RNA ALTERNATIVE SPLICING PREDICTION WITH DISCRETE COMPOSITIONAL ENERGY NETWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

A single gene can encode for different protein versions through a process called alternative splicing. Since proteins play major roles in cellular functions, aberrant splicing profiles can result in a variety of diseases, including cancers. Alternative splicing is determined by the gene's primary sequence and other regulatory factors such as RNA-binding protein levels. With these as input, we formulate the prediction of RNA splicing as a regression task and build a new training dataset (CAPD) to benchmark learned models. We propose discrete compositional energy network (DCEN) which leverages the compositionality of key components to approach this task. In the case of alternative splicing prediction, DCEN models mRNA transcript probabilities through its constituent splice junctions' energy values. These transcript probabilities are subsequently mapped to relative abundance values of key nucleotides and trained with ground-truth experimental measurements. Through our experiments on CAPD, we show that DCEN outperforms baselines and its ablation variants.[1]

1 INTRODUCTION

RNA plays a key role in the human body and other organisms as a precursor of proteins. RNA alternative splicing (AS) is a process where a single gene may encode for more than one protein isoforms (or mRNA transcripts) by removing selected regions in the initial pre-mRNA sequence. In the human genome, up to 94% of genes undergo alternative splicing (Wang et al., 2008). AS not only serves as a regulatory mechanism for controlling levels of protein isoforms suitable for different tissue types but is also responsible for many biological states involved in disease (Tazi et al., 2009), cell development and differentiation (Gallego-Paez et al., 2017). While advances in RNA-sequencing technologies (Bryant et al., 2012) have made quantification of AS in patients' tissues more accessible, an AS prediction model will alleviate the burden from experimental RNA profiling and open doors for more scalable in-silico studies of alternatively spliced genes.

Previous studies on AS prediction mostly either approach it as a classification task or study a subset of AS scenarios. Training models that can predict strengths of AS in a continuous range and are applicable for all AS cases across the genome would allow wider applications in studying AS and factors affecting this important biological mechanism. To this end, we propose AS prediction as a regression task and curate Context Augmented Psi Dataset (CAPD) to benchmark learned models. CAPD is constructed using high-quality transcript counts from the ARCHS4 database (Lachmann et al., 2018). The data in CAPD encompass genes from all 23 pairs of human chromosomes and include 14 tissue types. In this regression task, given inputs such as the gene sequence and an array of tissue-wide RNA regulatory factors, a model would predict, for key positions on the gene sequence, their relative abundance (ψ) in the mRNAs found in a patient's tissue.

Each mRNA transcript may contain one or more splice junctions, locations where splicing occurred in the gene sequence. We hypothesize that a model design which considers these splice junctions would be key for good AS prediction. More specifically, these splice junctions are produced through a series of molecular processes during splicing, so modeling the energy involved at each junction may offer an avenue to model the whole AS process. This is the intuitive behind our proposed discrete compositional energy network (DCEN) which models the energy of each splice junction,

[1] Source code and dataset will be made available.

composes candidate mRNA transcripts' energy values through their constituent splice junctions and predicts the transcript probabilities. The final component maps transcript probabilities to each known exon start and end's ψ values. Apart from better empirical performance over baselines in our experiments on CAPD, DCEN is also interpretable through its transcript probabilities and energy values. DCEN is trained end-to-end in our experiments with ground-truth ψ labels. Though DCEN is evaluated on AS prediction here, it can potentially be used for other applications where compositionality of objects (e.g., splice junctions/transcripts) applies. All in all, the prime contributions of our paper are as follows:

- We construct Context Augmented Psi Dataset (CAPD) to serve as a benchmark for machine learning models in alternative splicing (AS) prediction.
- To predict AS outcomes, we propose discrete compositional energy network (DCEN) to output ψ by modeling transcript probabilities and energy levels through their constituent splice junctions.
- Through experiments on CAPD, we show that DCEN outperforms baselines and ablation variants in AS outcome prediction, generalizing to genes from withheld chromosomes and of much larger lengths.

2 Background: RNA Alternative Splicing

RNA alternative splicing (AS) is a process where a single gene (DNA / pre-mRNA) can encode for multiple mRNAs, and consequently proteins, increasing the biodiversity of proteins encoded by the human genome. Pre-mRNAs contain two kinds of nucleotide segments, introns and exons. Each post-splicing mRNA transcript would only have a subset of exons while the introns and remaining exons are removed. A molecular machine called spliceosome joins the upstream exon's end with the downstream exon's start nucleotides to form a splice junction and removes the intronic segment between these two sites. In an example of an exon-skipping AS event in Figure 1, a single pre-mRNA molecule can be spliced into more than one possible mRNA transcripts (T_α, T_β) with different probabilities ($P_{T_\alpha}, P_{T_\beta} \in [0, 1]$). These probabilities are largely determined by local features surrounding the splice sites (exon starts/ends) such as the presence of key motifs on the exonic and intronic regions surrounding the splice sites nucleotide. Global contextual regulatory factors such as RNA-binding proteins and small molecular signals (Witten & Ule, 2011; Boo & Kim, 2020; Taladriz-Sender et al., 2019) can also influence the transcript probabilities, creating variability for AS outcomes in cells from different tissue types or patients. While exon-skipping is the most common form of AS, there are others such as alternative exon start/end positions and intron retention.

Measurement of Alternative Splicing Outcome One standard way to quantify AS outcome for a group of cells is through **percent spliced-in** ($\psi \in [0, 1]$). Essentially, ψ is defined as the ratio of relative abundance of an exon over all mRNA products. An exon with $\psi = 1$ means that it is included in all mRNAs found from experimental RNA-sequencing measurements while $\psi = 0$ means the exon is missing. ψ can also be annotated onto exon's key positions such as its start and end locations. This allows one to approach the AS prediction as a regression task of predicting ψ for each nucleotide of interest.

3 Related Work

We review prior art on RNA splicing prediction and energy-based models, highlighting those most similar to our work.

