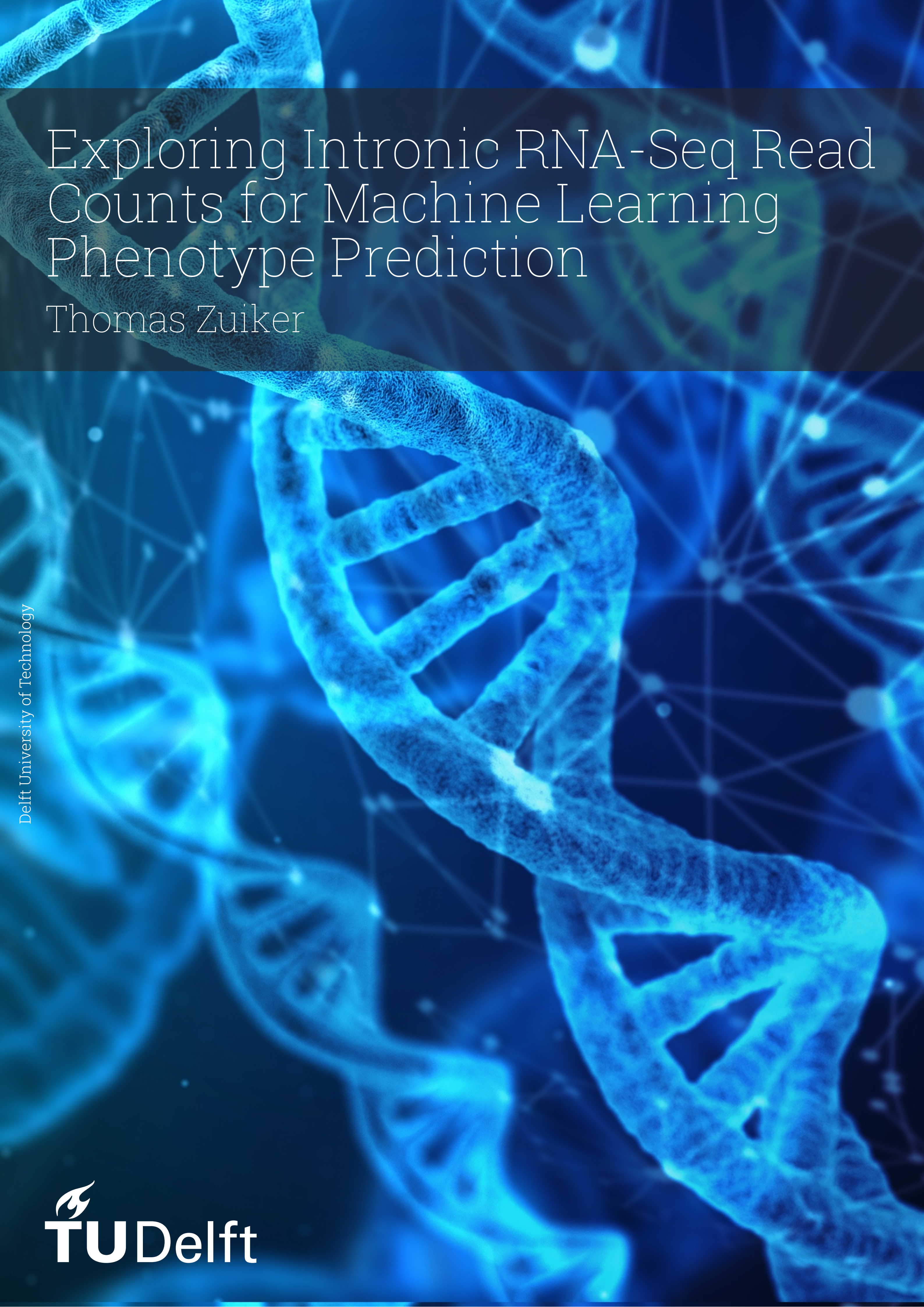


Exploring Intronic RNA-Seq Read Counts for Machine Learning Phenotype Prediction

Thomas Zuiker

Delft University of Technology

Exploring Intronic RNA-Seq Read Counts for Machine Learning Phenotype Prediction

by

Thomas Zuiker

Student number:	4493133
Masters programme:	Computer Science, Bioinformatics specialization
Faculty:	Electrical Engineering, Mathematics and Computer Science
Project Duration:	November, 2022 - November, 2023
Thesis Committee:	Dr. Joana Gonçalves, Pattern Recognition and Bioinformatics
	Dr. Thomas Höllt, Computer Graphics and Visualization
Daily Supervisor:	Roy Lardenoije PhD, Pattern Recognition and Bioinformatics

Exploring Intronic RNA-Seq Read Counts for Machine Learning Phenotype Prediction

Thomas Zuiker,[1,*] Roy Lardenoije[1] and Joana Gonçalves[1]

[1]Pattern Recognition and Bioinformatics, Intelligent Systems Dept., EEMCS Faculty, Delft University of Technology, The Netherlands

Abstract

The inclusion of intronic reads in the downstream analysis of RNA-sequencing (RNA-seq) data has long been controversial. Recent studies show that intronic reads do contain relevant biological signal. Additionally, studies have discovered differential expression unique to intronic reads in certain diseases. Nevertheless, most disease prediction studies only use exonic read counts as input to their models. In this study, we investigate the informativeness of intronic read counts for RNA-seq-based machine learning prediction tasks. Furthermore, we explore possibilities to combine exonic and intronic read counts to increase predictive performance. To this end, we use an RNA-seq dataset originating from four different brain regions and try to predict multiple different clinical labels, including Alzheimer's disease and dementia. We start by identifying differently expressed genes by performing differential gene expression (DGE) analysis. Next, we evaluate the predictive performance of both exonic and intronic read counts using logistic regression. Subsequently, we explore some basic machine learning techniques to combine the information contained in both sets. Furthermore, we construct our own model architectures with the aim of gaining information by using both sets. We show, for this dataset, that exonic and intronic reads have overlapping but also unique differentially expressed genes. Using these genes we show that the predictive performance using the exonic and intronic reads is very similar for all predicted labels. We further show that even though different genes are identified, the biologically relevant signal for the prediction task appears to be the same in exonic and intronic read counts. We are not able to leverage the combination of the counts to further increase predictive performance. Existing disease prediction models have neglected the inclusion of intronic reads. In light of our findings, machine learning models that incorporate intronic reads could potentially discover novel biological insights.

Key words: Transcriptomics, Machine Learning, Gene Expression, Introns

Introduction

Cells are the fundamental units of life, orchestrating a variety of complex functions essential for maintaining homeostasis and facilitating growth and reproduction. These complex cellular functions are made possible by proteins, which play pivotal roles in virtually every cellular process [1]. To produce these proteins, cells transcribe genetic information from deoxyribonucleic acid (DNA) into ribonucleic acid (RNA). This RNA, in turn, serves as a template for the synthesis of proteins. Not every segment of the genomic DNA of an organism encodes proteins, only regions corresponding to genes are transcribed and translated into proteins. Genes are composed of exons, which encode proteins directly, and introns which fulfil a multitude of regulatory functions [2]. After nascent RNA is formed, it undergoes RNA processing to become mature RNA. Among these processes is splicing, in which the intronic parts are removed from the nascent RNA strand, ensuring that only the exonic sequences remain (Figure 1). RNA-sequencing (RNA-seq) is a technique that allows one to measure the RNA content of a single cell (scRNA-seq) [4] or a population of cells (bulk RNA-seq) [5]. By performing RNA-seq, we obtain a digital representation of the RNA transcripts in the cells, called reads. We can then quantify these reads per gene they originate from to get a relative activity of that gene compared to other genes in the cells (Figure 2). In recent years, RNA-seq has gained immense scientific importance, providing detailed insights into gene expression and the specific cellular activities they regulate.

Although RNA-seq can be employed to capture all kinds of RNA molecules, the majority of studies focus on messenger RNA (mRNA). One of the most popular library preparation protocols, poly(A) RNA-seq, captures RNA by targeting the poly(A) tail, which is exclusive to mRNA. As a result, any intronic segments detected are typically considered experimental noise and excluded from subsequent analysis. Another widely used protocol

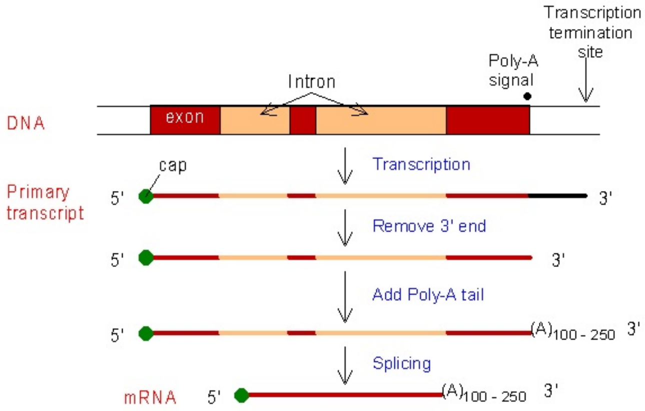


Figure 1: Illustration of general steps involved in RNA-processing. RNA-processing is the process in which nascent RNA matures into messenger RNA (mRNA). During RNA processes, the RNA molecule is capped and a poly(A) tail is added. Sequences originating from intronic regions are typically removed to obtain the mature mRNA molecule. Image from [3].

is total RNA-seq. While this approach does capture nascent RNA, most studies still only consider sequenced reads that align with annotated exons. This practice stems from the prevailing assumption that for protein-coding genes, the majority of RNA obtained from the experiment are mature mRNA transcripts [6]. In spite of this assumption, a large proportion of reads in both total RNA-seq, and even poly(A) RNA-seq, map to intronic regions [6].

There has been a growing interest in investigating the role of intronic reads in RNA-seq experiments. Gaidatzis et al. show that the comparison between exonic and intronic fold change between conditions can separate transcriptional and post-transcriptional regulation [8]. Hereby increasing the information that can be obtained from RNA-seq experiments. Lee et al. show that intronic reads do contain relevant biological signal and they demonstrate differences in the up and down-regulated genes between exonic and intronic read counts [6].

Genomes can contain thousands of genes. Consequently, the expression profiles from RNA-seq data will be very high dimensional. Due to this high-dimensional nature, RNA-seq data is complex and not readily interpretable by humans. Machine learning algorithms are well-suited for dealing with these types of data. Machine learning algorithms can learn to extract relevant biological signal. As a result, RNA-seq has been widely used in research for disease classification [9, 10]. However, since it is common practice to only quantify exonic reads, these models only utilize exonic read counts and disregard the intronic counts.

Numerous cellular processes have been identified that can cause disease when deficit or dysfunctional. Alternative splicing, for example, can be altered in disease [11]. Alternative splicing is directly involved in removing intronic segments from RNA strands, so it is plausible that dysfunction of this process could be reflected in the intronic reads. Indeed, existing research has demonstrated that certain diseases show aberrations that are exclusively manifested in intronic reads. A recent study by Koks et al. [12] found differential expression in intronic reads only, for Parkinson's disease. Another study by Maqueara et al. [13] found an association between retained introns and Alzheimer's disease (AD). Another process, which if dysfunctional can cause disease, is post-transcriptional regulation [14]. The aforementioned study by Gaidatzis et al. [8] shows that from the separate quantification of exonic and intronic read counts can be distinguished if a gene is transcriptional or post-transcriptionally regulated.

Despite these findings, to our knowledge, no studies have looked at the inclusion of intronic read counts for RNA-seq-based prediction tasks.

In this study, we investigate if intronic reads counts contain biologically relevant signal for RNA-seq-based prediction tasks. Furthermore, we investigate if intronic and exonic read counts contain the same information and we explore possibilities to leverage both reads to further increase predictive performance. We start by performing differential gene expression (DGE) analysis to elucidate the difference in differentially expressed genes. With these differentially expressed genes, we evaluate the predictive performances by predicting multiple clinical labels. We explore some basic machine learning techniques to combine the information from both exonic and intronic read counts. Lastly, we explore more complicated model architectures to utilize the potential difference in the information contained within the different read counts to increase predictive performance.

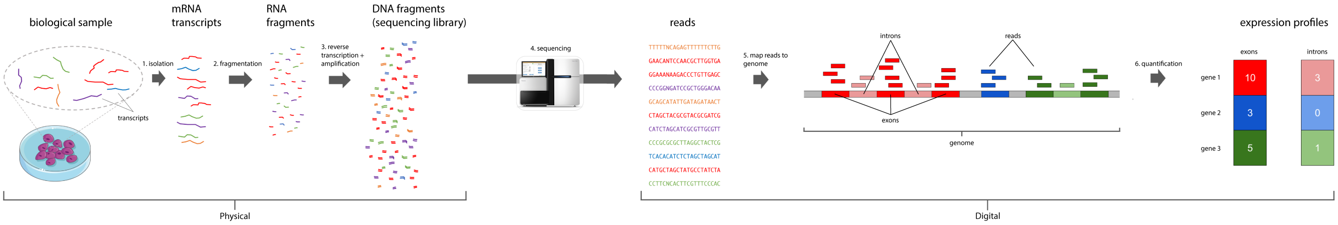


Figure 2: Illustration of the general steps involved in RNA-sequencing (RNA-seq) and consequent processing to obtain expression profiles. The first three steps in this illustration are referred to as library preparation. In this process, the RNA is isolated, cleaned, fragmented and amplified. A finished library preparation is loaded into a sequencing machine, which outputs the RNA transcripts as digital reads. Next, reads are aligned to a reference genome to find the gene they originate from. To obtain an expression profile, alignments for each gene are counted. Traditionally, only the alignments corresponding to exons are counted, producing a single expression profile based on exonic alignments. However, when intronic reads are also considered, we differentiate between the two, resulting in two distinct expression profiles. Image modified from [7].

Methods

Dataset & Preprocessing

Dataset

Brain tissue has a higher intronic read percentage compared to other organs [15] and has a high frequency of alternative splicing [16]. Therefore, we have chosen an RNA-seq dataset sequenced from brain tissue. In this study, we use a dataset from the *Aging, Dementia, and Traumatic Brain Injury (TBI) Project* [17], which represents a subset of the larger Adult Changes in Thought (ACT) study. This particular dataset has been made available by the Allen Institute. The Aging, Dementia, and TBI project is a detailed collection of neuropathologic, molecular and transcriptomic characterization of post-mortem brains. In this study, only the transcriptomic data is used. The dataset can be downloaded from https://portal.brain-map.org/ or at the Gene Expression Omnibus with accession number GSE104687. The dataset contains samples from 107 donors. From each donor, up to four samples originating from four different brain regions were taken. The four brain regions are hippocampus (HIP), parietal cortex (PCx), temporal cortex (TCx) and frontal white matter (FWM). The transcriptomic data was obtained using Illumina TruSeq Stranded Total RNA protocol. The transcriptomic data made available comes in the form of binary alignment map (BAM) files, specifically anonymized BAM files. A BAM file is a binary compression of the sequence alignment map (SAM) format. This format is used to store biological sequence data along with information on where the sequences align against a reference [18]. An anonymized BAM file contains all relevant alignment information but has the read sequences removed. This is not a problem for our study since we are only interested in where the reads align, not the specific nucleotide sequence reads consist of. The alignment process is performed in a multi-step approach. First, *RNA-Seq by Expectation Maximization* (RSEM) [19] was used to align the reads to the transcriptome. Reads that did not map to the transcriptome were aligned to the human hg38 genome using Bowtie [20].

Prediction Labels

In this dataset, roughly half of the donors had dementia, diagnosed by the *Diagnostic and Statistical Manual of Mental Disorders, 4th Edition* (DSM-IV) criteria [21]. The DSM-IV criteria are based on a series of psychometric tests which include verbal, memory, recollection and drawing tests. Furthermore, the donors' brains were post-mortem examined for AD-related pathology. Namely, neurofibrillary tangles and neuritic plaques. Neurofibrillary tangles were classified into the Braak stages [22] and neuritic plaques were assigned a *The Consortium to Establish a Registry for Alzheimer's Disease* (CERAD) score [23]. Based on the Braak stage and CERAD score, donors were assigned a probability of having had AD by the *National Institute on Aging and Reagan Institute* (NIA-Reagan) criteria [24]. These probabilities are high, intermediate, low and no chance of AD. Not all dementia cases could be attributed to AD and not all cognitively unimpaired donors were diagnosed with AD. Thus, the subsets representing donors with dementia and AD are not identical but do overlap. Every donor with cognitive impairment was age-matched with a cognitively unimpaired donor. We focused on multiple prediction tasks to facilitate a more comprehensive interpretation of the results. The labels that were chosen for the prediction tasks are, *sex*, *structure* (brain regions), *dementia* and dichotomized *NIA-Reagan* score. See Table 1 for the specification and proportions of the phenotypes of these labels. The *sex* label distinguishes between male and female. The *structure* label corresponds to the four brain regions (HIP, FWM, PCx, TCx). The positive class of the dichotomized *NIA-Regea* label is a high and intermediate chance of AD and the negative class is a low chance of AD and no AD.

Gene Count Matrices

Quantification of the alignments to obtain the gene expression matrices was performed using the *qCount* function in the *QuasaR* package [25]. We used all default settings except for the minimal mapping quality (MAPQ), *mapqMin*, which we set to ten. A low MAPQ score reflects a high probability that the read is aligned to the wrong position in the genome [26]. The software tool used to align the reads to the reference employs probabilistic alignment. It reports a lot of alignments with a MAPQ score of zero but does not mark them as secondary alignments. Setting *mapqMin* to ten, as is the default in other tools [27], removes all these secondary alignments. Furthermore, alignments marked as secondary, supplemental, or unmapped were not counted. Read alignments were counted for gene bodies, which is the entire gene from the transcription start site to the end of the transcript, and for exons within gene bodies. Subsequently, intronic read counts were obtained by subtracting the exonic read counts from the gene body read counts. The gene annotation provided by the Allen Institute contained 50,267 genes, which collectively included 317,003 exons. Overlapping genes were filtered out to avoid ambiguity. This resulted in a filtering of 10,489 genes. Thus, the final trimmed version of the annotation contained 40.493 genes with 247.929 exons. In the end, we obtained three distinct count matrices: one for exonic read counts only, one for intronic read counts only, and a third with both, which we refer to as the total counts.

