [CrossRef]
57. Sun, S.; Ling, S.C.; Qiu, J.; Albuquerque, C.P.; Zhou, Y.; Tokunaga, S.; Li, H.; Qiu, H.; Bui, A.; Yeo, G.W.; et al. ALS-causative mutations in FUS/TLS confer gain and loss of function by altered association with SMN and U1-snRNP. *Nat. Commun.* **2015**, *6*, 6171. [CrossRef]
58. Yamazaki, T.; Chen, S.; Yu, Y.; Yan, B.; Haertlein, T.C.; Carrasco, M.A.; Tapia, J.C.; Zhai, B.; Das, R.; Lalancette-Hebert, M.; et al. FUS-SMN protein interactions link the motor neuron diseases ALS and SMA. *Cell Rep.* **2012**, *2*, 799–806. [CrossRef] [PubMed]
59. Gertz, B.; Wong, M.; Martin, L.J. Nuclear localization of human SOD1 and mutant SOD1-specific disruption of survival motor neuron protein complex in transgenic amyotrophic lateral sclerosis mice. *J. Neuropathol. Exp. Neurol.* **2012**, *71*, 162–177. [CrossRef] [PubMed]
60. Kariya, S.; Re, D.B.; Jacquier, A.; Nelson, K.; Przedborski, S.; Monani, U.R. Mutant superoxide dismutase 1 (SOD1), a cause of amyotrophic lateral sclerosis, disrupts the recruitment of SMN, the spinal muscular atrophy protein to nuclear Cajal bodies. *Hum. Mol. Genet.* **2012**, *21*, 3421–3434. [CrossRef] [PubMed]
61. Zou, T.; Ilangovan, R.; Yu, F.; Xu, Z.; Zhou, J. SMN protects cells against mutant SOD1 toxicity by increasing chaperone activity. *Biochem. Biophys. Res. Commun.* **2007**, *364*, 850–855. [CrossRef]

62. Perera, N.D.; Sheean, R.K.; Crouch, P.J.; White, A.R.; Horne, M.K.; Turner, B.J. Enhancing survival motor neuron expression extends lifespan and attenuates neurodegeneration in mutant TDP-43 mice. *Hum. Mol. Genet.* **2016**, *25*, 4080–4093. [CrossRef]
63. Turner, B.J.; Alfazema, N.; Sheean, R.K.; Sleigh, J.N.; Davies, K.E.; Horne, M.K.; Talbot, K. Overexpression of survival motor neuron improves neuromuscular function and motor neuron survival in mutant SOD1 mice. *Neurobiol. Aging* **2014**, *35*, 906–915. [CrossRef]
64. Turner, B.J.; Parkinson, N.J.; Davies, K.E.; Talbot, K. Survival motor neuron deficiency enhances progression in an amyotrophic lateral sclerosis mouse model. *Neurobiol. Dis.* **2009**, *34*, 511–517. [CrossRef] [PubMed]
65. Liu, W.; Venugopal, S.; Majid, S.; Ahn, I.S.; Diamante, G.; Hong, J.; Yang, X.; Chandler, S.H. Single-cell RNA-seq analysis of the brainstem of mutant SOD1 mice reveals perturbed cell types and pathways of amyotrophic lateral sclerosis. *Neurobiol. Dis.* **2020**, *141*, 104877. [CrossRef] [PubMed]