Splice Site Classification The earliest task of machine learning on RNA splicing involves classification of splicing sites such as exon start and end positions in a given gene sequence, first using models such as decision trees (Pertea et al., 2001) and support vector machines (Degroeve et al., 2005). As deep learning gains wider adoption, a line of works uses neural networks for splice site prediction from raw sequence (Zuallaert et al., 2018; Zhang et al., 2018; Louadi et al., 2019; Jaganathan et al., 2019). In a recent example, Jaganathan et al. (2019) used a 1-D Resnet model to classify individual nucleotides in a pre-mRNA sequence into 3 categories: 1) intron's start, 2) intron's end or 3) none of the two classes. Unlike these models that only classify splice sites, we

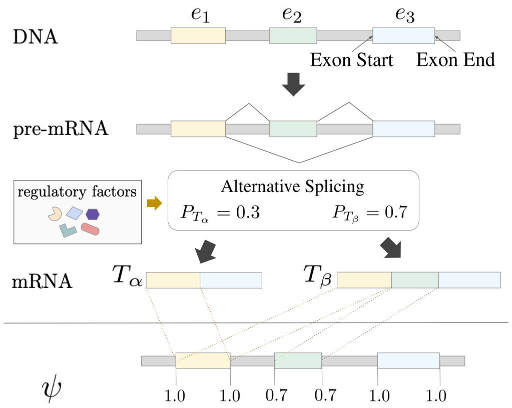

Figure 1: Mechanism of alternative splicing and its relationship with ψ annotations. Introns in the gene sequences are colored gray while exons (e_1, e_2, e_3) are colored otherwise.

propose DCEN to predict ψ levels of splice sites which involves the consideration of patient-specific input such as levels of RNA regulatory factors on top of just primary gene sequences.

Alternative Splicing Prediction The prior work in alternative splicing prediction can be categorized into two distinct groups. The first group framed the prediction as a classification task, whether an alternative splicing event would occur given input or change in input. The earliest examples involved using a Bayesian regression (Barash et al., 2010) and Bayesian neural network (Xiong et al., 2011) to predict whether an exon would be skipped or included in a transcript. Leung et al. (2014) used a neural network with dense layers to predict the type of AS event. Another deep learning-based approach (Louadi et al., 2019) utilized a CNN-based framework to classify between four AS event classes (exon skipping, alternative 3', alternative 5' or constitutive exon).

The second group, which includes DCEN, addresses the prediction as a regression rather than a classification task (Xiong et al., 2015). This formulation gives higher resolution in the AS event since predicted values correlate with the strength of the AS outcome. Bretschneider et al. (2018) proposed a deep learning model to predict which site is most likely to be spliced given the raw sequence input of 80-nt around the site. This approach is similar to the 2000-nt SpliceAI baseline which is outperformed by DCEN in our experiments. Since cellular signals such as RNA-binding proteins (RBPs) are observed to affect RNA splicing (Witten & Ule, 2011; Yee et al., 2019), models such as Huang & Sanguinetti (2017); Zhang et al. (2019) have emerged to incorporate both primary sequence features and RBP levels to better predict exon inclusion levels given a small number of experimental read counts. While also considering regulatory factors such as RBPs, our approach differs from Witten & Ule (2011); Yee et al. (2019) as we do not assume the availability of experimental read counts for the gene of interest. To the best of our knowledge, DCEN is the first approach to model whole transcript constructs (through energy levels) on top of the immediate neighborhood around the nucleotide of interest when predicting its ψ in the splicing process.

Energy-Based Models Most recent work in energy-based models (EBM) (LeCun et al., 2006) focused on the application of image generative modeling. Neural networks were trained to assign low energy to real samples (Xie et al., 2016; Song & Ou, 2018; Du & Mordatch, 2019; Nijkamp et al., 2019; Grathwohl et al., 2019) so that realistic-looking samples can be sampled from the low-energy regions of the EBM's energy landscape. Instead of synthesizing new samples, our goal here is to predict RNA splicing outcomes. Other applications of EBMs include anomaly detection (Song & Ou, 2018), protein conformation prediction (Du et al., 2020) and reinforcement learning (Haarnoja et al., 2017). Previous compositional EBMs such as Haarnoja et al. (2017); Du et al. (2019) considered high dimensional continuous spaces in their applications which makes sampling from the model intractable. In contrast, since genes consist of a finite number of known transcripts, DCEN considers discrete space where the probabilities of the transcripts are tractable through importance sampling with a uniform distribution.

4 CAPD: Context Augmented Psi Dataset

The core aim of Context Augmented Psi Dataset (CAPD) is to frame the alternative splicing prediction as a regression task to facilitate benchmarking of machine learning models. Each CAPD sample is a unique AS profile of a gene from the cells of a particular tissue type, from an individual patient. Its annotations contain $\psi \in [0, 1]$ of all the know exon starts and ends for the particular gene. Apart from the ψ labels, each data sample also contains the following as inputs: a) full sequence of the gene ($\mathbf{x}$), b) nucleotide positions of all the known transcripts ($\mathcal{T}$) on the full gene sequence and c) levels of RNA-regulatory factors ($\mathbf{x}_{\text{reg}}$).

Construction of CAPD We mine transcript abundance data from the publicly available ARCHS4 database (Lachmann et al., 2018). ARCHS4 database contains expression data for 84863 publicly available human RNA-seq samples that were mined from Gene Expression Omnibus (GEO) and aligned to human transcriptome Ensembl 90 cDNA (Zerbino, 2018) to produce count numbers for each transcript in each sample. In this initial CAPD version, $250 \times N_T$ tissue-type specific samples were collected, where N_T is tissue type, selected in accordance with Illumina Human Body Map 2.0 Project (GEO GSE30611): adipose tissue, blood, brain, breast, colon, heart, kidney, liver, lung, lymph node, prostate gland, skeletal muscle tissue, testes, thyroid gland. Standard normalization of RNA read counts (Mortazavi et al., 2008) follows and outlier removal is conducted to remove variabilities that may be due to technical artifacts rather than biological variability (Hicks et al., 2018), with more details in Appendix § A.3. This gives the normalized transcript count values (c_{T_α}) for each gene transcript (T_α).

To construct the levels of RNA-regulatory factors ($\mathbf{x}_{\text{reg}}$), we retrieve splicing-affecting RBPs from the RBPDB database (Cook et al., 2011) and RNA chemically modifying proteins from Basturea (2013). This results in a list of 3792 transcripts (206 genes) of RNA-binding proteins from the RBPDB database and 206 transcripts (20 genes) of RNA chemically modifying proteins. After removing overlapping transcripts from the two lists, this gives a 3971-dimensional $\mathbf{x}_{\text{reg}}$ where its values are extracted from the same normalized transcript count values above.