Quality Control & Outlier Removal

The Allen Institute notes in the accompanying documents that RNA quality is quite variable between samples. The RNA integrity number (RIN) assesses the integrity of RNA samples by providing a numerical score ranging from one to ten [28]. It measures the degree of degradation of RNA samples. A score of ten denotes highly intact RNA while a score of one signifies almost entirely degraded RNA. Samples with a RIN lower than four were filtered out, this filtered out 18 samples. Upon calculating the ratio as described in section Feature Engineering - Ratio we identified two outliers. In Figure 9 we plotted the ratio of the intronic read count to the total. We see two outliers on the right-hand side of the mean of the histogram. Performing a one-sided Smirnov-Grubss test confirmed *sample 360* and *sample 361* ($p<0.05$) as being outliers. We removed these outliers from all experiments. In the end, we are left with 357 samples from 106 donors.

Count Normalization, Transformation & Scaling

Within-sample normalization is needed to account for technical effects that arise from slight variations during the sequencing protocol [29]. This variation is referred to as a difference in sequencing depth. Count per million (CPM) is a normalization

Label	Category	Number of classes	Class proportions
Sex	binary	2	’M’: 0.59, ’F’: 0.41
Structure	multi-class	4	’TCx’: 0.26, ’HIP’: 0.25, ’FWM’: 0.25, ’PCx’: 0.24
Dementia	binary	2	’No Dementia’: 0.52, ’Dementia’: 0.48
Dichotomized NIA-Reagan	binary	2	1: 0.51, 0: 0.49

Table 1. Labels used in prediction task. HIP: hippocampus, FWM: frontal white matter, PCx: parietal cortex, TCx: temporal cortex, M: Male, F: Female. Dichotomized NIA-Reagan 1: High and intermediate chance of AD, Dichotomized NIA-Reagan 0: low chance of AD and no AD.

method that accounts for library size,

$$\mathrm{CPM}_i = \frac{10^6 \cdot \mathrm{counts}_i}{\sum_j (\mathrm{counts}_j)} \tag{1}$$

where i and j are gene indices within a single sample. Also, library size normalized raw counts can contain a bias stemming from variable gene transcript length [9]. Therefore, normalization by transcript length can be applied. Transcripts per million (TPM) is a normalization method that accounts for both within-sample sequencing depth and variable transcript lengths. TPM per gene is calculated as,

$$\mathrm{TPM}_i = \frac{10^6 \cdot (\text{ counts }_i / \text{ length }_i)}{\sum_j \left(\mathrm{counts}_j / \text{ length }_j\right)} \tag{2}$$

where i and j are gene indices within a single sample. These formulas are suitable in a standard setting where we only have a single library (i.e. exonic counts only). We need to extend these formulas so that they are also applicable in a setting where we have both exonic and intronic counts. Although *Gaidatzis et. al* [8] observe that intronic reads have sufficient coverage to perform library size normalization separately, *Lee et al.* observe that intronic to exonic proportions vary between samples, groups and experimental conditions and therefore could give a poor estimate of the actual sequencing depth [6]. They deviate from the normal approach to normalize over a single count set by taking the library size as the sum of the intronic and exonic counts. Following this approach we modify the TPM & CPM calculation,

$$\mathrm{CPM}_i, \mathrm{exon} = \frac{10^6 \cdot \text{exon counts}_i}{\sum_j (\text{exon counts}_j + \text{intron counts}_j)} \tag{3}$$

$$\mathrm{CPM}_i, \mathrm{intron} = \frac{10^6 \cdot \text{intron counts}_i}{\sum_j (\text{exon counts}_j + \text{intron counts}_j)} \tag{4}$$

$$\mathrm{TPM}_i, exon = \frac{10^6 \cdot (\text{ exon counts }_i / \text{ length }_i)}{\sum_j \left((\text{exon counts}_j + \text{intron counts}_j) / \text{ length }_j\right)} \tag{5}$$

$$\mathrm{TPM}_i, intron = \frac{10^6 \cdot (\text{intron counts }_i / \text{ length }_i)}{\sum_j \left((\text{exon counts}_j + \text{intron counts}_j) / \text{ length }_j\right)} \tag{6}$$

where i and j are gene indices within a single sample. We also report on results for the total counts as more tools start to default on counting both intronic and exonic reads. The total counts just use their own library size and are thus calculated by formula 1 and 2. We investigate whether the choice of normalization by library size or combined library size impacts prediction performance.

Bulk RNA-seq data has very high heteroscedasticity. Small gene counts can range from a dozen to a few hundred while highly expressed genes can have millions of counts. This can pose a problem for machine- or deep-learning algorithms as they cannot cope with these high values. Since we are not necessarily interested in the absolute number but rather a relative change between conditions, we can further transform the data. A common approach is to perform a logarithmic transformation. This transforms the very high counts down to workable numbers while having less influence on already low counts. We transform all the counts as,

$$logcount_i = log2(count_i + 1) \tag{7}$$

where i is a gene index within a sample. We add a single count to every gene to maintain a zero value after transformation for genes with zero counts. Lastly, we scale the counts by applying a z-score transformation. This gives every gene zero mean and unit variance. Although this scaling is not required for our logistic regression model, described in section Models. It is beneficial for gradient descent convergence, improves the comparability of model performance, and allows us to interpret the values of learned weights as feature importances.

Differential Gene Expression Analysis Gene Selection

Typically, RNA-seq data exhibits high dimensionality. Our dataset comprises more than 40,000 genes. Machine learning models are prone to overfitting when dealing with high dimensionality, particularly when combined with a limited sample size. This is commonly referred to as the curse of dimensionality [30]. Some machine learning methods can apply regularization to deal with this but this is not always sufficient. Furthermore, computational resources become a limitation when dealing with this high dimensionality. Fitting one model is not directly a problem. But in this study, we compare multiple models using multiple datasets (i.e. exon and intron) across multiple labels. In combination with our test split and cross-validation, described in section *Test, Train, Validation Split,* fitting all these models becomes infeasible for the scope of this study. To deal with the aforementioned problems, feature selection by DGE analysis is a common approach [31, 32], operating on the assumption that not all genes will be informative for the prediction task. DGE analysis identifies genes that are expressed at different levels between experimental conditions or groups. This is achieved using statistical methods that compare the abundance of RNA transcripts between these groups. A drawback of this approach is that the genes are fitted in a univariate fashion. Thus, this will potentially exclude genes that are important in combination

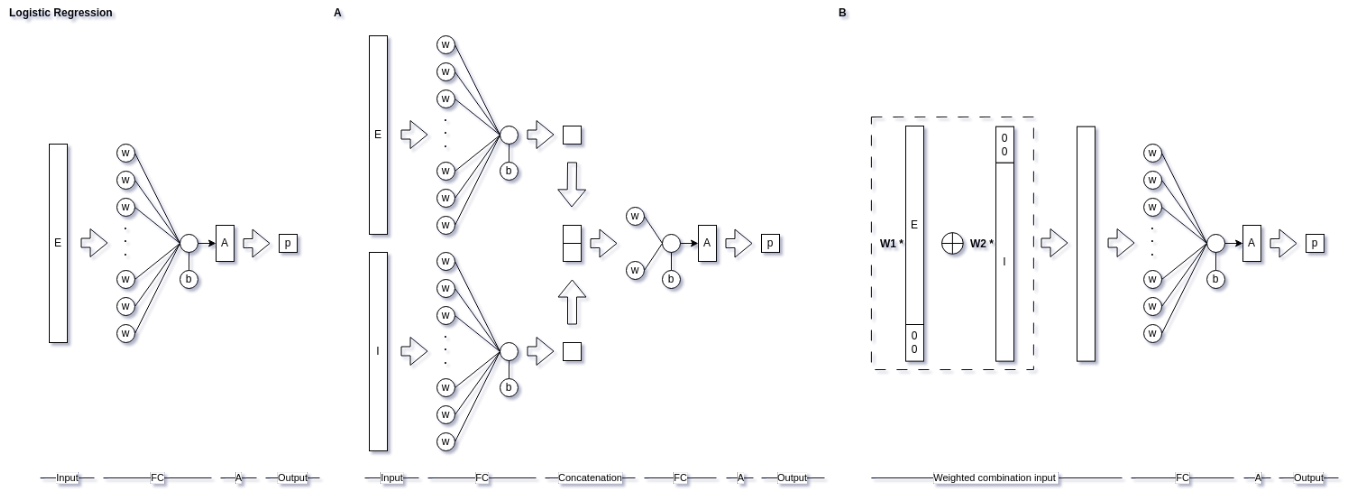


Figure 3: Diagram of logistic regression and our custom linear model implementations in PyTorch. Model A: end-to-end assemble approach. Model B: weighted combination of the exonic and intronic gene counts approach. FC: Fully connected layer, also called linear layer. W: learnable weight in fully connected layer. b: learnable bias weight. p: output, single probability for binomial prediction and probability vector for multinomial. w1 and w2: single learnable weights. 0: zero-padded input genes. A: activation function, sigmoid for binomial prediction and softmax for multinomial.

with other genes. The DGE analysis is performed using the well-established Limma-Voom [33] pipeline. This pipeline consists of three steps. First, the *edgeR::filterByExpr* function is used to filter out genes with low counts. Next, *limma::voom* is applied to transform the count data. Lastly, *limma::lmFit* and *limma::eBayes* are applied to fit the linear models and apply empirical Bayes moderation to increase the stability. *Limma* uses False Discovery Rate (FDR) adjustment to correct the p-values. This pipeline is implemented by modifying the tool created by Lee et al. [6], which they call **in**tron **d**ifferences to **ex**on (*index*). *Index* only handles exonic and intronic read counts, we extended the tool to also be able to deal with the total counts. Note that the Limma-Voom pipeline uses its own transformation and normalization, we thus supply the raw read counts as input for this analysis. We keep the DGE analysis simple, so we don't include any covariates or blocking factors. In initial DGE analyses, we observed high variability in the number of genes found between the different train-test splits (Supplemental Table 1). A standard approach for DGE analysis-based feature selection is to select genes with adjusted p-vales less than 0.05. We deviate from this approach for two reasons. First, the observed high variability between the folds compromises comparability. Second, after taking the intersection of the three validation folds we no longer find any genes for our *NIA-Reagan* label in some test folds. Since the goal is to select informative genes rather than report on significance we select the 1000 most differentially expressed genes (i.e. smallest adjusted p-value) per cross-validation fold and take the intersection over these folds. This approach has a lower coefficient of variation between the folds, which is better for comparability and stability across the folds (Supplemental Table 1).

Models

All models are implemented using the *PyTorch* framework [34, 35] in Python. Pytorch models are fitted using gradient descent. As our optimizer, we chose the commonly used *Adam* optimizer. To prevent overfitting and to add a level of constraint to the complexity of the models, L2 regularization was applied using weight decay. Weight decay is not mathematically exactly the same as L2 regularization but has the same effect, it helps penalise large coefficients, thereby ensuring that the model does not become overly complex.

Logistic Regression

We start by investigating the difference in predictive performance by applying logistic regression. Logistic regression falls under the class of generalized linear models and is used for modelling binary outcome variables. The model consists of one fully connected layer coupled with a Binary Cross Entropy with Logistic Loss (*BCEWithLogitsLoss*). For multinomial logistic regression, where the dependent variable can have more than two categories, we modified the model by replacing the sigmoid with a softmax function over the output layer. This ensures that the probabilities sum to one, providing a valid probability distribution over the multiple classes. Correspondingly, we changed the loss function to cross-entropy loss (*CrossEntropyLoss*), which is suitable for handling multiple categories (Figure 3).

Custom Linear Model Architecture

In the context of combining exonic and intronic counts, we extend our exploration beyond straightforward machine-learning approaches. PyTorch's backpropagation mechanism allows complicated model architectures. More complicated models do come with an inherent risk of overfitting. With an increased number of parameters, the models may fit the training set better or precisely but fail to generalize as effectively as simpler models. This is especially a risk for datasets with small sample sizes. We explore two custom linear model architectures, visually depicted in Figure 3. Given this risk of overfitting, we wanted to test model architectures that remain relatively simple and have some inherent motivation behind them. Our first model, A, can be conceptualized as training an ensemble, but instead of training two separate models, this model is trained end-to-end. This model does have more learnable parameters than the concatenation of the datasets, however, the addition of the last layer creates an information bottleneck. Thereby potentially limiting the overfitting ability. In our second model architecture, B, we take a weighted summation of our input gene counts and use this as input to our logistic regression module. The model can learn the optimal weights to sum the intronic and exonic read counts. Possibly outperforming the logistic regression model using the total counts, where the counts are essentially the exonic and intronic counts summed with weights one. The number of parameters is drastically less in this model compared to the concatenation of the counts. Mathematically these models can be written as, A,

$$F(E, I) = g(((EW_1 + w_{10})^\frown(IW_2 + w_{20}))W_3 + w_{30}) \tag{8}$$

And B,

$$F(E, I) = g((w_1 * E + w_2 * I)W_3 + w_{30}) \tag{9}$$

Here E and I denote the input feature, i.e. the exon and intron counts, capital W_x denotes a weights matrix and lower case w_x denotes a single weight, and $^\frown$ is the concatenation operator. In the binomial case, $g(x)$ denotes a sigmoid function providing a probability as output, which can be rounded to obtain the class prediction. For the multinomial case, $g(x)$ denotes a softmax function where the output is an n-dimensional array of probabilities. Here the class prediction is obtained by taking the index of the highest probability. All input count sets that are summed in model B need to be of the same size. Since we find a different number of genes for exonic and intronic counts we zero-pad the genes that are not selected. In this manner, we can still make a valid comparison to other models, since the input information is still the same. We train two extensions of model A. One which also incorporates the ratios and one where the ratios and the total information are incorporated (A-EIR, A-EIRT). For the second model, B, there is no inherent motivation to sum rations with the counts so we only do a weighted combination of exonic and intronic read counts (B-EI).

Non-Linear Models

Biological systems are known to exhibit non-linear characteristics [36, 37]. We investigate if non-linear models are able to leverage the combined information to increase the performance compared to the linear models. We modify the models from the previous section to include a non-linear activation function at every operation. Specifically, we employ the Rectified Linear Unit (ReLU) activation function. This results in the following mathematically adaption from Formula 8 and 9,

$$F(E, I) = g((\sigma(EW_1 + w_{10})^\frown\sigma(IW_2 + w_{20}))W_3 + w_{30}) \tag{10}$$

And B,

$$F(E, I) = g((\sigma(w_1 * E) + \sigma(w_2 * I))W_3 + w_{30}) \tag{11}$$

where σ denotes the ReLu activation function. We also apply this to the extension of A where the ratios and the total counts are included (act-A-EIR, act-A-EIRT).

Model Ensembles

Ensemble approaches can outperform a single model by leveraging the strengths of multiple individual models. Besides increasing the predictive performance, ensembles can also enhance the robustness of the predictions. We employ three simple ensemble approaches where we combine our logistic regression models for exonic, intronic and total counts as well as the count ratios. Our logistic regression models predict the probability of the input belonging to a certain class. We combine these probabilities to make an average confidence ensemble and a max confidence ensemble. Furthermore, we make a majority voting ensemble. Here the probabilities are first converted to a class and subsequently, we take the class with the highest occurrence among the individual models within the ensemble. In case of a tie, we pick at random between the tied classes.

Experimental Setup

Test Split & Cross-Validation

A significant level of variability was observed depending on the chosen train-test split during experiments. Since the objective was not to develop a model but rather to investigate the differences, a test fold split approach was employed. The original dataset was divided into five folds. Subsequently, within each training fold, three-fold cross-validation was conducted for hyperparameter optimization and DGE analysis. For every train-test split, we perform our DGE analysis thrice, once on each train part of the 3-fold cross-validation. Subsequently, we take the intersection of the genes identified by these three DGE analyses as our final gene selection. This prevents information leakage between the train and test set since the DGE analysis never sees the test data. Furthermore, by taking the intersection over the three folds we prevent information leakage to the validation set. The mean and variance needed for z-score normalization are calculated from the train set only and subsequently used to scale the validation and test sets. Each train/test fold was fully independent, resulting in different sets of hyperparameters and different sets of genes from the DGE analysis per train-test fold. This workflow is illustrated in Figure 4. Both the train-test and cross-validation splits are performed by randomly picking donors and selecting the samples from these donors. Consequently, no donor can have samples in multiple folds. Hereby preventing the models from learning donor-specific information during training. Given the low number of samples, we stratified for all relevant labels (*structure*, *sex*, *dementia*, and *NIA-Reagan*) to increase the stability of the results. For the final evaluation on the test fold, we still need a validation set to determine when to stop training. The split here also influences the outcome. We fit three models with each of the three train-validation splits and subsequently test these models on the corresponding test fold.

Hyperparameter Tuning

Three critical hyperparameters influence our models' ability to learn. Those are learning rate, batch size, and weight decay (L2 regularization). These parameters need to be optimized to find the optimal performance. To accomplish this we employ Optuna [38] as our optimization framework. With the default settings, Optuna uses a Tree-structured Parzen estimator for parameter selection. This is a surrogate model approach where Gaussian Mixture Models are fitted to find the best parameters. This approach allows us to dynamically search a large search space (Supplemental

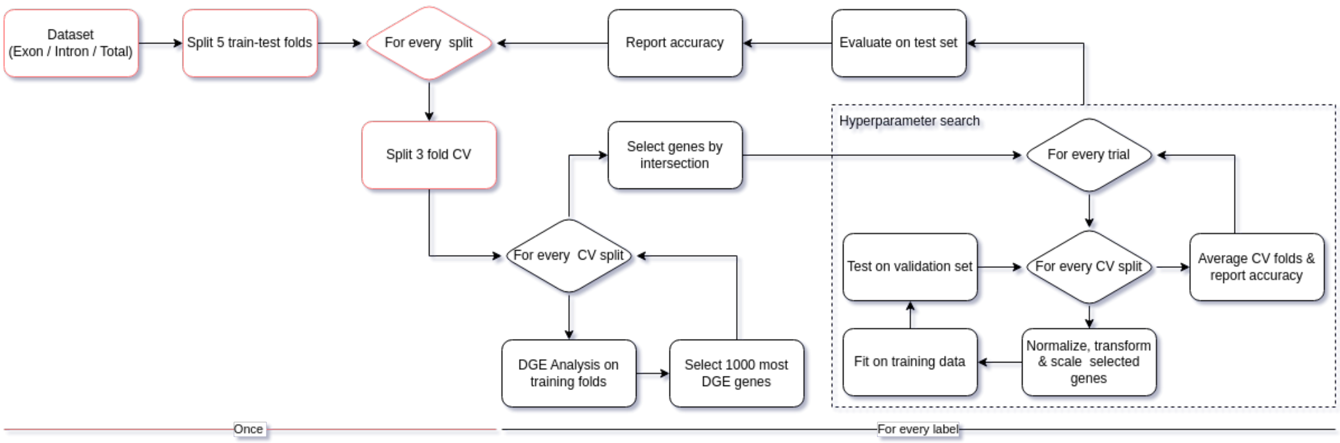


Figure 4: Diagram of the workflow to evaluate the performance of our models.

