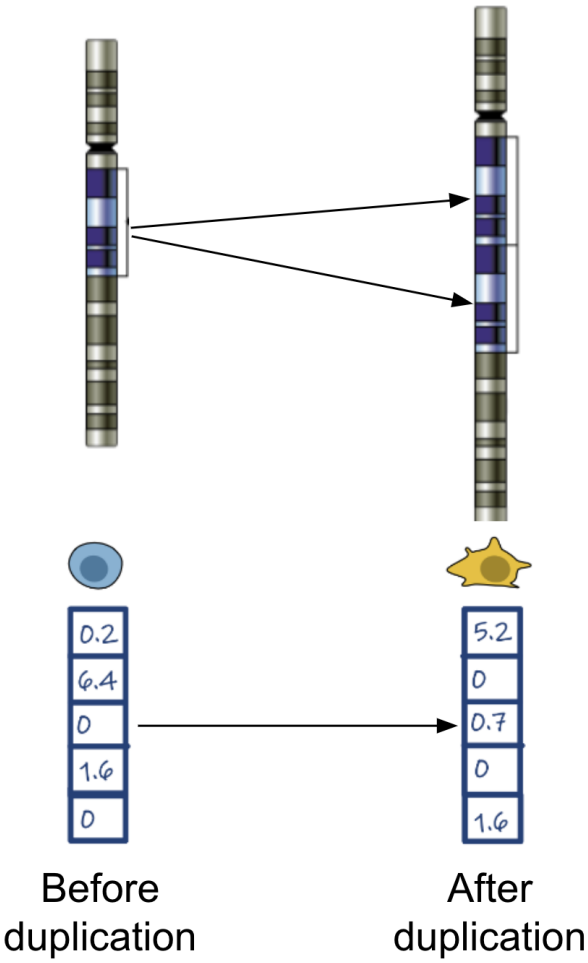
# Finding the Missing Link Between Copy Number Aberrations and Gene Expression in Cancer




Research in Computer Science Project
Semester: Fall 2022
Student: Arda Arslan
Main Supervisor: Josephine Yates
Co-Supervisors: Florian Barkmann, David Wissel
Supervising Professor: Prof. Dr. Valentina Boeva

ETH zürich

BOEVA LAB

# Abstract

Copy number aberrations (CNA) can regulate gene expression (GEX) of a cell, and this can result in several diseases including cancer. The relationship between CNA and GEX can be complex since some genes are not dosage-sensitive, some of them can be affected by compensatory mechanisms, and the molecular context of the cell can have an impact on the change in the gene expression levels. Moreover, both the estimation of GEX using CNA information and identifying CNA-GEX links are hard problems due to the large dimensionality of CNA and GEX vectors of a tumor sample compared to the number of samples in publicly available datasets. Previous studies tried to explain the effect of CNAs on GEX, however, the models which are claimed to perform well were neither interpretable nor explainable due to the dimensionality reduction techniques used. Besides, CNA-GEX links found in these works were not comprehensive. The main objectives of this work are to provide a model which can predict the gene expression of a tumor sample given its copy number aberration information in an interpretable or explainable way.

# Contents

# Introduction

Copy number aberrations (CNAs) are gains or losses of contiguous parts of a chromosome [Wu et al. 2014]. CNAs can lead to modifications in the gene expression (GEX) levels of a cell, and this means Z Score levels of the cell are altered as well. By developing a predictive model that can estimate Z Score based on CNAs, medical professionals can use CNA information of a tumor sample to classify this sample as benign or malignant, determine the subtype of the cancer, and choose the best treatment option for the patient. Another important information one can extract is CNA-Z Score links. This information can be useful for understanding the biological mechanisms that contribute to the development and progression of cancer. Both predicting Z Score from copy number aberrations (CNAs) and identifying the connections between CNAs and Z Scores are challenging due to the high dimensionality of the tumor sample's CNA and Z Score vectors, compared to the limited number of samples available in public datasets. One should also consider the correlation in the Z Scores of genes and the fact that genes whose expression values are correlated are affected by CNA of the same set of genes.