To generate gene-specific primary pre-mRNA sequences, we follow the procedure described in Jaganathan et al. (2019): pre-mRNA (synonymous to DNA, with T → U) sequences are extracted with flanking ends of 1000 nt from each side, while intergenic sequences are discarded. Pseudogenes, genes with sequence assembly gaps and genes with paralogs are excluded from the data. Exon coordinate information is retrieved from GENCODE Release 26 for GRCh38 (Frankish et al., 2019) comprehensive set. We omit genes with missing matching GENCODE ID, resulting in a total of 19399 unique human gene sequences. The coordinates of the gene pre-mRNA sequence start and end are determined by the left- or right-most position among all the transcripts for that gene, further extended with flanking ends of 1000 nt for context on each side. The CAPD dataset is split into train and test according to the chromosome number and length of the genes. All genes in chromosome 1, 3, 5, 7, and 9 are withheld as test samples (Test-Chr), similar to Jaganathan et al. (2019). To test that models can generalize to genes of longer lengths, we further withhold all genes from the other chromosomes that have >100K nt (Test-Long) and group them in the test set. The remaining genes are used for training. Key statistics of the CAPD is summarized in Table 1.

Label Annotation The ψ labels are constructed as follows: 1) the count values c_i for all known exon start and end (position i) are initialized to zero. 2) enumerating through all transcripts, each transcript count is added onto the counts of its constituent exons' start/end $\mathbf{c}_i \leftarrow \mathbf{c}_i + c_{T_m}, \forall i \in T_m$. 3) To compute ψ values, each count value is divided by the sum of transcript counts to normalize its value to $[0, 1]$, i.e., $\psi_i = \mathbf{c}_i / \sum_m c_{T_m}$.

Table 1: CAPD data statistics.

	Train	Test-Chr	Test-Long
# of unique genes	11,472	5,604	2,323
mean pre-mRNA length (nt)	26,196	73,355	247,041
mean # of exons	6.7	7.0	10.5

5 DCEN: Discrete Compositional Energy Network

In § 2, we learn that final mRNA splice isoforms may comprise of one or more splice junctions, points where upstream exon's end and downstream exon's start meet. Inspired by the creation of splice junctions by spliceosomes, the first stage of DCEN models the energy values of the splicing process at splice junctions. As the splice junctions are typically far enough (∼300 nt) and assumed to be independent of one another, the energy of a final mRNA transcript can be composed by the summation of its constituent splice junctions' energy. The second stage of DCEN derives probabilities for the formation of each transcript from their energy values. These transcript probabilities are then mapped to the relative abundance (ψ) of exon starts/ends at its corresponding splice sites. Since a particular splice junction may appear in more than one mRNA isoforms, we design DCEN to be invariant to the splice junctions. In the following § 5.1, we discuss the key components of DCEN while § 5.2 details its training process.

5.1 Model Architecture

Here, we detail the DCEN learned energy functions and how transcript probabilities can be derived from their energy levels through Boltzmann distribution and importance sampling. A summary of DCEN model architecture is shown in Figure 2.

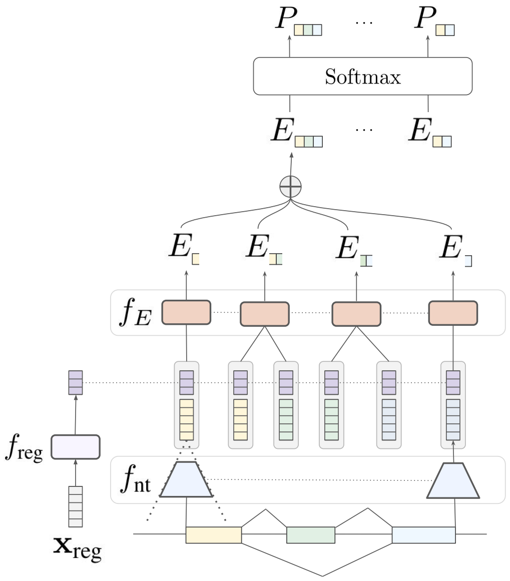

Figure 2: Model architecture of proposed discrete compositional energy network (DCEN).

Learned Energy Functions The weights of DCEN's learned energy functions consist of a 1) feature extractor, 2) regulatory factors encoder and 3) junction energy network. The feature extractor (f_{nt}) takes the pre-mRNA sequence as its input ($\mathbf{x} \in \mathbb{R}^{l \times 4}$) and outputs a hidden representation ($\mathbf{h}_{\text{nt}} \in \mathbb{R}^{l \times d}$) for each nucleotide position ($\mathbf{x}_i$) while the regulatory factors encoder (f_{reg}) takes in the levels of regulatory factors ($\mathbf{x}_{\text{reg}}$) to compute a gene-wide hidden states $\mathbf{h}_{\text{reg}} \in \mathbb{R}^d$:

$$\mathbf{h}_{\text{nt}} = f_{\text{nt}}(\mathbf{x}) \; , \quad \mathbf{h}_{\text{reg}} = f_{\text{reg}}(\mathbf{x}_{\text{reg}}) \tag{1}$$

We concatenate $\mathbf{h}_{\text{reg}}$ to all the position-specific $\mathbf{h}_{\text{nt}}$ to form a new position-wise hidden state ($\mathbf{h}_i$) that is dependent on regulatory factors. The representation of a particular splice junction ($J_k = (i, j)$) is the concatenation between the hidden states of upstream exon end's and downstream exon start's hidden states:

$$\mathbf{h}_{J_k} = [\mathbf{h}_i ; \mathbf{h}_j] \; , \quad \mathbf{h}_i = [\mathbf{h}_{\text{nt}i} ; \mathbf{h}_{\text{reg}}] \tag{2}$$

If an exon start/end is the first/last nucleotide of a transcript, its hidden state is concatenated with a learned start/end token instead ($\mathbf{h}_{\text{start}}$ or $\mathbf{h}_{\text{end}}$ respectively). To model the energy ($E_{J_k} \in \mathbb{R}$) of producing splice junction J_k, we feed its representation into the energy network (f_E). We sum up the energy values of all splice junctions (J_k) inside a mRNA transcript (T_α) to compose the total energy ($E_{T_\alpha} \in \mathbb{R}$) involved in producing the transcript from a splicing event:

$$E_{T_\alpha} = \sum_k E_{J_k} \quad , \quad \forall J_k \in T_\alpha \quad , \quad E_{J_k} = f_E(\mathbf{h}_{J_k}) \tag{3}$$

Transcript Probabilities from Energy Values After obtaining the energy levels (E_{T_α}) of all mRNA transcript candidates for a particular gene, we can compute the probabilities of these transcripts via a softmax operation through the theorem below.