Table 2) instead of relying on fixed values in a traditional grid search. The specific parameter we aim to optimize is the validation accuracy averaged over the three validation folds. Because of the large search space and the random initialization of the surrogate model, we search until there is no improvement for 175 consecutive trials. We observed that with this value all the models' improvements have plateaued. A common practice when fitting a model with gradient descent in PyTorch is to let the model run for sufficient epochs and select the model with the highest validation performance. We slightly deviate from this approach by taking a rolling average of the validation accuracy. We observed extremely erratic learning curves for some hyperparameter combinations. Those models would obtain the highest performance, but the accuracy could jump over 30 accuracy points per epoch. We argue that the models are not learning anything, but by luck, adjusting their weights to obtain high performance. By taking the rolling average, we enforce a parameter combination with which the model is genuinely able to learn the decision boundary rather than exhibiting this erratic behaviour.

Evaluation Metrics

Our models are fitted using gradient descent with cross-entropy loss. However, a minimal loss does not necessarily correspond to maximal accuracy. Since we have a classification task, we are interested in correctly predicted samples rather than minimizing a loss measure. Therefore we optimize for and report on accuracy instead of loss. Note that given our relatively small dataset of only approximately 71 samples per test fold, a single sample contributes about 1.4% when predicted correctly. We evaluate if the obtained accuracies of the models are significantly different ($p<0.05$) from each other, by performing a two-sided Wilcoxon rank-sum test. Which is an unpaired non-parametric test to compare if two samples follow the same distribution. A more standard approach would be to use a t-test, however, not all our models' accuracies were normally distributed.

Feature Engineering - Ratio

The intronic ratio is interesting to investigate for a couple of reasons. Firstly, an exon count-based machine learning model would utilise a relative difference in counts to classify a condition. However, a relative increase in both exon and intron will not result in a different ratio and can therefore not be picked up in the same way. A relative ratio difference can come from six possible changes. Only up or down-regulated exonic reads, only up or down-regulated intronic reads or when both are oppositely regulated, that is, exon up and intron down or the other way around. Another interesting aspect is that the ratios are normalization-independent as the divide will cancel out the normalization. Furthermore, taking the ratio scales all the values between zero and one. This scaling makes the ratios less sensitive to outliers. All in all, it is worth investigating if the ratios can maintain the same signal and its relative performance to the exon and intron counts. We calculate the ratio as the intronic part of the total counts.

$$\text{ratio}_i = \frac{\text{intron count}_i}{\text{intron count}_i + \text{exon count}_i} \tag{12}$$

For every gene i. The ratio is calculated using the raw counts without performing library normalization and logarithmic transformation but the ratios are z-score normalized. The *index* pipeline used for the exonic, intronic and total counts is incompatible with ratio values. Consequently, we substituted this part with a two-sided Wilcoxon rank-sum test. The p-values from the Wilcoxon rank-sum test were adjusted for multiple testing using the Benjamini-Hochberg false discovery procedure. As is used in the *limma-voom* pipeline. The other steps in our workflow are exactly the same for the ratios as for exonic, intronic and total counts.

Feature Space Concatenation

The most straightforward way to utilize both the exonic and intronic read counts is to supply both the count matrices as input features to a machine learning model. If one of the two contains different information we can potentially leverage that to increase the model performance. However, increasing the feature space also means that the model is more susceptible to overfitting. Another concern is multicollinearity. Logistic regression is known to suffer from multicollinearity [39]. If we select the same genes for different read counts during the DGE analysis we have a chance that these will be highly correlated. We explore three different concatenation options, namely exonic and intronic read counts concatenation (concat-EI). Secondly, exonic, intronic read counts and ratios concatenation (concat-EIR). Lastly, exonic, intronic, total read

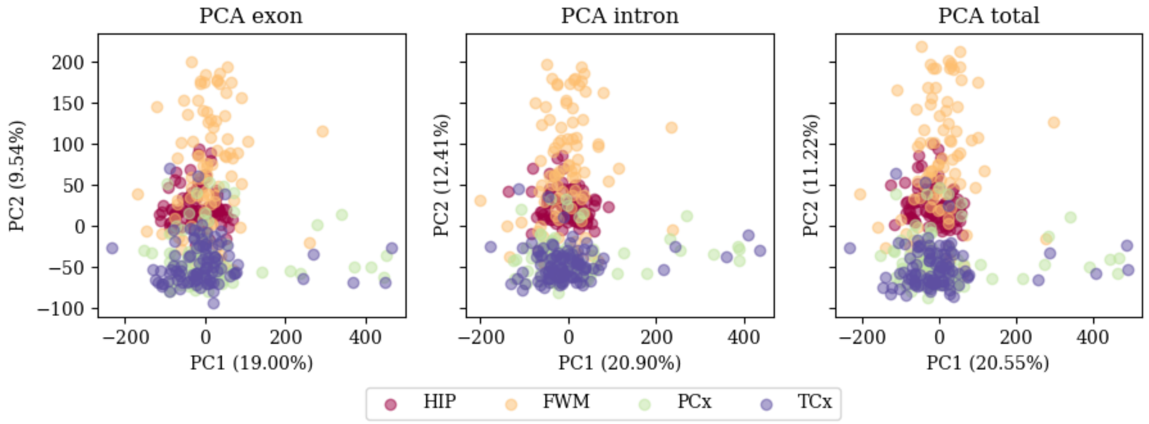


Figure 5: Scatter plot of the first two principal components of the PCA transformed data, for exon, intron and the total counts. Coloured by brain region, HIP: Hippocampus, FWM: frontal white matter, PCx: parietal cortex, TCx: temporal cortex

counts and ratio concatenation (concat-EIRT). Adding the total counts when the genes are the same does not provide additional information to the model, only if we select (partially) different genes during the DGE analysis.

Principal Component Analysis

Principal Component Analysis (PCA) is a transformation technique often used to reduce the dimensionality of datasets while retaining as much variance as possible. We employ it to explore our high-dimensional data in two dimensions. Furthermore, we utilize PCA to mitigate the effect of multicollinearity in our dataset, as PCA produces inherently uncorrelated principal components [40]. For visualization purposes, we transform our entire dataset at once. When using it in our models to mitigate multicollinearity we first transform our training set and subsequently use the found principal components to transform our validation and test sets. This prevents information leakage between the sets. For both use cases, we first apply TPM normalization and z-score normalization prior to PCA. Note that we only use the PCA transformation in our models in section *Reducing Multicollinearity by PCA transformation Does Not Provide Performance Improvement*. We use the *scikit-learn* [41] Python library to perform the PCA transformation. In our models, we use the maximum number of principal components possible, which is defined by $min(n_{samples}, n_{features})$. Depending on the number of genes found during DGE analysis we are thus limited by the number of genes selected or the number of training samples.

Code Availability

All software packages and their version numbers are listed in Supplemental Table 3. The code for all experiments is available at https://gitlab.ewi.tudelft.nl/goncalveslab/master-projects/msc-thesis-2223-thomas-zuiker/.

Results & Discussion

PCA and Correlation Analysis Reveal Potential Information Difference Between Exonic and Intronic Read Counts

We begin by inspecting the data to verify that intronic read counts contain relevant biological signals and investigate to what extent the information is different from the exonic or the total read counts. We commenced by performing PCA transformation on the whole dataset for exonic, intronic and total reads separately, and visualized the first two principal components (Figure 5). An immediate observation is that all three plots are very similar and the first two components seem to separate the brain regions. This verifies the findings by Lee et al. [6] that intronic reads contain relevant biological signal. While the first two principal components plots seem very similar, there are differences noticeable when looking at higher principal components (Supplemental Figure 1, 2 and 3). Notably, the ten first principal components of the intronic read all demonstrate greater explained variability compared to the exonic and total read counts, 55.66, 46.12% and 52.22% respectively. Considering that the PCA plots still look relatively the same, this could indicate less noise being present in the intronic reads or that they simply contain less information.

Because the overall PCA results appear very similar, we calculated the Pearson correlation coefficient (ρ) between each gene's exonic and intronic counts (Figure 6). We observe a large proportion of the genes having a correlation close to one (24.7%

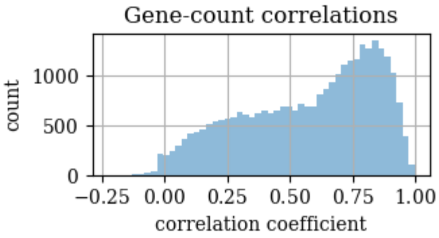


Figure 6: Histogram of Pearson correlation coefficients of correlation between exonic and intronic read counts, for all genes.

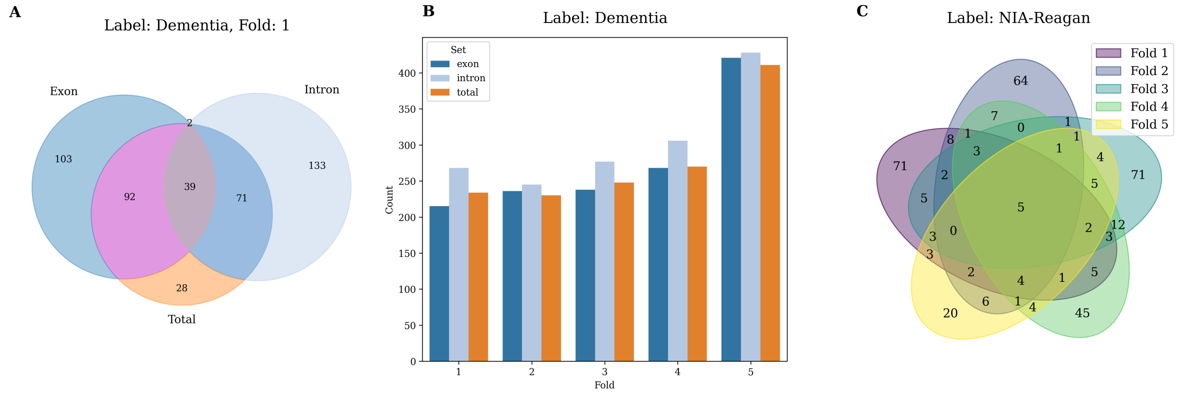


Figure 7: Visualization of genes identified by differential gene expression (DGE) analysis. Genes are selected by taking the 1000 most differentially expressed genes per cross-validation fold and subsequently taking the intersection of the cross-validation folds (Section Differential Gene Expression Analysis Gene Selection). The p-values are corrected for multiple testing using the False Discovery Rate (FDR) procedure. **A:** Venn diagram of the overlap between the DGE genes found for *dementia* in fold 1. **B:** Histogram of the number of DGE genes found per fold, grouped by the count set. **C:** Venn diagram of the overlap between the exonic DGE genes for different folds, for label *NIA-Reagan*. Note that these areas of the folds are not drawn to scale.

$\rho > 0.75$). However, there is also a very substantial proportion that is not close to one. Moreover, there are even a lot of genes with a correlation close to zero (11.1% $\rho < 0.25$). Having all correlations close to one would be a strong indicator that the information contained within the count sets is largely the same. Given that this is not the case, this offers motivation to investigate the differences within a machine-learning context.

Differential Gene Expression Analysis Reveals Unique Genes in Exonic, Intronic, and Total Counts

Similar to Lee, et al. [6], we find uniquely differential expressed genes from our exonic read counts as well as intronic read counts (Figure 7A). Besides the uniquely found genes, we also observe a substantial overlap between the two sets. This observation holds for all our labels across all our test folds (Supplemental Figure 4). Furthermore, our extension to also perform DGE analysis on the total counts reveals that this set also contains differentially expressed genes which are not found in either exonic counts or intronic counts (Figure 7A). Lee et al. report genes with opposite fold change (i.e. exonic upregulated and intronic downregulated or vice versa), but we don't observe a single gene exhibiting this behaviour in our DGE selected genes. We also don't observe this for the differentially expressed genes from the total counts with either exonic counts or intronic counts.

We observe a relatively large variability in the number of genes found across test folds. For example, we see for *dementia* almost double the number of genes found in fold five compared to the other folds, across all the count sets (Figure 7B). Only *structure* shows low variation, all other labels show high variation (Supplemental Figure 6). This could indicate high heterogeneity in the dataset, as different subgroups apparently have different expressions. The difference in genes identified per count set (Figure 7A) could be attributed to the variation between the test folds. To this end, we also inspect the genes obtained by taking the intersection over all the test folds. For all labels, the aforementioned observation still holds for these genes, providing more confidence that there are uniquely expressed genes present in the different count sets (Supplemental Figure 5).

Another observation is that the quantity of genes identified fluctuates considerably between different labels. For example, *structure* encompasses close to 800 genes while *NIA-Reagan* has roughly 100 (Supplemental Figure 6). This disparity provides insight into the extent to which informative genes are present within the data. The high number of genes found in *structure* is expected since we saw a strong signal in the PCA plot (Figure 5). However, *NIA-Reagan* finds only around a hundred genes while we take the intersection of the thousand most differentially expressed genes.

DGE analyses are frequently conducted to report significant genes in diseases or between conditions. We discern that our DGE analysis exhibits significant differences between genes found depending on sample inclusion or exclusion (i.e. the difference between the test folds). It becomes evident that careful interpretation is paramount when analyzing and reporting on DGE analyses, especially in datasets exhibiting variability akin to ours. Furthermore, the differences in genes found between the exonic, intronic and total counts are seldom considered in the literature, underscoring the necessity for careful interpretation of existing literature even more.

In light of our interest in a machine learning context, some of these observations warrant our attention. The observed high variability is not necessarily detrimental. A machine learning model can possess the capability to extract meaningful features while disregarding irrelevant ones. However, high heterogeneity in the data can potentially compromise the model's generalizability and stability. Furthermore, the little overlap between the genes found across the folds for *NIA-Reagan* and *dementia* could pose a problem (Figure 7C, Supplemental Figure 7). For every fold, we observe that the majority of the genes are unique to that fold. This suggests that the identified genes might not be genuinely informative for the prediction task and will not generalize to the test set. This is also supported by the observation in Supplemental Table 1 where we see that no genes are found for *NIA-Reagan* with an adjusted p-value less than 0.05 for some folds. On the other

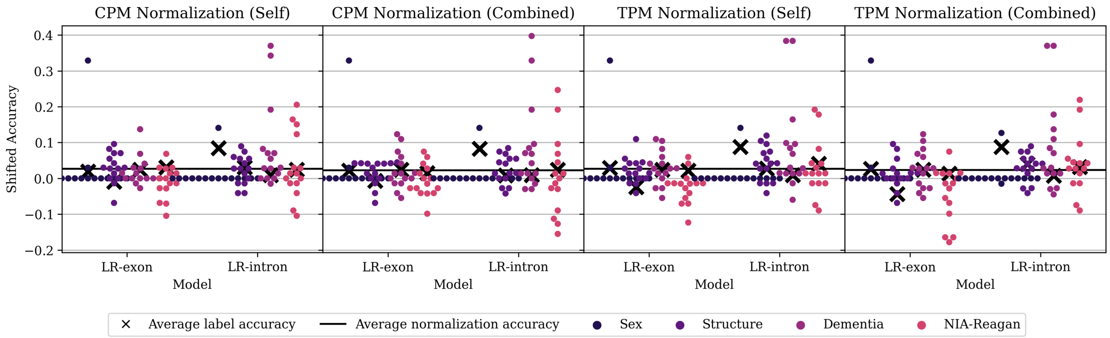


Figure 8: Model performance under different normalization techniques. The reported accuracies are shifted by the accuracy of no normalization.

Metric	percentage
Single exon genes	36.83%
Ratio of 1	5.60%
Ratio of 0	39.10%
Ratio 0<x<1	55.30%
Ratio 0 without SE genes	2.27%

Table 2. Table of characteristics of the count ratios.

hand, our observation that there are genes found unique to the exonic, intronic or total counts (Figure 7A) does provide further motivation to investigate if these found genes contain the same or different signal for a machine learning prediction task. The full results of the analysis are illustrated in Supplemental Figures 4, 6, and 7, and Supplemental Table 1.

Statistical Analysis of the Ratios Also Identifies Unique Genes

We analyzed the general properties of the computed ratios (Table 2). Notably, our dataset contains 14751 single exon genes (SE). SE genes are characterized by the absence of intronic regions, resulting in a ratio of zero. Excluding single-exon genes, the percentage of zero ratios is very small. We further inspected the obtained ratios by plotting the per-gene average and the per-sample average, see Figure 9 A and B. We see that the per-sample averages are normally distributed around 0.35. The gene averages, on the other hand, show an almost uniform distribution from zero to one, albeit with one spike at zero attributed to the SE genes. Our DGE analysis does not select a SE gene for any label or test fold. If we exclude the SE genes the intronic average becomes 0.43.

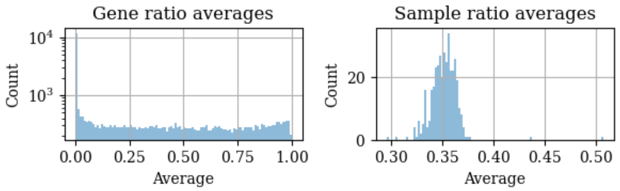


Figure 9: A: histogram of the ratios averaged by per gene. B: histogram of the ratios averaged per sample.