The code for our work is available at https://github.com/ardarslan/cna2zscore.

# Related Work

## 2.1 Correlation Analysis between Copy Number Aberration and Gene Expression

Previous work reported high correlation between CNA and GEX. The following works do not investigate how well GEX can be predicted using CNA information, they also do not report any trans link between a gene's CNA and another gene's GEX. They only assess the correlation between CNA and GEX with slightly different methodologies.

[Cheng et al. 2012] worked on explaining the effect of CNA on GEX using gastric cancer samples. They first identify the genes with at least 1.3 times difference in the number of copies between gastric cancer samples and matched noncancerous samples, and this results in a set of 163 genes. Out of these 163 genes, they select the genes whose CNA and GEX values have a Pearson correlation that is larger than or equal to 0.40, and this results in a set of 133 genes. The authors report the median Pearson correlation coefficient for these 133 genes as 0.69.

[Shao et al. 2019] worked on the same problem using all (31) cancer types. They reported that the Pearson's correlation coefficient between the median Z Score and thresholded CNA values for most of the genes in The Cancer Genome Atlas (TCGA) Pan-Cancer dataset ranges between 0.8 and 1.0. They used Z Score in their analysis instead of GEX since they worked with Pan-Cancer data.

[Alonso et al. 2017] studied the relationship between CNA and GEX using colorectal cancer samples. The authors define minimum recurrent regions (MRR) as regions with CNA that is observed in at least 5% of individuals. They estimate the quantity of stroma in each sample using the gene expression values of these samples. Then they used this estimated stroma information to adjust the calculated partial correlation between CNAs on MRRs and GEX values. They do

the analysis in two settings:

- Cis Analysis: The authors found that approximately 37% of the genes that were in CNA regions had a change in their GEX levels and they reported that the average partial Pearson correlation coefficient between CNA and GEX was 0.54 (after adjusting for stromal content of each sample).

- Trans Analysis: The authors analyzed the trans links between 14654 genes' expression and 13279 MRRs, which corresponds to more than 15 million links. They found that only 191 of these links were significant, which were between 42 unique genes and 168 unique MRRs. One interesting result is that 105 out of the 168 MRRs did not contain genes. These 105 MRRs probably include enhancer and promoter regions, which are generally not taken into account in other works.

## 2.2 Estimating Gene Expression from Copy Number Aberration

### 2.2.1 Non-Interpretable Approaches

[Seal et al. 2020] proposed a model which encodes concatenated 23604 CNA and 18996 DNA Methylation features into a smaller dimension (200) using a Deep Denoising Auto-Encoder (DDAE). After training DDAE, these 200 features from the bottleneck layer are fed into an MLP to predict GEX. The authors showed that their approach performed better than the other state-of-the-art models using liver hepatocellular carcinoma cancer samples. The problem with this model is that it is not intrinsically interpretable and we cannot use a post-hoc explainability method to understand the relationship between each pair of input and output variables. The authors provide plots for predicted and ground truth expression values for 100 randomly selected genes averaged over all samples per gene. These plots are misleading since they only show that the bias in the prediction errors is low, however, they do not tell anything about whether the model can capture the variance in the labels or not.

### 2.2.2 Interpretable Approaches

[Kim and Xing 2009] proposed TreeLasso (or Tree-guided Group Lasso), which they defined as a special case of overlapping group lasso where the overlaps of groups are selected by fitting a hierarchical agglomerative clustering tree on the response variables.

[Cheung et al. 2005], [Stranger et al. 2006] and [Zhu et al. 2008] claim that genes in the same biological pathway are usually co-regulated or co-expressed. [Kim and Xing 2009] add that the genes in the same biological pathway can be affected by the same set of relevant covariates.

When multiple independent Lassos per response variable are used to estimate the response variables using the covariates, it is assumed that the response variables are independent of each other, and the covariates that affect a gene's expression value are selected on a per-gene basis. However, none of these assumptions hold.

Using multi-task Lasso, one can solve the first problem, but not the second.