Theorem 5.1. *If the energy levels of all the possible discrete states of a system are known, the probability of a particular state T_i is the softmax output of its energy E_{T_i} with respect to those of all other possible states in the system, i.e.,*

$$P_i = \frac{\exp(-E_{T_i})}{\sum_j \exp(-E_{T_j})} = \text{Softmax}_i(E) \tag{4}$$

Its proof, deferred to the Appendix § A.1, can be derived through Boltzmann distribution and importance sampling. Since each mRNA transcript T_α can be interpreted as a discrete state of the alternative splicing event for a particular gene (system) in Theorem 5.1, we can compute its probability from its energy value. It is important to also consider a null state with energy E_{null} where none of the gene's mRNA transcripts is produced. In DCEN, E_{null} is a learned parameter. In summary, the probability of producing transcript T_α in an gene splicing event is

$$P_{T_\alpha} = \text{Softmax}_\alpha(E) \quad , \quad E = [E_{T_\alpha}, E_{T_\beta}, \ldots, E_{\text{null}}] \tag{5}$$

Exon Start/End Inclusion Levels from Transcript Probabilities We can compute the probability of a particular (exon start/end) nucleotide of position i by summing up the probabilities of all transcripts that contain that nucleotide.

$$P_i = \sum_m P_{T_m} \quad , \quad \forall T_m \in \mathcal{T}_i \tag{6}$$

where $\mathcal{T}_i$ is the set of transcripts containing the nucleotide of interest.

5.2 Training Algorithm

Regression Loss Since experimental PSI levels from CAPD are essentially the empirical observations of nucleotide present in the final mRNA transcript products, its normalized values ($y_i \in [0, 1]$) can be used as the ground-truth label for the predicted nucleotide probability. This allows us to train DCEN as a regression task by minimizing the mean squared error (MSE) between the predicted nucleotide probability and the normalized experimental PSI values:

$$L_\psi = \sum_i \|\tilde{y}_i - y_i\|_2^2 \quad , \quad \tilde{y}_i = P_i \tag{7}$$

In our experiments, only nucleotide positions that are either an exon start or end are involved in this regression training objective.

Classification Loss We also include a classification objective, similar to Jaganathan et al. (2019), where a classification head f_{cls} takes the nucleotide hidden states ($\mathbf{h}_{\text{nt}}$) as input to predict probability of every nucleotide as one of the 3 classes (exon start, end and neither) to give the classification loss:

$$L_{\text{cls}} = -\mathbf{y}_{\text{cls}}^\top \log f_{\text{cls}}(\mathbf{h}_{\text{nt}}) \tag{8}$$

where $\mathbf{y}_{\text{cls}}$ is the ground-truth labels for each nucleotide. This helps the DCEN learn features on the gene primary sequence that are important for RNA splicing. A summary of the training phase is shown in Algorithm 1.

Algorithm 1: Discrete Compositional Energy Network Training

```
Input: Training data $\mathcal{D}_{\text{train}}$, Learning rate $\gamma$,
for each training iteration do
    Sample $(\mathbf{x}, \mathbf{x}_{reg}, \mathbf{y}, \mathcal{T}, \mathcal{J}) \sim \mathcal{D}_{\text{train}}$
    $\mathbf{h}_{\text{nt}} \leftarrow f_{\text{nt}}(\mathbf{x})$,
    $L_{\text{cls}} \leftarrow -\mathbf{y}_{\text{cls}}^{\top} \log f_{\text{cls}}(\mathbf{h}_{\text{nt}})$    ▷ Compute exon start/end classification cross-entropy loss
    $\mathbf{h}_{\text{reg}} \leftarrow f_{\text{reg}}(\mathbf{x}_{reg})$
    $\mathbf{h}_i \leftarrow [\mathbf{h}_{\text{nt}\,i}; \mathbf{h}_{\text{reg}}]$
    $\mathbf{h}_{J_k} \leftarrow [\mathbf{h}_i; \mathbf{h}_j]$ , $J_k = (i, j)$    ▷ Get junction state from upstream exon end $i$ and downstream
      exon start $j$
    $E_{J_k} \leftarrow f_E(\mathbf{h}_{J_k})$    ▷ Compute splice junction energy
    $E_{T_\alpha} \leftarrow \sum_k E_{J_k}$ , $\forall J_k \in T_\alpha, \forall T_\alpha \in \mathcal{T}$    ▷ Compose transcript energy
    $P_{T_\alpha} \leftarrow \text{Softmax}_\alpha(E)$ , $E = [E_{T_\alpha}, E_{T_\beta}, \ldots]$    ▷ Compute transcript probabilities
    $\tilde{y_i} \leftarrow \sum_m P_{T_m}$ , $\forall T_m \in \mathcal{T}_i$
    $L_\psi \leftarrow \sum_i \|\tilde{y_i} - y_i\|_2^2$    ▷ Compute exon start/end inclusion regression loss
    $\theta \leftarrow \theta + \gamma \, \nabla_\theta(\lambda_{\text{reg}} L_\psi + \lambda_{\text{cls}} L_{\text{cls}})$
```

6 Experiments

We evaluate DCEN and baselines on the prediction of the ψ values of exon starts and ends in our new CAPD dataset. In the following, we describe the baseline models and DCEN ablation variants before discussing the experimental setup and how well these models generalize to withheld test samples.