Interestingly, there is also a considerable number of occurrences where the ratio is one, implying an absence of exonic read counts. We included the ratio genes found by our statistical tests in Supplemental Figures 4, 6, and 7. We find a very similar number of genes compared to exonic, intronic and total counts for every label and every fold (Supplemental Figure 6). Even the number of genes that overlap across the folds is very similar (Supplemental Figure 7). Therefore, a surprising observation is that the actual genes found, overlap very little with the three other sets. See for example *dementia* test fold 1, there is almost no overlap between the ratio and the other three sets (Supplement Figure 4). To some extent, this is expected. Given that we don't observe any opposite fold change, any gene that is both differentially expressed in the exonic and intronic reads will only exhibit a slight to no change in the ratio. Consequently, these genes will not be significantly different between conditions when performing our statistical tests. Nonetheless, it is still striking to see that the number of found genes is so similar but the actual genes are not.

Model Performance Is Comparable Between Library Size Normalization and Combined Size Normalization

We investigated if combined library size (exonic and intronic counts) normalization would outperform own library size normalization (only exonic or intronic counts). We trained our models using two normalization methods (CPM, TPM) using both library size options. This resulted in four different ways of normalizing. We also evaluated the performance of no normalization. We shifted the results of the obtained performance by subtracting the performance of no normalization, to better visualise the difference. The first clear observation is the relatively high variability in the accuracies within a label across the folds (Figure 8). We see some clear improvements but also some occurrences where the performance is worse than no normalization. We attribute this to two factors. First, the dataset is relatively small thus a single flip of a prediction already has quite a significant influence on the results. Furthermore, the models are also somewhat sensitive to random initiation. The hyperparameter search is randomly initiated as well as the weights upon fitting the final model. Combined these two factors can make it so that in some cases the normalized version underperforms compared to the unnormalized ones. As far as our interest goes in a performance difference between self and combined normalization, we can not observe a substantial difference. This is not unexpected as brain

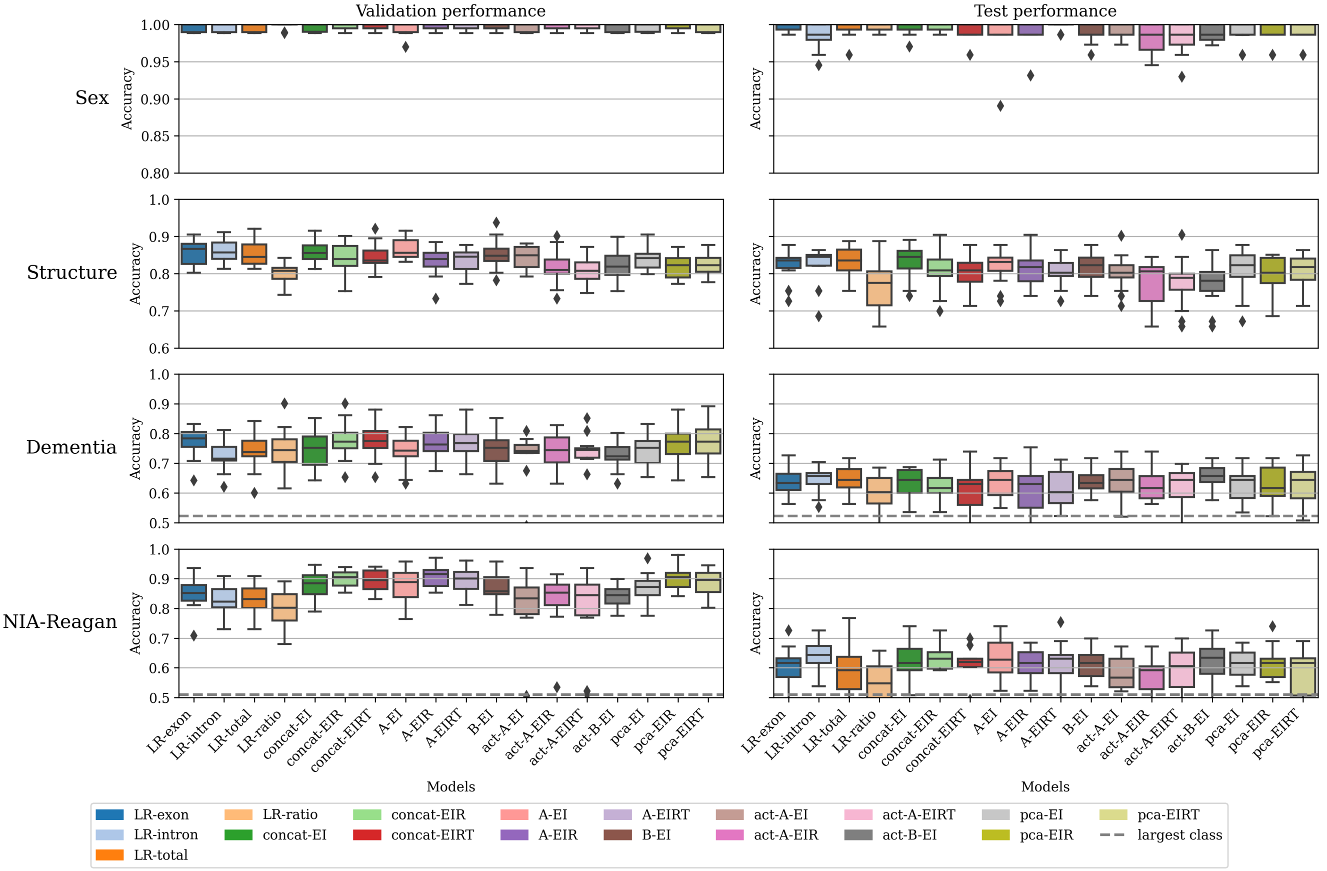


Figure 10: Boxplots of model performance using different model architectures and input data. The first column depicts validation performance. The second column depicts the test performance. The rows correspond to the prediction labels. The largest class of a label is indicated by the grey line, which indicates a guessing performance.

regions have a high percentage of intronic read counts to the total. We observed this when looking at the per-sample averages, where the mean is roughly 35% (Figure 9B). Other organs exhibit a lower intronic to total percentage, as found by Lee et al. [6]. Their mean of intronic to total reads in total RNA-seq is 21%. There the choice of normalization might be more pronounced. Thus, for further research, we suggest investigating if our observed indifference also holds for samples from a lower intronic percentage region. Since we observe no substantial difference we use TPM combined library size normalization for our experiments, as TPM should theoretically contain less bias than the CPM normalization.

Exonic, Intronic and Total Counts Show Similar Performances

To investigate the difference in performance between the different count sets, we trained logistic regression models to predict different labels. In Figure 10 we plotted the results of the validation and test performance of our models using exonic counts (LR-exon), intronic counts (LR-intron) and total counts (LR-total). From the results, we can make observations regarding trends and patterns of the general performance and we can make observations regarding the difference between the different inputs of the three aforementioned models. Firstly, regarding general performance. We see for *sex* nearly perfect performance, *structure* also has good performance but *dementia* and *NIA-Reagan* perform just slightly better than learning to classify everything as the largest class, or guessing. Notably, for every label except *sex* we see quite a high variance between the folds for both validation and test performance. Furthermore, we see for *dementia* and especially *NIA-Regan* a large discrepancy between the validation performance and the test performance. Our hypothesis that the *NIA-Reagan* would not generalize well defined in the DGE analysis section partly holds. We see very good validation performance that does not generalize at all for some folds. However, for other folds, the model is able to learn beyond just learning the largest class. This can be explained by our DGE analysis observation. We saw almost no overlap between the genes found in the different folds. From this, we assume that the differences that get picked up during DGE analysis are differences specific to the validation set. As we take the intersection between the cross-validation folds we preselect on genes that are coincidentally donor-specific to the whole train set. This results in excellent validation performance and poor test performance. Now we consider the relative difference between the three aforementioned models. An immediate observation is that intron, exon and total all seem to perform very similarly. Once again confirming the finding by Lee et al. [6] that the intronic reads contain relevant biological signal. Hereby further substantiating the invalidity of the assumption that intronic reads result from experimental noise. Since we are only comparing fifteen accuracy scores (3 fits times 5 folds) per model per label, and considering the random initialization, with these high variability between the folds we cannot with confidence conclude that one outperforms the other. *Dementia* and *NIA-Reagan* both have a higher validation performance using exonic reads, but for test performance, the intronic reads perform better. For *sex* intronic reads seem to perform slightly worse than exonic reads. For *structure*, both are very similar. Testing for significance confirms that none of the intronic and total count models are significantly different from the exonic models for all labels (Supplemental Table 4).

Ratio Performance Holds Up To Exon, Intron and Total Counts Model Performance

Considering that we select very different genes for the ratios compared to DGE analysis of exonic, intronic and total counts, it is interesting to see if the signal for the prediction task is also present in the ratios. The performance of the logistic regression models using the ratios is plotted in Figure 10 (LR-ratio). For *NIA-Reagan* and *structure* we see a clear drop in validation performance, which is not the case for the validation performance of *dementia* and *sex*. The test performance is substantially impaired for *NIA-Reagan* and *structure*, less so for *dementia* and not at all for *sex*. Testing for significance confirms that *NIA-Reagan* and *structure* are significantly worse than the exonic read count performance (Supplemental Table 4). However, *sex* and *dementia* are not significantly different. Despite totally different genes being used to make the prediction we still observe that the model is able to learn. However, our suggestion that the ratios might suffer less from outliers and that they are normalization-independent does not translate to improved performance.

Concatenting Count Sets Does Not Improve Performance

We established that model performance is comparable between the exonic, intron, and total read counts and somewhat to the ratios. However, we also observed different genes are being selected from the DGE analysis. The models could be learning different decision boundaries, hereby classifying different samples as correct or wrong. We do observe differences in correctly predicted samples (Supplemental Figure 8, 9). Although subtle, this difference could potentially be exploited to increase the performance above an individual model's performance. The most straightforward approach to potentially exploit different information contained within the different datasets is to simply concatenate the features, in our case transformed gene counts. The results from our concatenation models are depicted in Figure 10 (concat-EI, concat-EIR, concat-EIRT). We observe that for *NIA-Reagan* we improve the average validation performance to around 0.9. However, this increase does not translate to improved test performance. We do see a substantial reduction in the test variance of concat-EIRT. The validation performance for *dementia* is very similar to the exonic count performance. The test score has slightly deteriorated for all concatenation models. For *structure* we observe a deterioration in the test performance and *sex* is very similar to previously obtained scores, for both validation and test performance. None of the models are able to realize an average accuracy increase over the best base model, and none of the results are significantly different from the base models (Supplemental Table 5, 6, 7 and 8). Considering all these observations we conclude that, for our dataset, there is no clear improvement on the performance made by concatenating the counts or ratios in this way. We inspect the feature importances to assess the contribution of the different input sets in the concatenation models (Supplemental Figure 10). We observe that overall the models do not disregard any of the sets by fully focusing on one, instead we generally see the feature importance are relatively evenly divided over the different input sets. Except when predicting *structure*. The ratios seem to be much more important both in LR-EIR and LR-EIRT. This observation is surprising as we saw that only ratio as input performs significantly worse. It is in agreement with the observation that concat-EIR and concat-EIRT both have reduced test performance.

Custom Linear Models Do Not Resolve Overfitting

We observe that no features are disregarded in the concatenation models (Supplemental Table 10). In the absence of an improvement, the models could suffer from the increased feature space. To this end, we try our custom linear models. Much like the concatenation we generally observe an increase in the validation performance for *NIA-Reagan* and *Dementia* but the test performance does not seem to improve over previous models (Figure 10). We observe the more features we add the more the performance is deteriorated (i.e. LR-EIRT is worse than LR-EIR etc.). We also observe this dropped performance in the test set for *structure*. *Sex* does see a slight test improvement for A-EIRT but this is not significant (Supplemental Table 5). Again, none of the results are significantly different from the base models (Supplemental Table 5, 6, 7 and 8).

Non-linear Architecture Does Not Provide A Benefit

The addition of a non-linear activation function enables the model to learn non-linear relationships in the data. Across all labels, except *sex* we observe that applying activation functions to model A causes a drop in validation performance compared to the model without activation function. We do not observe a clear improvement in either stability or predictive performance in the test set across all models and all labels. However, the average accuracy for *dementia* does increase from 64.8% for LR-intron to 65.4% for act-B-EI, however, this is not significant (Supplemental Table 7). Considering the variation between the folds this increase is arguably negligible. Moreover, this model is not the best performing on the validation set. In a practical setting, this model would therefore not have been selected as the final model.

Reducing Multicollinearity by PCA transformation Does Not Provide Performance Improvement

When we inspect the per-gene exonic-intronic correlation of the selected genes we generally observe a high correlation (Supplemental Figure 11). The drop in performance observed in our custom models could be due to the increased correlation between the input features. Using uncorrelated principal components reduces multi-collinearity. However, by transforming our validation and test set with the principal components calculated from the training set we assume that taking the linear combination in this way also captures the variation of the validation and test set. This poses a potential drawback for data with high heterogeneity. We trained logistic regression models with the exonic and intronic data combinedly transformed (pca-EI), exonic intronic and ratios combinedly transformed (pca-EIR) and exonic, intronic ratios and total counts combinedly transformed (pca-EIRT) (Figure 10). The performance for *sex*

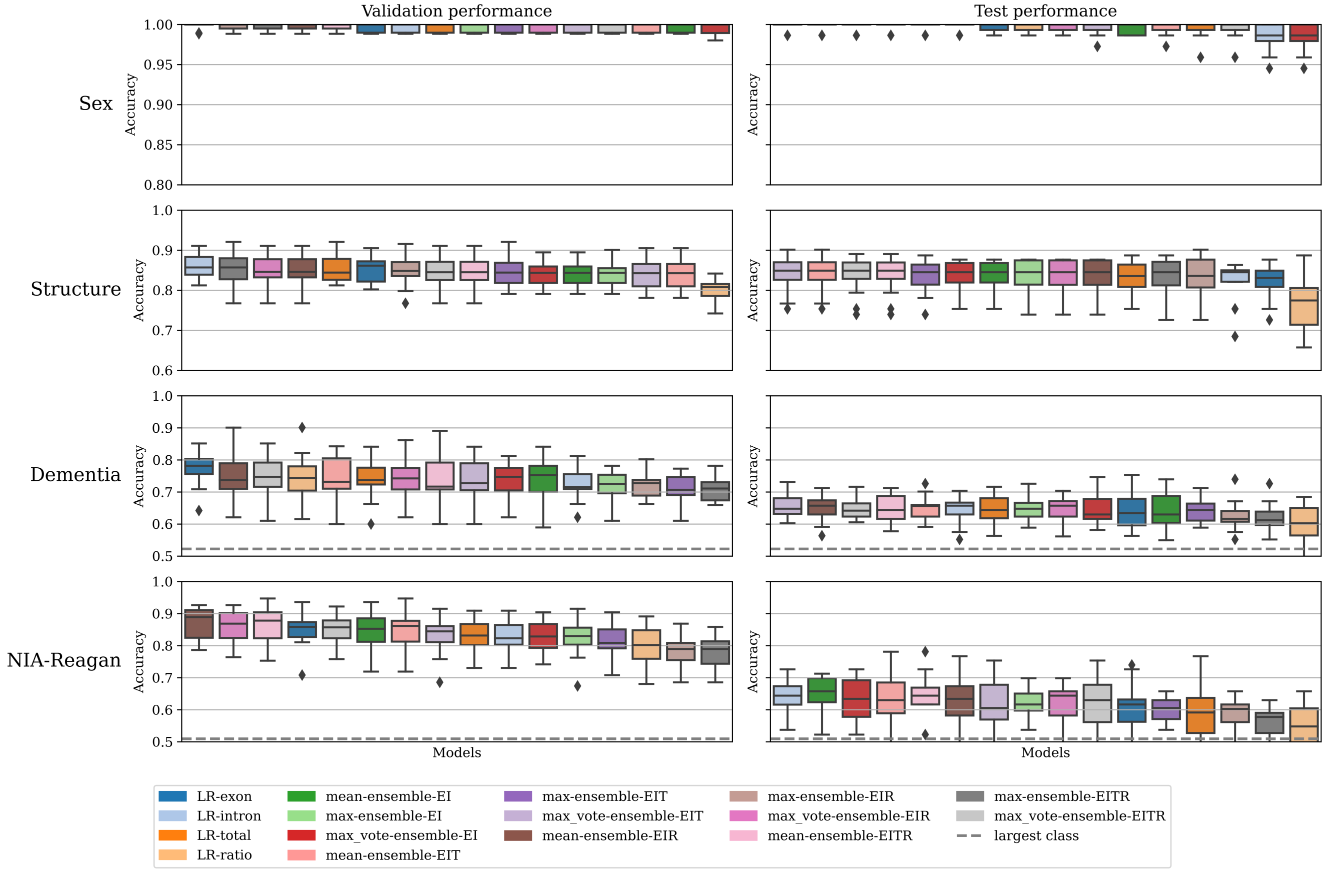


Figure 11: Predictive performance of our base models and their ensembles. The first column depicts validation data. The second column depicts test data. The rows correspond to the prediction labels. The largest class of a label is indicated by the grey line, which indicates a guessing performance. The models are ordered from left to right based on their mean accuracy.

remains excellent. For *structure* we see a substantial decline in both validation and test performance, which can be attributed to the aforementioned drawback. The effect for *dementia* mainly seems to be an increase in variation. The validation performance for *NIA-Reagan* increases further with *pca-EIR* above the previous best, but again no apparent test performance increase. None of the models is significantly better than the base models (Supplemental Table 5, 6, 7 and 8). As the training accuracy is very good to excellent for all labels (Supplemental Figure 12)), we conclude that the heterogeneity in the data causes the PCA transformation to have an adverse effect on the model performance.

Model Ensembles Fail to Improve Performance and Robustness

Our ensemble approaches only yielded an increase in validation performance for *NIA-Reagan* (Figure 11). Nevertheless, this did not result in an increased test performance. For the other labels, it is the other way around. They all have a base model with the best validation performance but do have an ensemble approach as the best test performance, albeit very marginal. We also don't observe a clear improvement in stability. Hence, we conclude that these ensemble approaches are not beneficial. This further indicates that the information, relevant to the prediction task, contained within the different datasets is largely the same.