If we use L1/L2 regularization (group-lasso regularization) instead of L1 regularization in the multi-task Lasso, then we assume that all of the response variables are affected by the same set of relevant covariates, however, this is not the case.

TreeLasso can both capture the correlation in the response variables (due to the multi-task setting) and also allows correlated response variables to be affected by the same set of covariates.

The first step of TreeLasso is to fit a hierarchical agglomerative clustering tree to the response variables. This tree can be used to group the response variables in a hierarchical setting. Then we minimize the following loss function:

$$\frac{1}{2}\|\mathbf{Y} - \mathbf{XB}\|_F^2 + \lambda \sum_j \sum_{v \in V} w_v \|\boldsymbol{\beta}_j^{G_v}\|_2$$

The shape of the matrices are as follows:

Y: (N, P)

X: (N, Q)

B: (Q, P)

where N is the number of samples, Q is the number of covariates, and P is the number of response variables.

In the loss function, $\lambda$ is a coefficient that determines the importance of the whole regularization term.

j under the summation operator represents the order of the covariate that we are regularizing the parameters of.

v under the summation operator represents the internal node in the tree whose descendant leaf nodes correspond to the response variables that we are regularizing the parameters of.

$w_v$ determines the importance of regularizing the parameters of the response variables which are descendant leaf nodes of the internal node v in the tree. These are taken as input from the user.

$\|\boldsymbol{\beta}_j^{G_v}\|_2$ is the L1-L2 loss term. Assuming that there are two response variables, the long version of this term is as follows:

$$w_{v_1}|\beta_1^j| + w_{v_2}|\beta_2^j| + w_{v_3}\left(\sqrt{(\beta_1^j)^2 + (\beta_2^j)^2}\right)$$

Here, we apply L1 and L2 regularization to the parameters which connect the jth covariate to the response variables.

To minimize the loss function in Tree-Lasso, the authors use the smoothing proximal gradient (SPG) method [Chen et al. 2012] that was implemented to minimize a convex loss function like

the overlapping group lasso.

[Dutta et al. 2021] used Sparse Canonical Correlation Analysis (sCCA) to identify orthogonal gene expression modules that are correlated with CNA modules. The main motivation for using sCCA for this problem is similar to the one in [Kim and Xing 2009]. In this work, the authors try to exploit the fact that there are groups of genes such that the expression of each gene in a group is affected by a certain group of genes' CNA values. Assuming we have n samples and p CNA regions, G is a matrix with shape nxp where $G_{ij}$ corresponds to the number of insertion or deletions for sample i at CNA region j. E is a matrix with shape nxg where $E_{ij}$ corresponds to the normalized gene expression level of sample i for gene j. The authors iteratively find 14 components of CNA and GEX variables using sCCA. At iteration k, they find a sparse vector $u_k$ whose nonzero elements correspond to the genes whose CNAs are selected, and a sparse vector $v_k$ whose nonzero elements correspond to the genes whose GEXs are selected. To find $u_k$ and $v_k$, the authors suggest solving the following optimization problem:

$$(u_k, v_k) = argmax\ v_k^T E^T G u_k$$

$$s.t.\ ||u_k||_1 \leq c_u$$

$$||v_k||_1 \leq c_v$$

$$||u_k||_2 = 1$$

$$||v_k||_2 = 1$$

$u_k$ is orthogonal to $u_i$ for i=1, ..., k-1

By solving these optimization problems for k=1, ..., 14, sCCA selects 824 genes, whose expression levels are regulated by 1851 CNA sites overall.

## 2.3 Validation of the Identified Cis and Trans Regulated Genes

When the time we write this report, we could not find an experimental study that introduces a CNA for a certain gene to measure the change in the expression level of the genes.