Baselines and Ablation Variants SpliceAI is a 1D convolutional Resnet (He et al., 2016) trained to predict splice sites on pre-mRNA sequences. We train three variants of SpliceAI to compare as baselines in our experiments: the first (**SpliceAI-cls**) is trained only on the classification objective, similar to the original paper, to predict whether a nucleotide is an exon start, end or neither of them. The second (**SpliceAI-reg**) is trained only on a regression objective like DCEN to directly predict the PSI levels of nucleotides while the third variant (**SpliceAI-cls+reg**) is trained on both the classification and regression objectives. Both SpliceAI-reg and SpliceAI-cls+reg also have a regulatory factors encoder (f_{reg}) similar to DCEN's to compute $\mathbf{h}_i$ and a regression head (f_ψ) to output PSI level for each nucleotide. For a direct comparison, DCEN's feature extractor f_{nt} takes the same architecture as the SpliceAI Resnet. Two DCEN ablation variants are also evaluated: The **Junction-psi** model predicts the psi levels of a particular splice junction directly rather than its energy level in the case of DCEN. A simpler ablation variant (**SpliceAI-ML** or SpliceAI-match layers) substitutes DCEN's f_E with a position-wise feedforward MLP containing the same number of parameters to verify that DCEN's better performance is not due to more learned parameters.

Data & Models We use the CAPD dataset (§ 4) for the training and evaluation of all models. 10% of the CAPD training genes are randomly selected as validation set for early-stopping while the rest are used as training samples for the models. For DCEN's f_{nt} and the SpliceAI baselines, we follow the same setup as the SpliceAI-2K model in Jaganathan et al. (2019) which is a Resnet made up of 1-D convolutional layers with a perceptive window of 2K nucleotides, 1K on each flanking sides. The SpliceAI Resnet model has a total of 12 residual units and hidden states of size 32. The number of channels in $\mathbf{h}_{\text{reg}}$ and $\mathbf{h}_{\text{nt}}$ are 32 while $\mathbf{h}$ has a size of 64 channels. We use a 3-layer MLP for f_{reg}. The regression head f_ψ in SpliceAI-reg and SpliceAI-cls+reg is a 3-layer MLP with a sigmoid activation to outputs a scalar PSI value. DCEN's f_E is a 4-layer MLP and outputs a scalar energy value for each splice junction. Intermediate hidden states of f_{reg} and f_E all have dimension of 32.

Training & Evaluation All models are trained with Adam optimizer with a learning rate of 0.001 in our experiments. Due to the data's large size, the training is early-stopped when the model's validation performances plateau: less than 25% of the full train dataset for all models in our experiments. For the training of SpliceAI models, samples are fed into the model with a batch size of 8 sequences with a maximum length of 7K nucleotides (5K labeled + 2K flanking). For training of DCEN and its ablation variants, a pretrained SpliceAI-cls&reg model was used as the weights of f_{reg}, f_{nt} and only the parameters of f_E is trained to reduce training time. In each training iteration of DCEN and its ablation variants, a batch of 16 genes was used to train the weights. We evaluate all the models on withheld test samples with two standard regression metrics: Spearman rank correlation and Pearson correlation. Pearson correlation measures the linear relationship between the ground-truth and predicted exon start/end inclusion levels while Spearman rank correlation is based on the ranked order of the prediction and ground-truth values. The models are evaluated on data of 70 patients in total (5 for each of 14 tissue types).

Results The DCEN outperforms all baselines and ablation variants for both regression metrics when evaluated on the withheld test samples, as shown in Table 2. Even with same number of parameters, the DCEN shows better performance than the SpliceAI-ML and Junction-psi model. Even with more trained parameters, Junction-psi model performs worse than the SpliceAI-reg and SpliceAI-cls+reg baselines while the SpliceAI-ML does not show clear improvement over these two baselines. These observations indicate that DCEN's design to compose transcripts' energy through splice junctions and infer their probabilities from energy values is key for better prediction. When evaluated separately test samples from chromosomes (1, 3, 5, 7, and 9) not seen during the training phase, DCEN maintains its performance (Table 4 in Appendix), showing that it generalizes across novel gene sequences.

Table 2: Performance of DCEN and baselines on withheld test samples.

Model	Spearman Rank Correlation	Pearson Correlation
SpliceAI-cls	0.402	0.353
SpliceAI-reg	0.595	0.586
SpliceAI-cls+reg	0.594	0.584
SpliceAI-ML	0.590	0.589
Junction-psi	0.579	0.523
DCEN (ours)	**0.640**	**0.666**

Long gene sequences DCEN was trained only on genes sequences of length less than 100K nucleotides. From Table 3, we observe that DCEN still outperforms the other baselines by a substantial margin and retains most of its performance on these samples when evaluated on genes with long sequences (>100K nucleotides). Compared to shorter introns in genes of shorter length, splicing of pre-mRNA with very large introns was observed to occur in a more nested and sequential manner (Sibley et al., 2015; Suzuki et al., 2013). Since genes with long sequences contain more large introns, this difference in the splicing mechanism may explain the slight drop in DCEN's performance on genes with longer sequences.

Table 3: Performance of DCEN and baselines on Test-Long, withheld samples with long gene sequences (>100K nucleotides).

Model	Spearman Rank Correlation	Pearson Correlation
SpliceAI-cls	0.403	0.349
SpliceAI-reg	0.603	0.579
SpliceAI-cls+reg	0.601	0.576
SpliceAI-ML	0.598	0.582
Junction-psi	0.582	0.510
DCEN (ours)	**0.646**	**0.657**

7 Conclusions

We curate CAPD to benchmark learning models on alternative splicing (AS) prediction as a regression task, to facilitate future work in this key biological process. By exploiting the compositionality of discrete components, we propose DCEN to predict the AS outcome by modeling mRNA transcripts' probabilities through its constituent splice junctions' energy levels. Through our experiments on CAPD, we show that DCEN outperforms baselines and other ablation variants in predicting AS outcomes. Our work shows that deconstructing a task into a hierarchy of discrete components can improve performance in learning models while improving interpretability.

REFERENCES

Yoseph Barash, John A. Calarco, Weijun Gao, Qun Pan, Xinchen Wang, Ofer Shai, Benjamin J. Blencowe, and Brendan J. Frey. Deciphering the splicing code. *Nature*, 2010. ISSN 00280836. doi: 10.1038/nature09000.