Limitations & Further Research

Our ability to benefit from including intronic read counts is constrained by our initial step of feature selection by our DGE analysis. This constrains us in two ways. *Limma-Voom* considers genes in a univariate fashion hereby not considering gene interactions or genes that are only differentially expressed in a combination with another gene. Furthermore, in subsequent steps, we are limited to the information contained within the selected genes. We limit ourselves to the top thousand most differentially expressed genes. Although, this is a viable approach when performance is the goal. It could be that the first thousand differentially expressed genes within exonic and intronic reads contain the same signal and that the differences only occur in less differentially expressed genes. We suggest further research to explore other ways of feature selection, or if time and computational resources allow it, take all genes into consideration.

Another constraint of our approach is that we limit ourselves to the use of relatively simple models. We employed logistic regression and other simple model architectures. Recent work has shown the successful application of complex deep learning architectures, such as graph networks, convolution neural networks and autoencoders to RNA-seq data [42, 43]. Our results did not directly warrant the implementation of these complex networks as we saw that increasing the number of learnable parameters did not result in improved performance. However, it may be that these models are better able to extract differences between exonic and intronic reads. More complex models might be better in modelling the complex relationships between exonic and intronic counts or more advanced regularization techniques used in these architectures might be better equipped to handle the high dimensionality. Furthermore, graph convultion networks for example can incorporate additional gene-gene interaction information, which may also relate to the exonic-intronic difference. These model architectures are already being used for RNA-seq-based prediction tasks. Given the ease of obtaining the intronic read counts when one is obtaining the exonic read counts, we would advise trying the combination of exonic and intronic reads as input for these complex deep learning networks.

Additionally, we suggest exploring different datasets related to other diseases or originating from other tissue types. Our findings are exclusive to our dataset's brain regions, as highlighted before we have a relatively large percentage of intronic to total reads. Apparently to such an extent that the information relevant for the prediction task is largely the same in the exonic and intronic reads. However, this might be different for other tissue types. Existing literature supports the possibility that aberrations may only manifest in intronic read counts. Now that we have confidently established that intronic reads do contain biologically relevant signal, it is worth exploring the vast number of existing RNA-seq datasets.

Shortcommings of Related Work

We were not able to get a good predictive performance for the dichotomized *NIA-Reagan* label. Nonetheless, good to even excellent predictive performance is obtained in the literature for predicting AD from gene expression data [32, 44, 31]. We note that some of these studies perform one or more steps of their workflow in such a way that information leakage is present. Dag et al. [44] (92.9% accuracy) performs trimmed mean of M values (TMM) normalisation on the entire dataset at once. This normalization technique uses information from all the samples to scale individual samples. In this way, the test set is too optimistically normalized. Another questionable approach is to perform DGE analysis on the training and validation set combined. Alamro et al. [32] does this and obtains a near-perfect validation score (0.979 AUC). However, testing it on an independent test set (0.75 AUC) reveals a large discrepancy. Mahendran et al. [31] even perform DGE analysis on the entire dataset at one, achieving a test score of 96.78% accuracy. Considering that we already observed severe overfitting with our intersection approach, their model will probably perform substantially worse on a fully independent test set.

Conclusion

In this work, we found that intronic reads do contain strong biological signal for RNA-seq-based prediction tasks. For all our labels intronic and total read counts performed similarly with respect to the traditional approach of using exonic read counts. The count ratios, however, did perform significantly worse for *NIA-Reagan* and *structure*. Our DGE analysis identified overlapping and uniquely expressed genes for every count set. We investigated a number of ways to extract additional information from these different genes to realize a predictive performance gain above the base models. Despite these different genes, we were not able to realize a statistically significant improvement. Furthermore, we made three observations by analyzing our results. First, the PCA plots of exonic and intronic read counts showed a very high degree of similarity. Second, we observed high correlations between the exonic and intronic read counts of our DGE selected genes. Lastly, we saw that when combining the intronic and exonic counts in our concatenation models, the relative feature importances of both sets are approximately the same. From these observations, we conclude that the relevant biological signal in the different count sets is largely the same and that for our dataset there is no trivial way to leverage the inclusion of intronic read counts to obtain an increase in predictive performance.

References

1. Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. Analyzing Protein Structure and Function. In *Molecular Biology of the Cell. 4th edition.* Garland Science, 2002.
2. Bong-Seok Jo and Sun Shim Choi. Introns: The Functional Benefits of Introns in Genomes. *Genomics & Informatics*, 13(4):112–118, December 2015.
3. RNA Processing - Biochemistry - Medbullets Step 1.
4. Thale Kristin Olsen and Ninib Baryawno. Introduction to Single-Cell RNA Sequencing. *Current Protocols in Molecular Biology*, 122(1):e57, April 2018.
5. Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics*, 10(1):57–63, January 2009.
6. Stuart Lee, Albert Y Zhang, Shian Su, Ashley P Ng, Aliaksei Z Holik, Marie-Liesse Asselin-Labat, Matthew E Ritchie, and Charity W Law. Covering all your bases: incorporating intron signal from RNA-seq data. *NAR Genomics and Bioinformatics*, 2(3):lqaa073, September 2020.
7. RNA-seq: the basics, January 2021.
8. Dimos Gaidatzis, Lukas Burger, Maria Florescu, and Michael B. Stadler. Analysis of intronic and exonic reads in RNA-seq data characterizes transcriptional and post-transcriptional regulation. *Nature Biotechnology*, 33(7):722–729, July 2015. Number: 7 Publisher: Nature Publishing Group.
9. Aaron M. Smith, Jonathan R. Walsh, John Long, Craig B. Davis, Peter Henstock, Martin R. Hodge, Mateusz Maciejewski, Xinmeng Jasmine Mu, Stephen Ra, Shanrong Zhao, Daniel Ziemek, and Charles K. Fisher. Standard machine learning approaches outperform deep representation learning on phenotype prediction from transcriptomics data. *BMC Bioinformatics*, 21(1):119, March 2020.
10. P. Roman-Naranjo, A. M. Parra-Perez, and J. A. Lopez-Escamez. A systematic review on machine learning approaches in the diagnosis and prognosis of rare genetic diseases. *Journal of Biomedical Informatics*, 143:104429, July 2023.
11. Jamal Tazi, Nadia Bakkour, and Stefan Stamm. Alternative splicing and disease. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1792(1):14–26, January 2009.
12. Sulev Koks, Abigail L Pfaff, Vivien J Bubb, and John P Quinn. Longitudinal intronic RNA-Seq analysis of Parkinson's disease patients reveals disease-specific nascent transcription. *Experimental Biology and Medicine*, 247(11):945–957, June 2022.

13. Karol Andrea Arizaca Maquera, Justin Ralph Welden, Giorgi Margvelani, Sandra C. Miranda Sardón, Samantha Hart, Noémie Robil, Alvaro Gonzalo Hernandez, Pierre de la Grange, Peter T. Nelson, and Stefan Stamm. Alzheimer's disease pathogenetic progression is associated with changes in regulated retained introns and editing of circular RNAs. *Frontiers in Molecular Neuroscience*, 16, 2023.
14. Anita H. Corbett. Post-transcriptional Regulation of Gene Expression and Human Disease. *Current opinion in cell biology*, 52:96–104, June 2018.
15. Amy Webb, Audrey C. Papp, Amanda Curtis, Leslie C. Newman, Maciej Pietrzak, Michal Seweryn, Samuel K. Handelman, Grzegorz A. Rempala, Daqing Wang, Erica Graziosa, Rachel F. Tyndale, Caryn Lerman, John R. Kelsoe, Deborah C. Mash, and Wolfgang Sadee. RNA sequencing of transcriptomes in human brain regions: protein-coding and non-coding RNAs, isoforms and alleles. *BMC Genomics*, 16:990, November 2015.
16. Chun-Hao Su, Dhananjaya D, and Woan-Yuh Tarn. Alternative Splicing in Neurogenesis and Brain Development. *Frontiers in Molecular Biosciences*, 5, 2018.
17. Jeremy A Miller, Angela Guillozet-Bongaarts, Laura E Gibbons, Nadia Postupna, Anne Renz, Allison E Beller, Susan M Sunkin, Lydia Ng, Shannon E Rose, Kimberly A Smith, Aaron Szafer, Chris Barber, Darren Bertagnolli, Kristopher Bickley, Krissy Brouner, Shiella Caldejon, Mike Chapin, Mindy L Chua, Natalie M Coleman, Eiron Cudaback, Christine Cuhaciyan, Rachel A Dalley, Nick Dee, Tsega Desta, Tim A Dolbeare, Nadezhda I Dotson, Michael Fisher, Nathalie Gaudreault, Garrett Gee, Terri L Gilbert, Jeff Goldy, Fiona Griffin, Caroline Habel, Zeb Haradon, Nika Hejazinia, Leanne L Hellstern, Steve Horvath, Kim Howard, Robert Howard, Justin Johal, Nikolas L Jorstad, Samuel R Josephsen, Chihchau L Kuan, Florence Lai, Eric Lee, Felix Lee, Tracy Lemon, Xianwu Li, Desiree A Marshall, Jose Melchor, Shubhabrata Mukherjee, Julie Nyhus, Julie Pendergraft, Lydia Potekhina, Elizabeth Y Rha, Samantha Rice, David Rosen, Abharika Sapru, Aimee Schantz, Elaine Shen, Emily Sherfield, Shu Shi, Andy J Sodt, Nivretta Thatra, Michael Tieu, Angela M Wilson, Thomas J Montine, Eric B Larson, Amy Bernard, Paul K Crane, Richard G Ellenbogen, C Dirk Keene, and Ed Lein. Neuropathological and transcriptomic characteristics of the aged brain. *eLife*, 6:e31126, November 2017. Publisher: eLife Sciences Publications, Ltd.
18. Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, August 2009.
19. Bo Li and Colin N. Dewey. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12(1):323, August 2011.
20. Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, April 2012. Number: 4 Publisher: Nature Publishing Group.
21. A. Gmitrowicz and A. Kucharska. [Developmental disorders in the fourth edition of the American classification: diagnostic and statistical manual of mental disorders (DSM IV – optional book)]. *Psychiatria Polska*, 28(5):509–521, 1994.
22. Heiko Braak, Irina Alafuzoff, Thomas Arzberger, Hans Kretzschmar, and Kelly Del Tredici. Staging of Alzheimer disease-associated neurofibrillary pathology using paraffin sections and immunocytochemistry. *Acta Neuropathologica*, 112(4):389–404, 2006.
23. J. C. Moms, A. Heyman, R. C. Mohs, J. P. Hughes, G. van Belle, G. Fillenbaum, E. D. Mellits, and C. Clark. The Consortium to Establish a Registry for Alzheimer's Disease (CERAD). Part I. Clinical and neuropsychological assesment of Alzheimer's disease. *Neurology*, 39(9):1159–1159, September 1989. Publisher: Wolters Kluwer Health, Inc. on behalf of the American Academy of Neurology Section: Articles.
24. K. L. Newell, B. T. Hyman, J. H. Growdon, and E. T. Hedley-Whyte. Application of the National Institute on Aging (NIA)-Reagan Institute criteria for the neuropathological diagnosis of Alzheimer disease. *Journal of Neuropathology and Experimental Neurology*, 58(11):1147–1155, November 1999.
25. Dimos Gaidatzis, Anita Lerch, Florian Hahne, and Michael B. Stadler. QuasR: quantification and annotation of short reads in R. *Bioinformatics*, 31(7):1130–1132, April 2015.
26. Peter Robinson Hansen, Peter. SAM/BAM Format. In *Computational Exome and Genome Analysis*. Chapman and Hall/CRC, 2017. Num Pages: 18.
27. Simon Anders, Paul Theodor Pyl, and Wolfgang Huber. HTSeq – A Python framework to work with high-throughput sequencing data, August 2014. Pages: 002824 Section: New Results.
28. Andreas Schroeder, Odilo Mueller, Susanne Stocker, Ruediger Salowsky, Michael Leiber, Marcus Gassmann, Samar Lightfoot, Wolfram Menzel, Martin Granzow, and Thomas Ragg. The RIN: an RNA integrity number for assigning integrity values to RNA measurements. *BMC Molecular Biology*, 7(1):3, January 2006.
29. Ciaran Evans, Johanna Hardin, and Daniel M Stoebel. Selecting between-sample RNA-Seq normalization methods from the perspective of their assumptions. *Briefings in Bioinformatics*, 19(5):776–792, February 2017.
30. Richard Bellman. Dynamic Programming. *Science*, 153(3731):34–37, July 1966. Publisher: American Association for the Advancement of Science.
31. Nivedhitha Mahendran, P. M. Durai Raj Vincent, Kathiravan Srinivasan, and Chuan-Yu Chang. Improving the Classification of Alzheimer's Disease Using Hybrid Gene Selection Pipeline and Deep Learning. *Frontiers in Genetics*, 12:784814, November 2021.
32. Hind Alamro, Maha A. Thafar, Somayah Albaradei, Takashi Gojobori, Magbubah Essack, and Xin Gao. Exploiting machine learning models to identify novel Alzheimer's disease biomarkers and potential targets. *Scientific Reports*, 13(1):4979, March 2023. Number: 1 Publisher: Nature Publishing Group.
33. Matthew E. Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W. Law, Wei Shi, and Gordon K. Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47, April 2015.
34. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. arXiv:1912.01703 [cs, stat].

35. PyTorch documentation — PyTorch 2.1 documentation.
36. Santosh Manicka, Kathleen Johnson, Michael Levin, and David Murrugarra. The nonlinearity of regulation in biological networks. *npj Systems Biology and Applications*, 9(1):1–9, April 2023. Number: 1 Publisher: Nature Publishing Group.
37. Arno Steinacher, Declan G. Bates, Ozgur E. Akman, and Orkun S. Soyer. Nonlinear Dynamics in Gene Regulation Promote Robustness and Evolvability of Gene Expression Levels. *PLOS ONE*, 11(4):e0153295, April 2016. Publisher: Public Library of Science.
38. Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA, July 2019. Association for Computing Machinery.
39. Habshah Midi, S.K. Sarkar, and Sohel Rana. Collinearity diagnostics of binary logistic regression model. *Journal of Interdisciplinary Mathematics*, 13(3):253–267, June 2010. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/09720502.2010.10700699.
40. Lexi V Perez. Principal Component Analysis to Address Multicollinearity.
41. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.
42. Diksha Pandey and P. Onkara Perumal. A scoping review on deep learning for next-generation RNA-Seq. data analysis. *Functional & Integrative Genomics*, 23(2):134, April 2023.
43. Ahmed Elmahy, Sherin Aly, and Fayek Elkhwsky. Cancer Stage Prediction From Gene Expression Data Using Weighted Graph Convolution Network. In *2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)*, pages 231–236, September 2021.
44. Osman Dag, Merve Kasikci, Ozlem Ilk, and Metin Yesiltepe. GeneSelectML: a comprehensive way of gene selection for RNA-Seq data via machine learning algorithms. *Medical & Biological Engineering & Computing*, 61(1):229–241, January 2023.

Supplementary Materials

Supplemental Tables

Label	Set	Method	Fold 0	Fold 1	Fold 2	Fold 3	Fold 4	STD	CV
Sex	exon	top 1000 intersection	193	241	141	151	184	39.52	**0.22**
		p <0.05	829	987	457	434	471	255.21	0.40
		padj <0.05	80	89	40	42	41	24.05	0.41
	intron	top 1000 intersection	216	216	191	181	199	15.44	**0.08**
		p <0.05	1314	1103	736	527	782	313.16	0.35
		padj <0.05	121	94	67	49	58	29.44	0.38
	total	top 1000 intersection	225	226	178	154	197	30.94	**0.16**
		p <0.05	1132	1099	593	478	627	306.22	0.39
		padj <0.05	122	104	56	49	61	32.50	0.41
structure	exon	top 1000 intersection	748	748	735	771	743	13.40	**0.02**
		p <0.05	11265	12560	11479	11495	11851	509.41	0.04
		padj <0.05	10817	12260	10999	11002	11471	583.75	0.05
	intron	top 1000 intersection	714	708	712	709	697	6.60	**0.01**
		p <0.05	11522	12600	12082	11550	11982	442.75	0.04
		padj <0.05	11127	12305	11698	11152	11627	482.22	0.04
	total	top 1000 intersection	769	745	728	756	727	18.10	**0.02**
		p <0.05	12110	13317	12413	12373	12729	462.78	0.04
		padj <0.05	11759	13073	12064	12026	12445	509.53	0.04
Dementia	exon	top 1000 intersection	215	236	238	268	421	83.45	0.30
		p <0.05	3818	3899	4164	4357	5187	547.84	**0.13**
		padj <0.05	2190	2449	2593	2728	3927	672.85	0.24
	intron	top 1000 intersection	268	245	277	306	428	72.25	0.24
		p <0.05	3400	3873	3925	3736	4610	442.14	**0.11**
		padj <0.05	1718	2349	2526	2188	3226	550.18	0.23
	total	top 1000 intersection	234	230	248	270	411	75.65	0.27
		p <0.05	3959	4331	4505	4471	5313	495.44	**0.11**
		padj <0.05	2356	2880	3015	2947	4104	638.89	0.21
NIA-Regan	exon	top 1000 intersection	118	106	118	99	62	23.06	**0.23**
		p <0.05	213	544	317	740	131	249.97	0.64
		padj <0.05	1	8	0	30	0	12.85	1.65
	intron	top 1000 intersection	97	126	78	130	98	21.82	**0.21**
		p <0.05	158	470	211	805	205	272.34	0.74
		padj <0.05	1	3	0	37	1	16.02	1.91
	total	top 1000 intersection	118	108	104	106	83	12.81	**0.12**
		p <0.05	186	565	295	869	158	300.63	0.73
		padj <0.05	0	4	0	39	1	16.96	1.93

Supplemental Table 1. Table of differentially expressed genes identified by DGE analysis, along with the standard deviation (STD) and the coefficient of variation (CV), which is the standard deviation divided by the mean. For each test fold, genes are selected by taking the intersection of genes identified from the corresponding cross-validation folds.

Hyperparameter	**Minimal value**	**Maximal value**
Learning rate	1e-7	1e-2
Batch size	15	128
Weight decay	1e-5	1e2

Supplemental Table 2. Table of hyperparameters searched and their search range.