We concluded that there are three ways to validate cis and trans regulated genes after they are identified:

- [Dutta et al. 2021] worked on METABRIC dataset [Curtis et al. 2012] [Pereira et al. 2016] to identify cis and trans links between CNA and GEX. Then they validated these links using the TCGA dataset. They took the gene modules and the corresponding CNA sets that were determined from the METABRIC dataset, and only kept the ones that also existed in the TCGA dataset as well. Then they checked the average correlation for these components using TCGA dataset. They found that for all 14 components, the average correlation was significantly high (with a p-value less than 0.05). The authors state that this is an expected result since the components mostly consisted of cis interactions. However, in our case, this validation is still applicable since we can check whether a gene that is cis-regulated in METABRIC dataset is also cis-regulated in TCGA dataset or not, and we can also check whether a trans-link identified in

METABRIC dataset also exists in TCGA dataset or not.

- [Kim and Xing 2009] uses a simulated dataset to test the algorithm they propose. A similar simulated dataset can be used to compare performance of different algorithms, and then the best performing one can be used on a real CNA-Z Score dataset. The simulated dataset, and the predictions made by different algorithms are provided below (image and the caption are taken from [Kim and Xing 2009]):

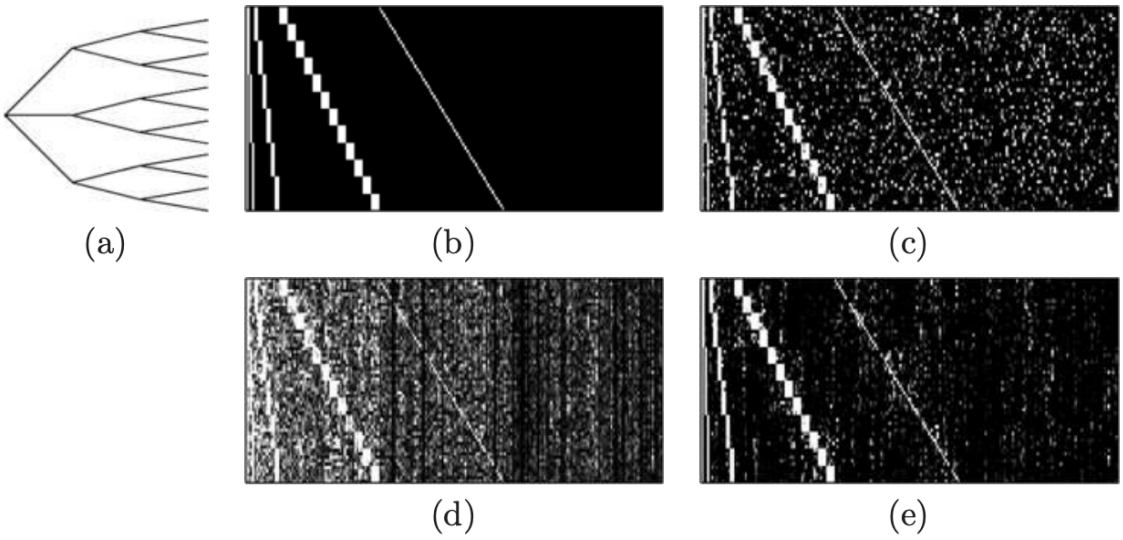



FIG. 3. *An example of regression coefficients estimated from a simulated data set.* (a)*: Hierarchical clustering tree of four levels over responses. Only the top three levels are shown to avoid clutter.* (b)*: True regression coefficients. Estimated parameters are shown for* (c)*: lasso,* (d)*: $L_1/L_2$-regularized multli-task regression, and* (e)*: tree lasso. The rows represent responses and the columns covariates.*

- [Srihari et al. 2016] used the cis and trans-regulated genes to cluster breast cancer samples. Then they report how much these clusters match with the breast cancer subtype clusters reported by [Curtis et al. 2012] using adjusted rand index (ARI) metric (0.69 in their case). The authors report that the genes which contribute to formation of each cancer subtype are distinct. The authors also plot Kaplan-Meier plots for each of the clusters, and they report that the log-rank test p-value for the Kaplan-Meier estimates is significantly low ($p < 0.0001$).

# Dataset

## 3.1 Data Sources

Copy number aberration (CNA) data can be found from here [Goldman et al. 2020].

Reverse phase protein array (RPPA) data can be found from here [Goldman et al. 2020].