Georgeta N. Basturea. Research Methods for Detection and Quantitation of RNA Modifications. *Materials and Methods*, 2013. doi: 10.13070/mm.en.3.186.

Sung Ho Boo and Yoon Ki Kim. The emerging role of RNA modifications in the regulation of mRNA stability, 2020. ISSN 20926413.

Hannes Bretschneider, Shreshth Gandhi, Amit G. Deshwar, Khalid Zuberi, and Brendan J. Frey. COSSMO: Predicting competitive alternative splice site selection using deep learning. In *Bioinformatics*, 2018. doi: 10.1093/bioinformatics/bty244.

Douglas W. Bryant, Henry D. Priest, and Todd C. Mockler. Detection and quantification of alternative splicing variants using RNA-seq. *Methods in Molecular Biology*, 2012. ISSN 10643745. doi: 10.1007/978-1-61779-839-9_7.

Kate B. Cook, Hilal Kazan, Khalid Zuberi, Quaid Morris, and Timothy R. Hughes. RBPDB: A database of RNA-binding specificities. *Nucleic Acids Research*, 2011. ISSN 03051048. doi: 10.1093/nar/gkq1069.

Sven Degroeve, Yvan Saeys, Bernard De Baets, Pierre Rouzé, and Yves Van de Peer. SpliceMachine: Predicting splice sites from high-dimensional local context representations. *Bioinformatics*, 2005. ISSN 13674803. doi: 10.1093/bioinformatics/bti166.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. 2019.

Yilun Du, Joshua Meier, Jerry Ma, Rob Fergus, and Alexander Rives. Energy-based models for atomic-resolution protein conformations. *arXiv preprint arXiv:2004.13167*, 2020.

Adam Frankish, Mark Diekhans, Anne Maud Ferreira, Rory Johnson, Irwin Jungreis, Jane Loveland, Jonathan M. Mudge, Cristina Sisu, James Wright, Joel Armstrong, If Barnes, Andrew Berry, Alexandra Bignell, Silvia Carbonell Sala, Jacqueline Chrast, Fiona Cunningham, Tomás Di Domenico, Sarah Donaldson, Ian T. Fiddes, Carlos García Girón, Jose Manuel Gonzalez, Tiago Grego, Matthew Hardy, Thibaut Hourlier, Toby Hunt, Osagie G. Izuogu, Julien Lagarde, Fergal J. Martin, Laura Martínez, Shamika Mohanan, Paul Muir, Fabio C.P. Navarro, Anne Parker, Baikang Pei, Fernando Pozo, Magali Ruffier, Bianca M. Schmitt, Eloise Stapleton, Marie Marthe Suner, Irina Sycheva, Barbara Uszczynska-Ratajczak, Jinuri Xu, Andrew Yates, Daniel Zerbino, Yan Zhang, Bronwen Aken, Jyoti S. Choudhary, Mark Gerstein, Roderic Guigó, Tim J.P. Hubbard, Manolis Kellis, Benedict Paten, Alexandre Reymond, Michael L. Tress, and Paul Flicek. GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Research*, 2019. ISSN 13624962. doi: 10.1093/nar/gky955.

L. M. Gallego-Paez, M. C. Bordone, A. C. Leote, N. Saraiva-Agostinho, M. Ascensão-Ferreira, and N. L. Barbosa-Morais. Alternative splicing: the pledge, the turn, and the prestige: The key role of alternative splicing in human biological systems, 2017. ISSN 14321203.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Stephanie C. Hicks, F. William Townes, Mingxiang Teng, and Rafael A. Irizarry. Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics*, 2018. ISSN 14684357. doi: 10.1093/biostatistics/kxx053.

Yuanhua Huang and Guido Sanguinetti. BRIE: Transcriptome-wide splicing quantification in single cells. *Genome Biology*, 2017. ISSN 1474760X. doi: 10.1186/s13059-017-1248-5.

Kishore Jaganathan, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F. McRae, Siavash Fazel Darbandi, David Knowles, Yang I. Li, Jack A. Kosmicki, Juan Arbelaez, Wenwu Cui, Grace B. Schwartz, Eric D. Chow, Efstathios Kanterakis, Hong Gao, Amirali Kia, Serafim Batzoglou, Stephan J. Sanders, and Kyle Kai How Farh. Predicting Splicing from Primary Sequence with Deep Learning. *Cell*, 2019. ISSN 10974172. doi: 10.1016/j.cell.2018.12.015.

Alexander Lachmann, Denis Torre, Alexandra B. Keenan, Kathleen M. Jagodnik, Hoyjin J. Lee, Lily Wang, Moshe C. Silverstein, and Avi Ma'ayan. Massive mining of publicly available RNA-seq data from human and mouse. *Nature Communications*, 2018. ISSN 20411723. doi: 10.1038/s41467-018-03751-6.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Michael K.K. Leung, Hui Yuan Xiong, Leo J. Lee, and Brendan J. Frey. Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 2014. ISSN 14602059. doi: 10.1093/bioinformatics/btu277.

Zakaria Louadi, Mhaned Oubounyt, Hilal Tayara, and Kil To Chong. Deep splicing code: Classifying alternative splicing events using deep learning. *Genes*, 2019. ISSN 20734425. doi: 10.3390/genes10080587.

Ali Mortazavi, Brian A. Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 2008. ISSN 15487091. doi: 10.1038/nmeth.1226.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Advances in Neural Information Processing Systems*, pp. 5232–5242, 2019.

Michael C. Oldham, Genevieve Konopka, Kazuya Iwamoto, Peter Langfelder, Tadafumi Kato, Steve Horvath, and Daniel H. Geschwind. Functional organization of the transcriptome in human brain. *Nature Neuroscience*, 2008. ISSN 10976256. doi: 10.1038/nn.2207.

M. Pertea, X. Lin, and S. L. Salzberg. GeneSplicer: A new computational method for splice site prediction. *Nucleic Acids Research*, 2001. ISSN 03051048. doi: 10.1093/nar/29.5.1185.

Christopher R. Sibley, Warren Emmett, Lorea Blazquez, Ana Faro, Nejc Haberman, Michael Briese, Daniah Trabzuni, Mina Ryten, Michael E. Weale, John Hardy, Miha Modic, Tomaž Curk, Stephen W. Wilson, Vincent Plagnol, and Jernej Ule. Recursive splicing in long vertebrate genes. *Nature*, 2015. ISSN 14764687. doi: 10.1038/nature14466.