Package	**Version**
pandas	1.5.2
numpy	1.24.3
pytorch	1.13.1
sklearn	1.2.0
scipy	1.9.3
optuna	3.1.1
Python	3.11.0rc1
R	4.1.2
index	1.0
edgeR	3.36.0
GenomicFeatures	1.46.5
eisaR	1.6.0
QuasR	1.34.0

Supplemental Table 3. Table listing software packages and their versions.

Label	Model	Mean Accuracy	P-value
sex	LR-exon	0.9963	1.0000
sex	LR-intron	0.9863	0.1150
sex	LR-total	0.9945	0.9339
sex	LR-ratio	0.9963	1.0000
structure_acronym	LR-exon	0.8211	1.0000
structure_acronym	LR-intron	0.8211	0.6783
structure_acronym	LR-total	0.8307	0.5897
structure_acronym	LR-ratio	0.7668	0.0161
act_demented	LR-exon	0.6392	1.0000
act_demented	LR-intron	0.6480	0.4807
act_demented	LR-total	0.6468	0.6482
act_demented	LR-ratio	0.6048	0.1198
nia_grouped	LR-exon	0.6109	1.0000
nia_grouped	LR-intron	0.6452	0.0971
nia_grouped	LR-total	0.5921	0.6482
nia_grouped	LR-ratio	0.5284	0.0362

Supplemental Table 4. Table with accuracies and significance tests per model. We test if the obtained test accuracies for our models using intronic read counts, total read counts and the count ratios are significantly different compared to the traditional approach of using exonic read counts for RNA-seq-based prediction tasks. Significance is tested by the Wilcoxon ranked-sum test.

Label	Model	Base Model	p-value	statistic	mean accuracy	delta mean	std	delta std
Sex	A-EIRT	LR-exon	0.7557	0.3111	0.9973	0.0009	0.0055	-0.0006
Sex	LR-ratio	LR-exon	1.0000	0.0000	0.9963	0.0000	0.0061	0.0000
Sex	concat-EIR	LR-exon	1.0000	0.0000	0.9963	0.0000	0.0061	0.0000
Sex	concat-EI	LR-exon	0.9339	-0.0830	0.9953	-0.0011	0.0086	0.0026
Sex	concat-EIRT	LR-exon	0.6936	-0.3940	0.9936	-0.0027	0.0110	0.0050
Sex	pca-EIR	LR-exon	0.6334	-0.4770	0.9918	-0.0046	0.0140	0.0079
Sex	pca-EIRT	LR-exon	0.6334	-0.4770	0.9918	-0.0046	0.0140	0.0079
Sex	B-EI	LR-exon	0.2717	-1.0992	0.9909	-0.0055	0.0119	0.0058
Sex	pca-EI	LR-exon	0.3837	-0.8710	0.9908	-0.0055	0.0139	0.0078
Sex	A-EI	LR-exon	0.6936	-0.3940	0.9890	-0.0073	0.0270	0.0210
Sex	A-EIR	LR-exon	0.6334	-0.4770	0.9881	-0.0082	0.0228	0.0168
Sex	act-B-EI	LR-exon	0.0225	-2.2813	0.9871	-0.0093	0.0107	0.0047
Sex	act-A-EIRT	LR-exon	0.0815	-1.7421	0.9834	-0.0130	0.0199	0.0138
Sex	act-A-EIR	LR-exon	0.0401	-2.0532	0.9388	-0.0576	0.1642	0.1582
Sex	act-A-EI	LR-exon	0.3837	-0.8710	0.9132	-0.0831	0.2089	0.2029

Supplemental Table 5. Significance testing for label *Sex*. The base model is the best-performing logistic regression model with a single data input set. We test if the obtained test accuracies for each model are significantly different than the base model. The p-value is calculated using a Wilcoxon rank-sum test, if the null hypothesis is rejected ($p<0.05$) then the samples are significantly different. Delta mean is the difference between the mean accuracy of the model and the mean accuracy of the best base model.

Label	Model	Base Model	p-value	statistic	mean accuracy	delta mean	std	delta std
Structure	concat-EI	LR-total	0.9010	-0.1244	0.8292	-0.0015	0.0432	0.0015
Structure	A-EI	LR-total	0.6482	-0.4563	0.8220	-0.0086	0.0437	0.0020
Structure	B-EI	LR-total	0.2998	-1.0370	0.8153	-0.0154	0.0389	-0.0028
Structure	pca-EI	LR-total	0.3507	-0.9333	0.8104	-0.0202	0.0558	0.0141
Structure	A-EIR	LR-total	0.1585	-1.4103	0.8089	-0.0218	0.0430	0.0013
Structure	pca-EIRT	LR-total	0.1198	-1.5554	0.8067	-0.0240	0.0428	0.0011
Structure	concat-EIR	LR-total	0.1300	-1.5139	0.8060	-0.0247	0.0517	0.0101
Structure	A-EIRT	LR-total	0.0620	-1.8665	0.8051	-0.0256	0.0348	-0.0068
Structure	concat-EIRT	LR-total	0.1198	-1.5554	0.8033	-0.0274	0.0486	0.0069
Structure	act-A-EI	LR-total	0.0649	-1.8458	0.8023	-0.0284	0.0440	0.0024
Structure	pca-EIR	LR-total	0.0649	-1.8458	0.7966	-0.0340	0.0485	0.0069
Structure	act-A-EIRT	LR-total	0.0079	-2.6546	0.7752	-0.0554	0.0612	0.0195
Structure	act-B-EI	LR-total	0.0070	-2.6961	0.7736	-0.0571	0.0542	0.0126
Structure	act-A-EIR	LR-total	0.0051	-2.7998	0.7732	-0.0575	0.0559	0.0143
Structure	LR-ratio	LR-total	0.0070	-2.6961	0.7668	-0.0639	0.0643	0.0226

Supplemental Table 6. Significance testing for label *Structure*. The base model is the best-performing logistic regression model with a single data input set. We test if the obtained test accuracies for each model are significantly different than the base model. The p-value is calculated using a Wilcoxon rank-sum test, if the null hypothesis is rejected ($p<0.05$) then the samples are significantly different. Delta mean is the difference between the mean accuracy of the model and the mean accuracy of the best base model.

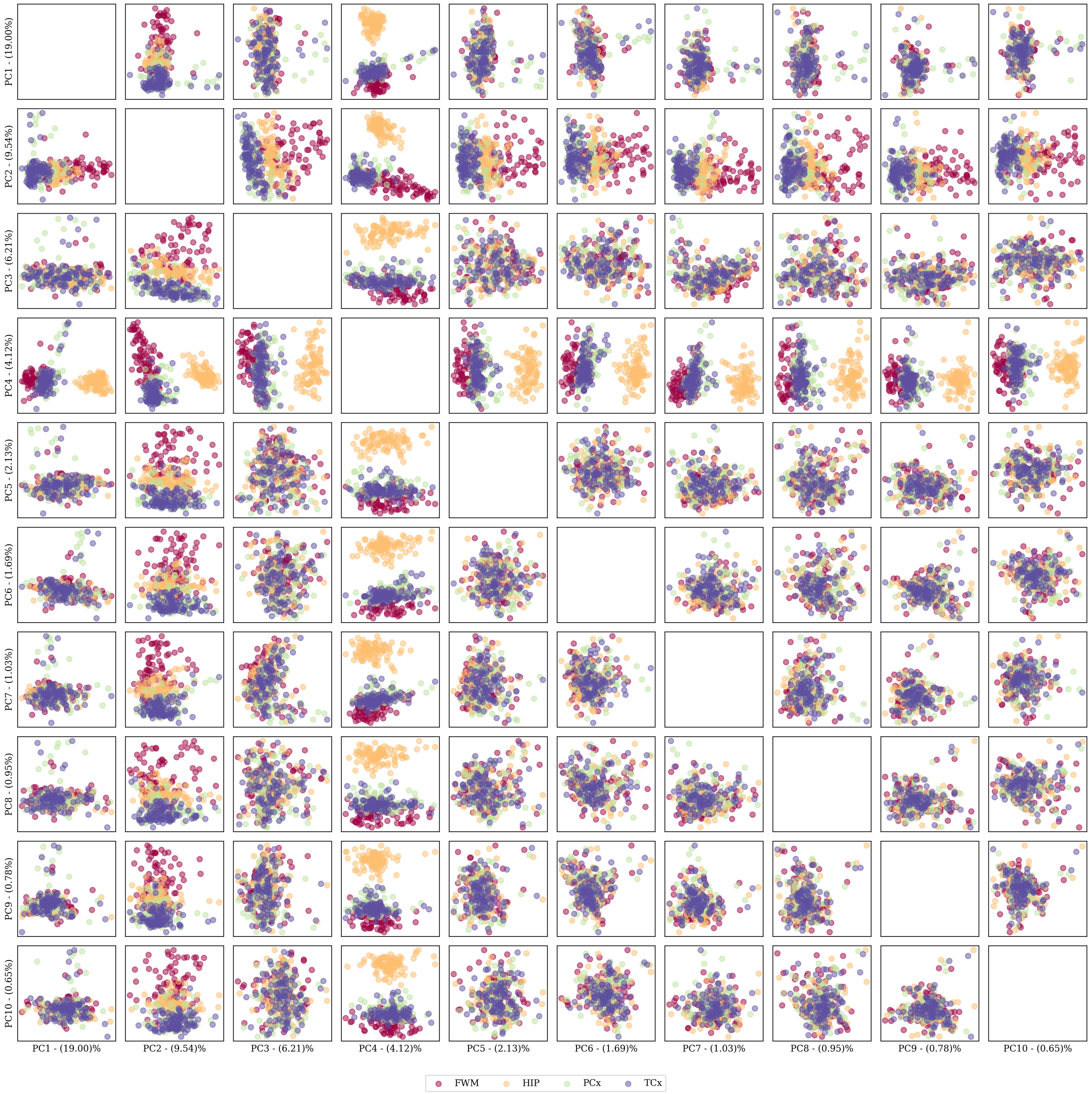
Label	Model	Base Model	p-value	statistic	mean accuracy	delta mean	std	delta std
Dementia	act-B-EI	LR-intron	0.5476	0.6014	0.6543	0.0063	0.0408	-0.0007
Dementia	B-EI	LR-intron	0.4679	-0.7259	0.6410	-0.0070	0.0381	-0.0033
Dementia	A-EI	LR-intron	0.6187	-0.4977	0.6353	-0.0127	0.0525	0.0110
Dementia	concat-EI	LR-intron	0.5476	-0.6014	0.6335	-0.0145	0.0476	0.0061
Dementia	act-A-EI	LR-intron	0.7716	-0.2903	0.6325	-0.0155	0.0805	0.0390
Dementia	pca-EIR	LR-intron	0.3401	-0.9540	0.6309	-0.0171	0.0595	0.0180
Dementia	pca-EIRT	LR-intron	0.3615	-0.9125	0.6267	-0.0213	0.0625	0.0210
Dementia	pca-EI	LR-intron	0.2372	-1.1821	0.6235	-0.0245	0.0487	0.0072
Dementia	act-A-EIRT	LR-intron	0.3195	-0.9955	0.6186	-0.0294	0.0645	0.0231
Dementia	concat-EIR	LR-intron	0.0890	-1.7006	0.6172	-0.0308	0.0617	0.0202
Dementia	concat-EIRT	LR-intron	0.1300	-1.5139	0.6155	-0.0325	0.0665	0.0250
Dementia	A-EIRT	LR-intron	0.1914	-1.3066	0.6155	-0.0325	0.0613	0.0199
Dementia	A-EIR	LR-intron	0.0712	-1.8043	0.6088	-0.0392	0.0677	0.0263
Dementia	act-A-EIR	LR-intron	0.0971	-1.6591	0.6080	-0.0400	0.0811	0.0397
Dementia	LR-ratio	LR-intron	0.0362	-2.0946	0.6048	-0.0431	0.0524	0.0109

Supplemental Table 7. Significance testing for label *Dementia*. The base model is the best-performing logistic regression model with a single data input set. We test if the obtained test accuracies for each model are significantly different than the base model. The p-value is calculated using a Wilcoxon rank-sum test, if the null hypothesis is rejected ($p<0.05$) then the samples are significantly different. Delta mean is the difference between the mean accuracy of the model and the mean accuracy of the best base model.

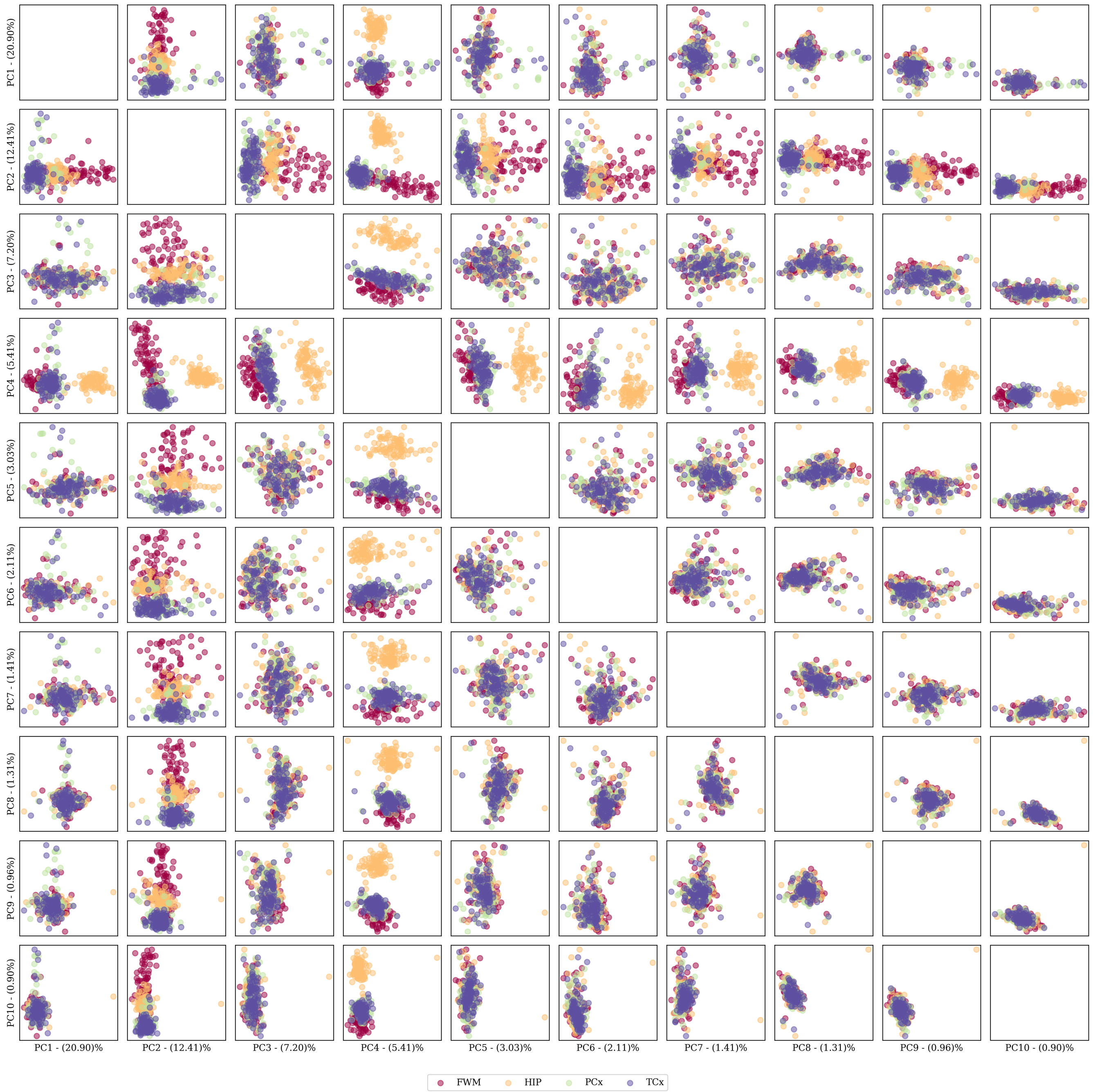
Label	Model	Base Model	p-value	statistic	mean accuracy	delta mean	std	delta std
NIA-Reagan	A-EI	LR-intron	0.3725	-0.8918	0.6298	-0.0154	0.0633	0.0176
NIA-Reagan	concat-EI	LR-intron	0.1466	-1.4517	0.6223	-0.0229	0.0609	0.0152
NIA-Reagan	pca-EI	LR-intron	0.1354	-1.4932	0.6154	-0.0298	0.0482	0.0024
NIA-Reagan	act-B-EI	LR-intron	0.3401	-0.9540	0.6153	-0.0299	0.0802	0.0344
NIA-Reagan	concat-EIR	LR-intron	0.1711	-1.3688	0.6093	-0.0359	0.0766	0.0308
NIA-Reagan	B-EI	LR-intron	0.0745	-1.7836	0.6076	-0.0376	0.0579	0.0121
NIA-Reagan	A-EIRT	LR-intron	0.1013	-1.6384	0.6069	-0.0384	0.0761	0.0304
NIA-Reagan	concat-EIRT	LR-intron	0.1249	-1.5347	0.6039	-0.0413	0.0745	0.0288
NIA-Reagan	A-EIR	LR-intron	0.1466	-1.4517	0.6012	-0.0440	0.0754	0.0296
NIA-Reagan	pca-EIR	LR-intron	0.0465	-1.9909	0.5996	-0.0456	0.0724	0.0266
NIA-Reagan	act-A-EIRT	LR-intron	0.0712	-1.8043	0.5916	-0.0536	0.0755	0.0297
NIA-Reagan	act-A-EI	LR-intron	0.0037	-2.9035	0.5821	-0.0631	0.0496	0.0038
NIA-Reagan	pca-EIRT	LR-intron	0.0213	-2.3020	0.5802	-0.0650	0.0762	0.0305
NIA-Reagan	act-A-EIR	LR-intron	0.0017	-3.1316	0.5797	-0.0655	0.0539	0.0081
NIA-Reagan	LR-ratio	LR-intron	0.0007	-3.3805	0.5284	-0.1168	0.0928	0.0470

Supplemental Table 8. Significance testing for label *NIA-Reagan*. The base model is the best-performing logistic regression model with a single data input set. We test if the obtained test accuracies for each model are significantly different than the base model. The p-value is calculated using a Wilcoxon rank-sum test, if the null hypothesis is rejected ($p<0.05$) then the samples are significantly different. Delta mean is the difference between the mean accuracy of the model and the mean accuracy of the best base model.