Gene expression (GEX) data can be found from here [Goldman et al. 2020].

Tumor purity data can be found from here.

Cancer type data can be found from here.

To calculate Z-Scores, we first group GEX data by cancer type, then for each group, and for each gene in this group, we subtract the mean GEX, and divide the result by standard deviation (plus an epsilon=1e-10) of GEX values for this group and for this gene.

We concatenate one-hot encoded cancer type columns to the input data only if the dataset has samples from all cancer types.

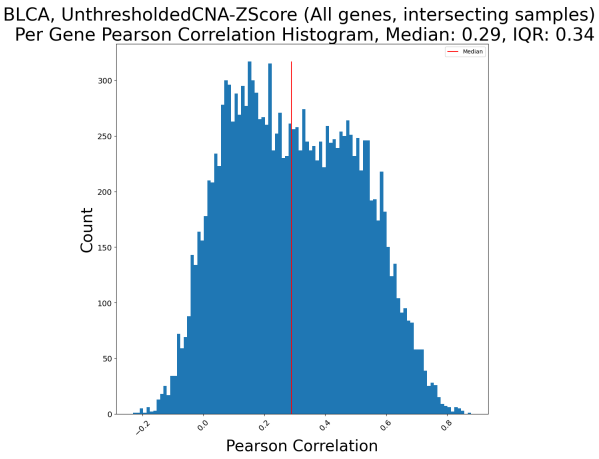
## 3.2 Exploratory Data Analysis

The number of BLCA samples and the number of all samples in each dataset can be found below:

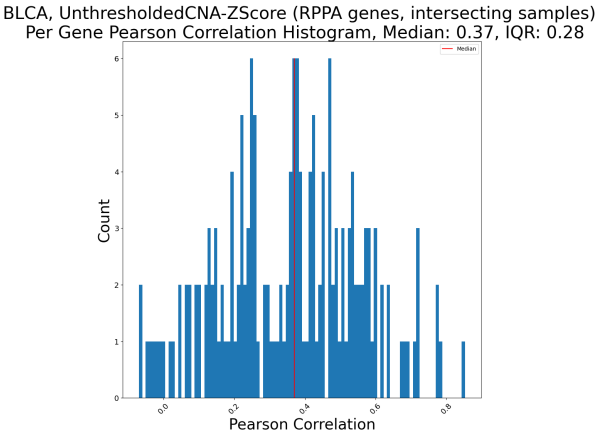
| Dataset | Number of BLCA Samples | Number of All Samples |
|---|---|---|
| Unthresholded CNA + Purity -> ZScore | 403 | 9673 |
| RPPA -> ZScore | 335 | 6857 |

To understand the nature of our data, we plot per gene input-output correlation histograms using BLCA cancer samples. The output is always ZScore.

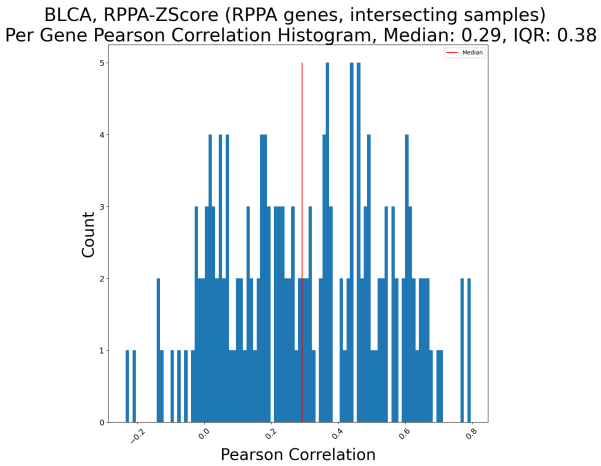
- When the input is unthresholded CNA (all genes):




- When the input is unthresholded CNA (RPPA genes):




- When the input is RPPA (RPPA genes):


BLCA, RPPA-ZScore (RPPA genes, intersecting samples)
Per Gene Pearson Correlation Histogram, Median: 0.29, IQR: 0.38
Median
Count
Pearson Correlation
0
1
2
3
4
5
−0.2
0.0
0.2
0.4
0.6
0.8

# Implementation Details

## 4.1 Deep Learning Models

We use PyTorch 1.10.0 [Paszke et al. 2019] as the deep learning framework. The following deep learning-based models are implemented:

- DLPerGene: This model uses a linear layer per gene each with an input dimension of (number of non-gene columns + 1) and an output dimension of 1. Number of non-gene columns can be one of the following values:

  - 0 if both the tumor purity column and the one-hot encoded cancer type columns are not in the input features.