Yunfu Song and Zhijian Ou. Learning neural random fields with inclusive auxiliary generators. *arXiv preprint arXiv:1806.00271*, 2018.

Hitoshi Suzuki, Toshiki Kameyama, Kenji Ohe, Toshifumi Tsukahara, and Akila Mayeda. Nested introns in an intron: Evidence of multi-step splicing in a large intron of the human dystrophin pre-mRNA. *FEBS Letters*, 2013. ISSN 00145793. doi: 10.1016/j.febslet.2013.01.057.

Andrea Taladriz-Sender, Emma Campbell, and Glenn A. Burley. Splice-switching small molecules: A new therapeutic approach to modulate gene expression. *Methods*, 2019. ISSN 10959130. doi: 10.1016/j.ymeth.2019.06.011.

Jamal Tazi, Nadia Bakkour, and Stefan Stamm. Alternative splicing and disease, 2009. ISSN 09254439.

Eric T. Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F. Kingsmore, Gary P. Schroth, and Christopher B. Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 2008. ISSN 00280836. doi: 10.1038/nature07509.

Joshua T. Witten and Jernej Ule. Understanding splicing regulation through RNA splicing maps, 2011. ISSN 01689525.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pp. 2635–2644, 2016.

Hui Y. Xiong, Babak Alipanahi, Leo J. Lee, Hannes Bretschneider, Daniele Merico, Ryan K.C. Yuen, Yimin Hua, Serge Gueroussov, Hamed S. Najafabadi, Timothy R. Hughes, Quaid Morris, Yoseph Barash, Adrian R. Krainer, Nebojsa Jojic, Stephen W. Scherer, Benjamin J. Blencowe, and Brendan J. Frey. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 2015. ISSN 10959203. doi: 10.1126/science.1254806.

Hui Yuan Xiong, Yoseph Barash, and Brendan J. Frey. Bayesian prediction of tissue-regulated splicing using RNA sequence and cellular context. *Bioinformatics*, 2011. ISSN 13674803. doi: 10.1093/bioinformatics/btr444.

Brian A. Yee, Gabriel A. Pratt, Brenton R. Graveley, Eric L. van Nostrand, and Gene W. Yeo. RBP-Maps enables robust generation of splicing regulatory maps. *RNA*, 2019. ISSN 14699001. doi: 10.1261/rna.069237.118.

Daniel R. et al. Zerbino. Ensembl 2018. *Nucleic Acids Research*, 2018. ISSN 13624962. doi: 10.1093/nar/gkx1098.

Yi Zhang, Xinan Liu, James MacLeod, and Jinze Liu. Discerning novel splice junctions derived from RNA-seq alignment: A deep learning approach. *BMC Genomics*, 2018. ISSN 14712164. doi: 10.1186/s12864-018-5350-1.

Zijun Zhang, Zhicheng Pan, Yi Ying, Zhijie Xie, Samir Adhikari, John Phillips, Russ P. Carstens, Douglas L. Black, Yingnian Wu, and Yi Xing. Deep-learning augmented RNA-seq analysis of transcript splicing. *Nature Methods*, 2019. ISSN 15487105. doi: 10.1038/s41592-019-0351-9.

Jasper Zuallaert, Fréderic Godin, Mijung Kim, Arne Soete, Yvan Saeys, and Wesley De Neve. Splicerover: Interpretable convolutional neural networks for improved splice site prediction. *Bioinformatics*, 2018. ISSN 14602059. doi: 10.1093/bioinformatics/bty497.

A Appendix

A.1 Proof

Theorem A.1. *If the energy levels of all the possible discrete states of a system is known, the probability of a particular state T_i is the softmax output of its energy E_{T_i} with respect to those of all other possible states in the system, i.e.,*

$$P_i = \frac{\exp(-E_{T_i})}{\sum_j \exp(-E_{T_j})} = \mathrm{Softmax}_i(E) \tag{9}$$

Proof. From Boltzmann distribution, the probability that a system takes on a particular state (x) can be expressed as:

$$\begin{aligned} p_\theta(x) &= \frac{\exp\left(-E_\theta(x)\right)}{Z(\theta)} \\ &= \frac{h(x)}{Z(\theta)} \end{aligned} \tag{10}$$

where

$$\begin{aligned} Z(\theta) &= \int \exp\left(-E_\theta(x)\right) dx \\ &= \int h(x)\, dx \end{aligned} \tag{11}$$

is known as the partition function.

Since the probabilities of all possible states sum to 1, we have

$$1 = \mathbb{E}_{x \sim p_\theta}[1] = \sum_x \frac{h(x)}{Z} \tag{12}$$

which gives

$$Z = \sum_x h(x)\ . \tag{13}$$

Through importance sampling with another probability distribution q, we can express Z as

$$\begin{aligned} Z &= \sum_x \frac{h(x)}{q(x)} q(x) \\ &= \frac{1}{n} \sum_i \frac{h(x_i)}{q(x_i)} \quad , \quad x_i \sim q\ . \end{aligned} \tag{14}$$

Using an uniform discrete distribution as q where all k possible states (x_i) have the same probability $q(x_i) = (1/k)$, we get

$$\begin{aligned} Z &= \frac{1}{k} \sum_i \frac{h(x_i)}{(1/k)} \\ &= \sum_i h(x_i) \end{aligned} \tag{15}$$

Combining Eq. (11) and (15), this gives

$$\begin{aligned} p_\theta(x_i) &= \frac{h(x_i)}{\sum_j h(x_j)} \\ &= \frac{\exp\left(-E_\theta(x_i)\right)}{\sum_j \exp\left(-E_\theta(x_j)\right)} \\ &= \mathrm{Softmax}_i(E) \end{aligned} \tag{16}$$

□

A.2 Additional Results

Table 4: Performance of DCEN and baselines on Test-Chr, test samples from chromosomes (1, 3, 5, 7, and 9) different from the training set.