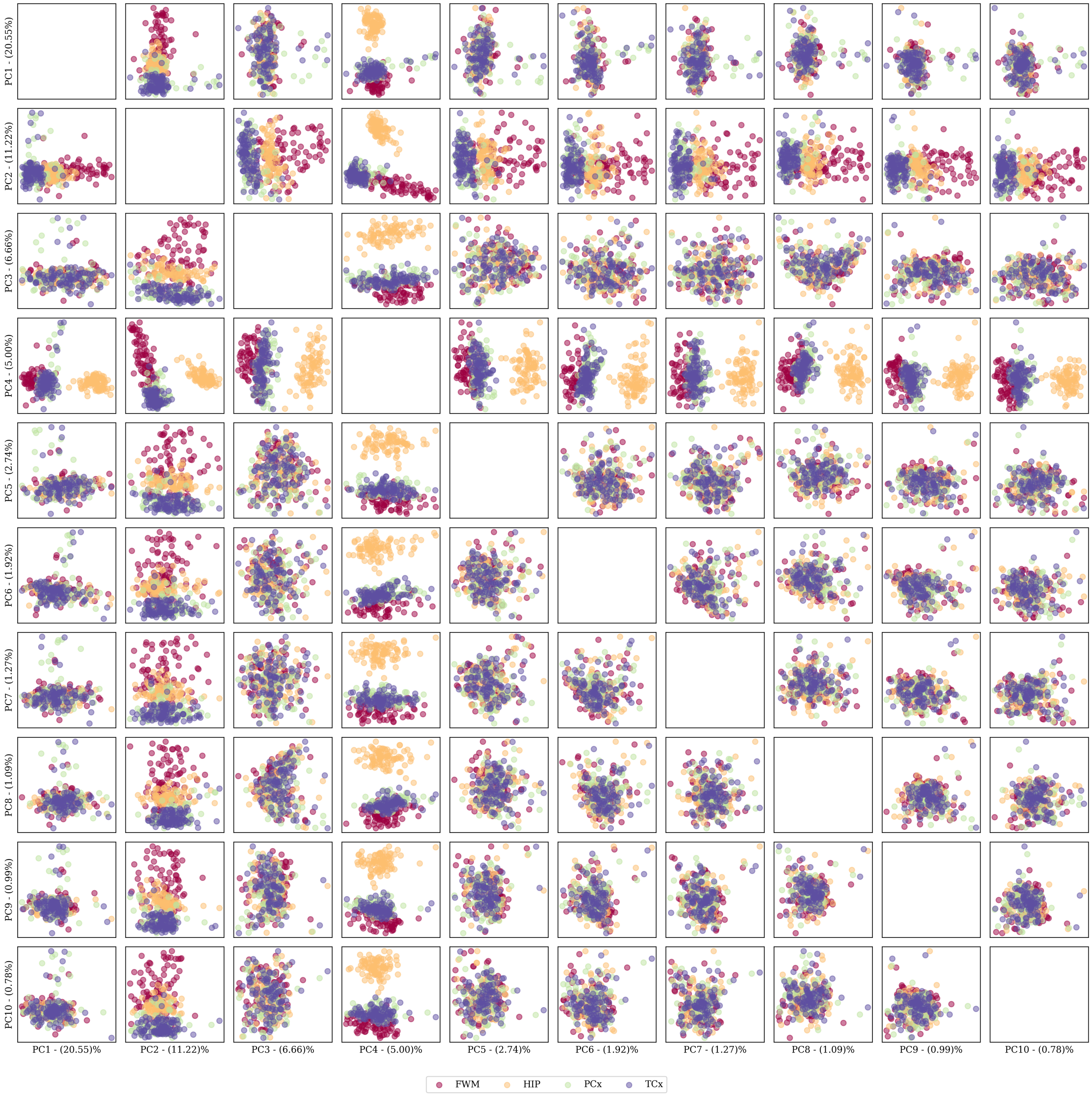
Supplemental Figures



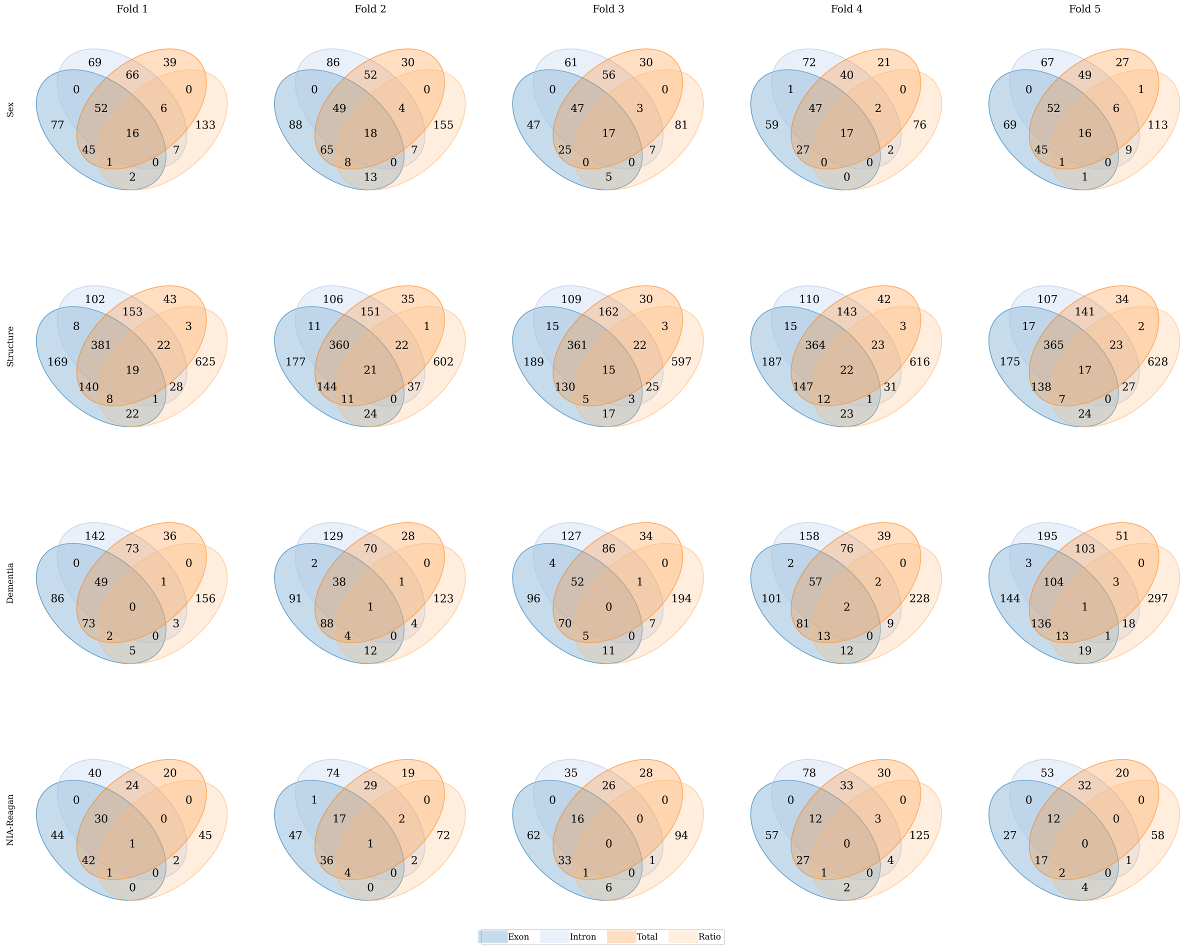
Supplemental Figure 1: PCA pairs plot of the first 10 principal components of the transformed exonic read counts. The samples are coloured by brain region; Hippocampus (HIP), frontal white matter (FWM), parietal cortex (PCx) and temporal cortex (TCx). The row and column indices represent the specific principal components being plotted against each other. The labels include the percentage of explained variability of that principal component.



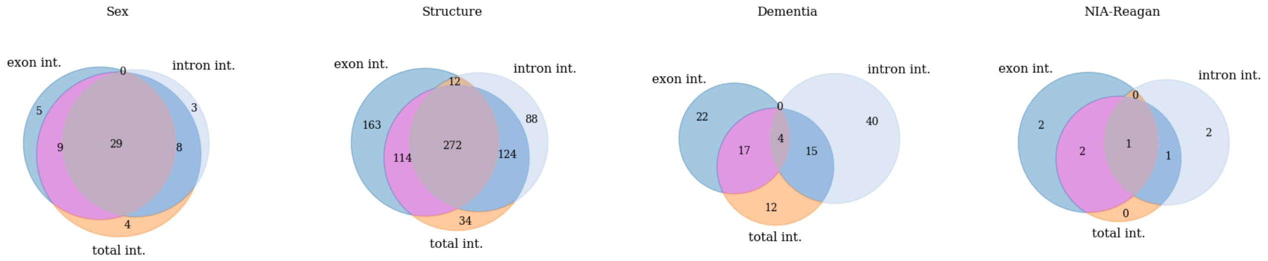
Supplemental Figure 2: PCA pairs plot of the first 10 principal components of the transformed intronic read counts. The samples are coloured by brain region; Hippocampus (HIP), frontal white matter (FWM), parietal cortex (PCx) and temporal cortex (TCx). The row and column indices represent the specific principal components being plotted against each other. The labels include the percentage of explained variability of that principal component.



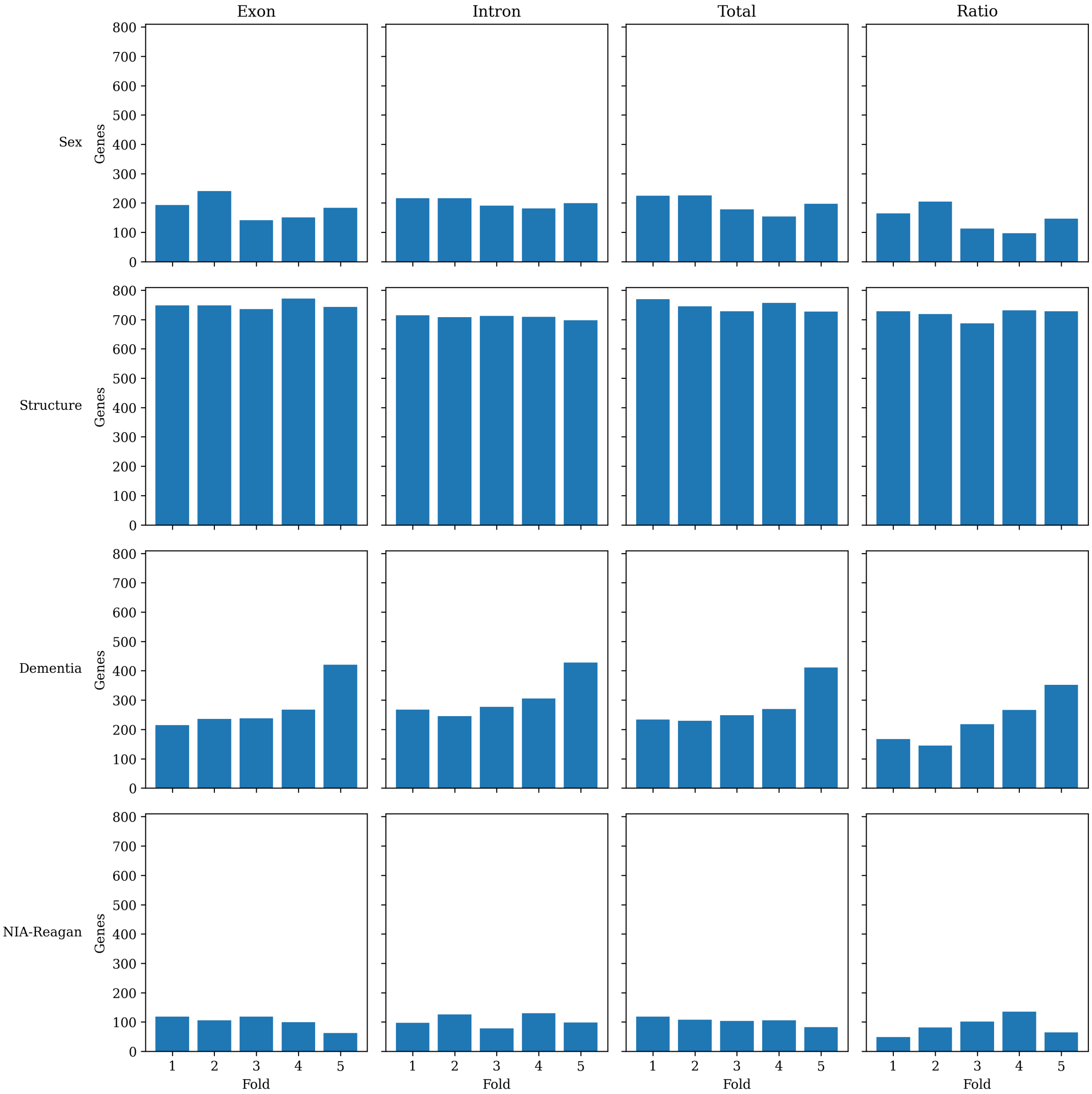
Supplemental Figure 3: PCA pairs plot of the first 10 principal components of the transformed total read counts. The samples are coloured by brain region; Hippocampus (HIP), frontal white matter (FWM), parietal cortex (PCx) and temporal cortex (TCx). The row and column indices represent the specific principal components being plotted against each other. The labels include the percentage of explained variability of that principal component.



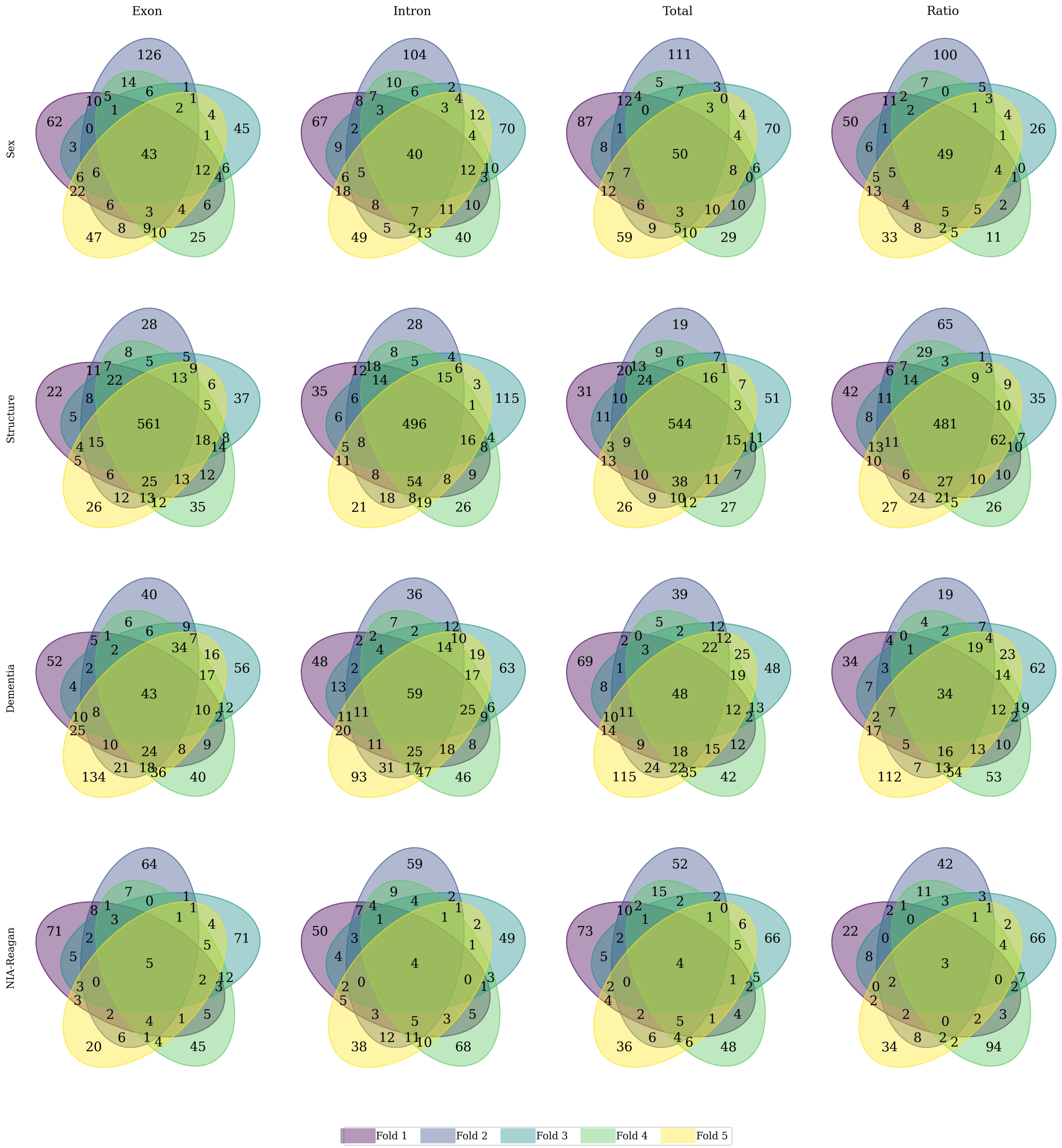
Supplemental Figure 4: Grid of Venn diagrams of the overlap in genes identified by performing differential gene expression (DGE) analysis for exonic, intronic, and total read counts, and the count ratios. The Venn diagram illustrates the overlap between the different count sets. Genes are selected by taking the 1000 most differentially expressed genes per cross-validation fold and subsequently taking the intersection of the cross-validation folds (Section Differential Gene Expression Analysis Gene Selection). The p-values are corrected for multiple testing using the False Discovery Rate (FDR) procedure. The rows of this grid indicate the different labels, and the columns are the different test folds. Note that the area of the overlap is not relative to the number inside it.