  - 1 if only the tumor purity column is in the input features.

  - 29 if only the one-hot encoded cancer type columns are in the input features.

  - 30 if both the tumor purity column and the one-hot encoded cancer type columns are in the input features.

- DLGeneEmbeddings: This model trains a "gene_embedding_size" dimensional embedding for each gene, and applies a linear layer to the concatenation of this embedding with non-gene columns. The linear layer is the same for each gene, and has an input dimension of (number of non-gene columns+1) and an output dimension of 1. Number of non-gene columns can take one of the values explained in "DLPerGene" model.

- DLLinearZeroDiagonal: This model trains two tensors. The first tensor has a shape equal to the number of upper diagonal elements of the weight matrix with shape (number of genes, number of genes + number of non-gene columns). Number of non-gene columns

can take one of the values explained in "DLPerGene" model. The second tensor has a shape equal to the number of lower diagonal elements of the weight matrix just explained. During forward pass, these two tensors are used to construct a weight matrix whose diagonal elements are zeros.

- DLMLP: This model trains "num_non_linear_layers" hidden layers, and one output layer, which is linear. Each hidden layer consists of the following operations: Linear mapping, batch normalization, non-linear activation, dropout.
- DLResConMLP: This model has only one difference from the DLMLP model. In this model, there is a residual connection between the input gene columns and output gene columns. There is no residual connection outgoing from non-gene columns.
- DLTransformer: This model is similar to DLGeneEmbeddings model. In this one, we first apply a SelfAttention layer to each gene's gene embedding, and then apply a Layer-Normalization. We concatenate the generated embeddings with non-gene columns, and finally apply a linear layer. The linear layer is again the same for each gene and has an input dimension of (gene embedding size + number of non-gene columns) and an output dimension of 1.
- DLInterpretableMLP: This model predicts the weight and bias matrices of a linear layer by using an MLP. Input dimension of this MLP is the same with number of input columns. Output dimension of this MLP is (number of input columns * number of output columns + number of output columns). This model is able to exploit nonlinear relationships in the input features which was not the case in DLLinear model. During forward pass, after predicting the weight and bias matrices of a linear layer we use these matrices for the linear layer operation. This model is interpretable since we can directly visualize the predicted cis and trans relationships between genes.
- DLPerChromosome: This model creates a model per chromosome. The input dimension of each model is equal to (number of genes in the related chromosome + number of non-gene columns), output dimension of each model is equal to number of genes in the related chromosome. All the models starting with "DL" can be used as the submodels in DLPerChromosome model.

## 4.2 Deep Learning Model Details

In all our deep learning experiments, we use a train-validation-test split with ratios 0.6-0.2-0.2 (stratified by cancer type), Adam optimizer, mean squared error loss function, a maximum gradient norm of 10.0, batch size of 32, early stopping patience of 8, ReduceLROnPlateau scheduler, learning rate scheduler factor of 0.5, learning rate scheduler patience of 4, a minimum learning rate of 2.5e-5, and we set the maximum number of epochs to 200. For these experiments we tune the following hyperparameters with the options listed:

- Dropout: 0.10, 0.25, 0.50
- Number of nonlinear layers (For models except "Linear model" and "Linear model with zero diagonal"): 1, 2