Model	Spearman Rank Correlation	Pearson Correlation
SpliceAI-cls	0.403	0.357
SpliceAI-reg	0.592	0.596
SpliceAI-cls+reg	0.593	0.594
SpliceAI-ML	0.587	0.599
Junction-psi	0.578	0.536
DCEN (ours)	**0.638**	**0.678**

A.3 CAPD Sample Batch Effect Removal, Clustering and Outliers

For our model to learn better, we try to homogenize the samples retrieved from ARCHS4 database: enough to correct for samples and variability that might confuse the model, but careful enough to not wash out the biological variability.

As samples from Gene Expression Omnibus (GEO) processed by ARCHS4 are uploaded voluntarily by different research groups working on different research problems and are not always perfectly annotated (ARCHS4 database contains all the metadata with sample description, experimental conditions etc. – all the information provided by the authors of the study), there might be some samples representing unique and outlying studies from which the model might not learn well.

Therefore, in addition to batch effect correction, we perform an outlier removal procedure. Significant outliers are removed from the batch using a simple z-score process described in (Oldham et al., 2008) for similar tasks. Pairwise correlation between RPM-normalised counts is calculated using Pearson's correlation and the mean of all these correlations is found (μ_{all}). Sample i with average distance from μ_{all} larger than an arbitrarily set threshold is removed:

$$d_i = \frac{\mu_i - \mu_{all}}{\sigma_{all}}; \quad h = -2; \quad \text{if } d_i < h \to \text{removed}$$

Here we provide some illustrations on the homogeneity of the data with 100 kidney tissue samples used as an example.

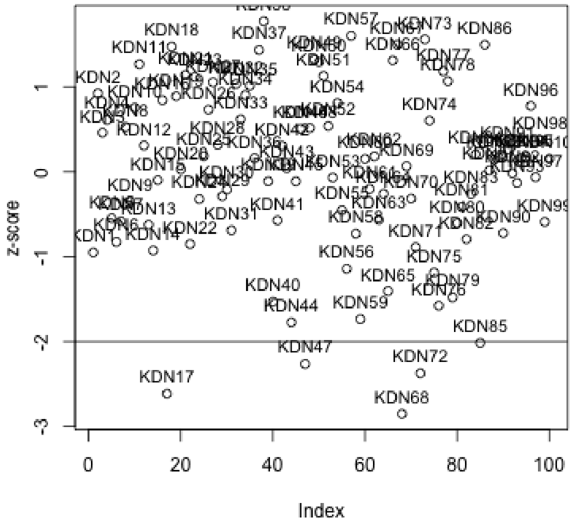

Figure 3: Outlier plot for 100 kidney tissue samples, randomly retrieved from ARCHS4 database. x-axis is the sample index, y-axis is the z-score of the sample. Samples KDN17, KDN47, KDN68, KDN72, KDN85 will be removed from the downstream analyses.

Samples with the smallest overall correlations with other samples in Fig. 4 are likely to be removed by the outlier detection procedure. The homogeneity of samples belonging to one tissue type is varying, but usually, only $\approx 2{-}5\%$ of the sample population is considered outlying by the algorithm.

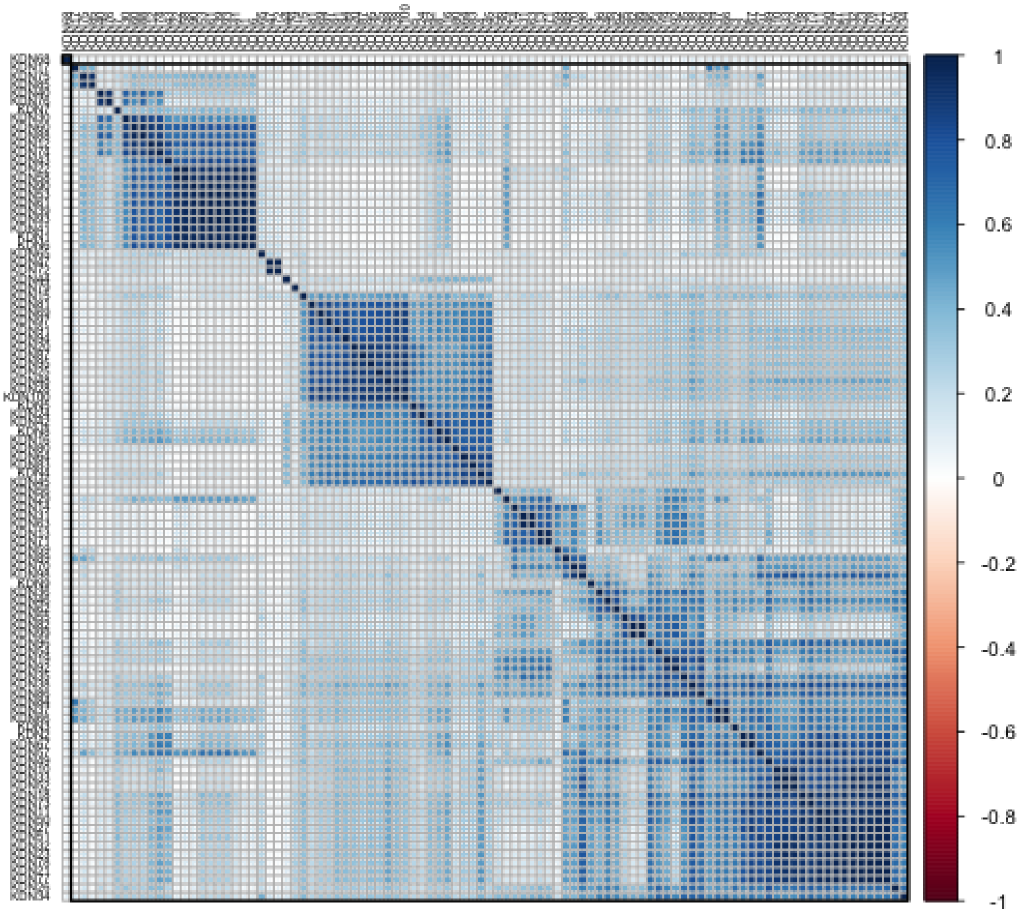

Figure 4: Correlation plot for 100 kidney tissue samples, randomly retrieved from ARCHS4 database. Correlation is calculated with the Pearson formula. Clustering for the plot is performed using R hierarchical clustering.