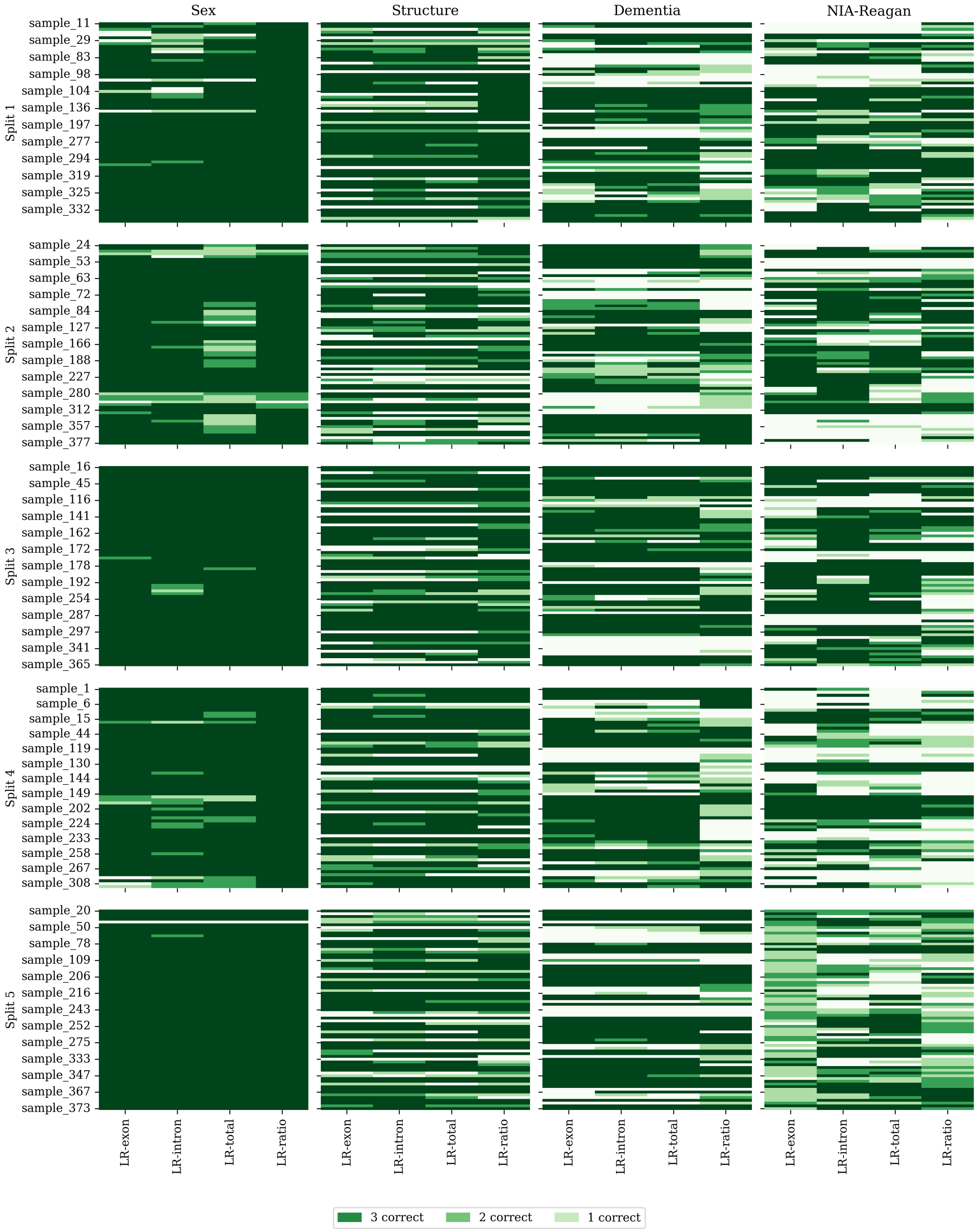
Supplemental Figure 5: Venn diagrams of the overlap of the genes in the different count sets, obtained by taking the intersection (int.) over the test folds. The number of genes for each count set has a corresponding ellipse in Supplemental Figure 7.



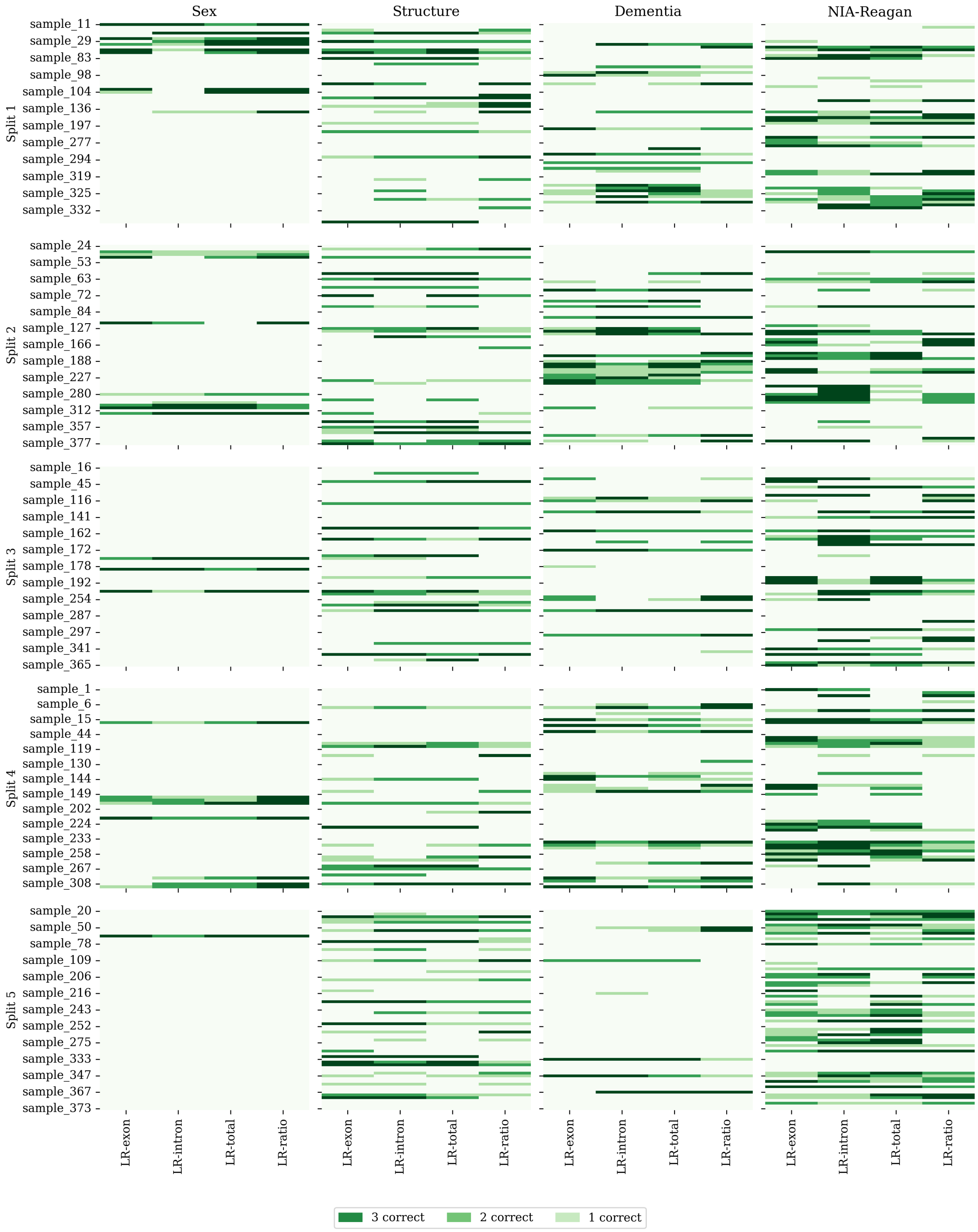
Supplemental Figure 6: Grid of bar charts of the number of genes selected by DGE analysis per label, test fold and data set. The grid rows indicate the prediction labels and the grid columns indicate the data sets. Genes are selected by taking the 1000 most differentially expressed genes per cross-validation fold and subsequently taking the intersection of the cross-validation folds (Section Differential Gene Expression Analysis Gene Selection). The p-values are corrected for multiple testing using the False Discovery Rate (FDR) procedure.



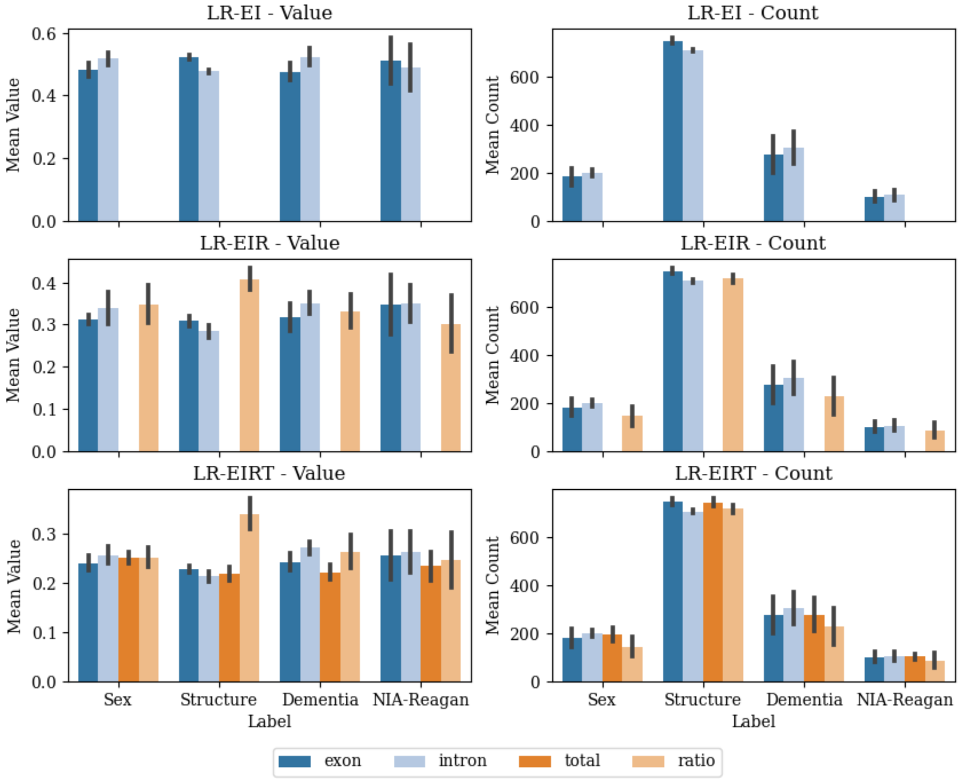
Supplemental Figure 7: Grid of Venn diagrams of the overlap in genes identified by performing differential gene expression (DGE) analysis for exonic, intronic, and total read counts, and the count ratios. The Venn diagram illustrates the overlap in genes between the different test folds. Genes are selected by taking the 1000 most differentially expressed genes per cross-validation fold and subsequently taking the intersection of the cross-validation folds (Section Differential Gene Expression Analysis Gene Selection). The p-values are corrected for multiple testing using the False Discovery Rate (FDR) procedure. The rows of this grid indicate the different labels and the columns indicate the different count sets. Note that the area of the overlap is not relative to the number inside it.



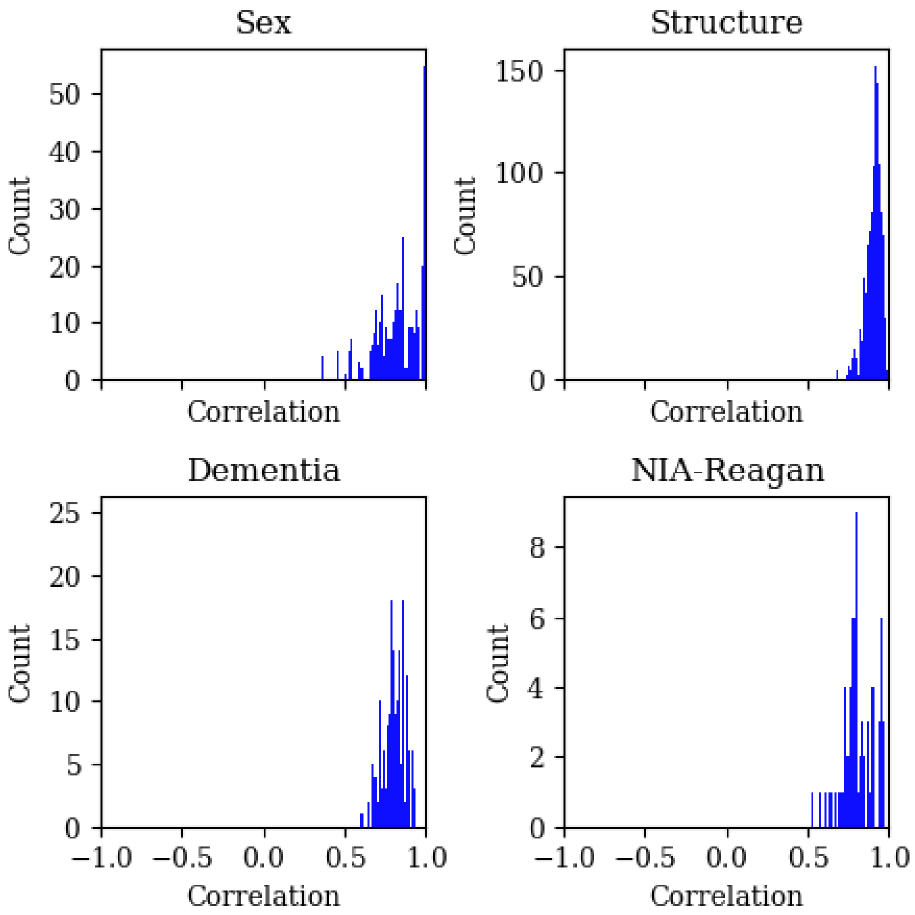
Supplemental Figure 8: Grid of Heatmaps illustrating correctly predicted samples. The rows of the heatmap grid correspond to the test folds. The columns of the heatmap grid correspond to the prediction label. Each model is fitted three times using the three validation folds. Three correct predictions is indicated by the darkest green colour, the slightly less green is twice correct and the fadest green is once correct. Three wrong predictions is portrayed in white.



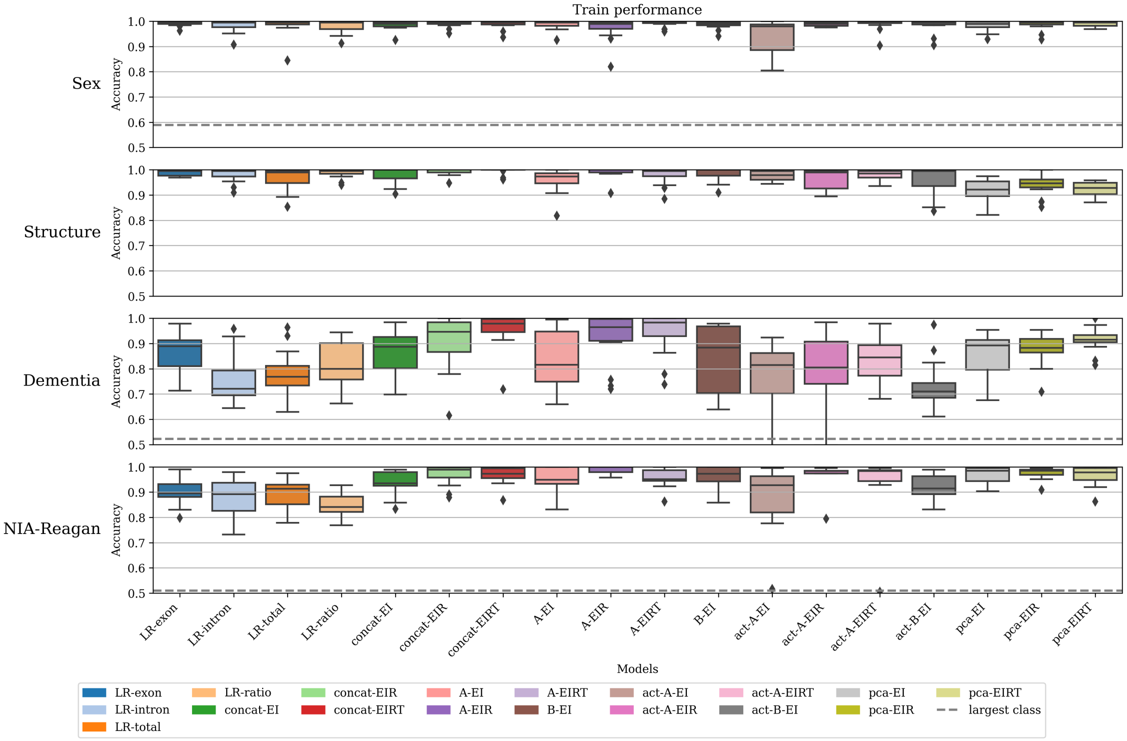
Supplemental Figure 9: Grid of Heatmaps illustrating correctly predicted samples. This figure contains the same data as Supplemental Figure 8, but in this figure, every total agreement between the models is made white. This visualizes where the models are in disagreement.



Supplemental Figure 10: Bar charts of relative feature importance of the feature concatenation logistic regression models. The rows depict the different models, exonic and intronic counts concatenation (LR-EI), exonic and intronic counts and the count ratios concatenation (LR-EIR) and exonic, intronic, and total counts and the count ratios concatenation (LR-EIRT). The left column of the charts is the relative feature importance, split per data set. More elaborately, the weights are obtained by taking the sum of the absolute normalized weights. Subsequently, the weights are split in accordance with their originating count sets. The right column is the same approach but shows the number of genes instead of the weight values. Thus, the relative feature importance in the left column originates from the number of genes in the right column. The height of the bar is the mean over the test folds and the error bars show the standard deviation.



Supplemental Figure 11: Histogram of Pearson correlation coefficients of the correlations between each gene's exonic and intronic read counts. The correlation coefficients in this plot are from genes found for both exonic and intronic reads during the DGE analysis. Corresponding to the intersection of the Venn diagram of the exon and intron ellipses in Supplemental Figure 4. The correlation coefficients of the different folds are aggregated into a single histogram.



Supplemental Figure 12: Boxplot of train accuracies per prediction label. Train accuracies of the models in Figure 10.