- Hidden dimension options (For models except "Linear model" and "Linear model with zero diagonal"): 0.10, 0.25, 0.50 times the maximum of input dimension and output dimension of the network.
- L1 regularization coefficient for the diagonal parameters of the weight matrix (only for "Linear model"): 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1
- L1 regularization coefficient for the non-diagonal parameters of the weight matrix (only for "Linear model"): 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1
- L2 regularization coefficient for the diagonal parameters of the weight matrix (only for "Linear model"): 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1
- L2 regularization coefficient for the non-diagonal parameters of the weight matrix (only for "Linear model"): 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1
- L1 regularization coefficient for the parameters of the weight matrices (for models except the "Linear model"): 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1
- L2 regularization coefficient for the parameters of the weight matrices (for models except the "Linear model"): 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1
- L1 regularization coefficient for predicted weights (only for "Interpretable MLP model"): 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1
- Initial learning rate: 0.001, 0.01
- Use CNA Adjusted Z Score (only for "Linear model" and "Linear model with zero diagonal"): False, True
- Gene embedding size (only for "Gene embeddings model" and "Transformer model"): 4, 16, 64

## 4.3 Statistical Machine Learning Models

We use scikit-learn [Pedregosa et al. 2011] for the statistical machine learning experiments. The following models are implemented:

- SklearnLinear: This model uses MultiTaskLasso from scikit-learn with a max_iter=5000.
- SklearnPerGene: This model uses separate Lasso (from scikit-learn) models for each gene.
- SklearnPerChromosome: This model creates a model per chromosome. The input dimension of each model is equal to (number of genes in the related chromosome + number of non-gene columns), output dimension of each model is equal to number of genes in the related chromosome. All the models starting with "Sklearn" can be used as the submodels in SklearnPerChromosome model.

## 4.4 Statistical Machine Learning Model Details

In statistical machine learning experiments, we use a fixed test split with size 0.2 of the dataset size. The remaining samples are used as train sets (0.6 of the dataset size) and validation sets (0.2 of the dataset size) in a 5-fold cross-validation fashion. For all splits, we stratify by cancer type. For these experiments, we tune the following hyperparameters with the options listed below:

- L1 regularization coefficient for the parameters of the weight matrices: 0.0, 0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1.
- Per Chromosome: false, true

## 4.5 Interpretability and Explainability

To identify which genes are cis or trans-regulated, we implemented two methods:

- Linear Weight Matrix Analysis: When the model is linear, its weight matrix is intrinsically interpretable. The weight at the ith row and the jth column gives an idea about the contribution of the jth input variable to the ith output variable. There is one problem with this approach: Usually when there is a CNA on a segment of a chromosome, we observe similar CNA values for the genes next to each other. If there are three genes (A, B, C) next to each other, who generally have three copies (instead of two) in average across given tumour samples for a certain cancer type, then each of these genes would have similar CNA values. Assuming B is a cis-regulated gene (with a correlation 1.0 between its CNA and Z Score) and its expression is not affected by changes in other genes' expression values, then normally we would expect to see a weight 1.0 in the weight matrix of our linear model which maps gene B's CNA to gene B's Z Score. However, due to the correlation in CNA values of gene A, B and C, the weight matrix might have the following weights instead:

  - Between CNA of gene A and Z Score of gene B: 0.33

  - Between CNA of gene B and Z Score of gene B: 0.33

  - Between CNA of gene C and Z Score of gene B: 0.33

  However this solution is not correct. To handle this problem, one can do one of the following:

  - Identify MRRs as defined by [Alonso et al. 2017], and try to predict and interpret the relationship between CNAs on these regions and expression of genes using a linear model.

  - Use an approach similar to sCCA [Dutta et al. 2021], which finds highly correlated CNA components and GEX components. The authors report that each CNA component generally consists of genes on the same chromosome and that are close to each other.

  - Use an approach similar to TreeLasso [Kim and Xing 2009], which allow a response variable to be affected by a selected set of covariates.

- Shapley Values [Lundberg and Lee 2017]: When the model is non-linear, Shapley values can be used to post-hoc explain the contribution of each input variable to each output variable.

# Experiments

In all experiments, we trained the models on samples from all cancer types and evaluated the models on samples from the BLCA cancer type.

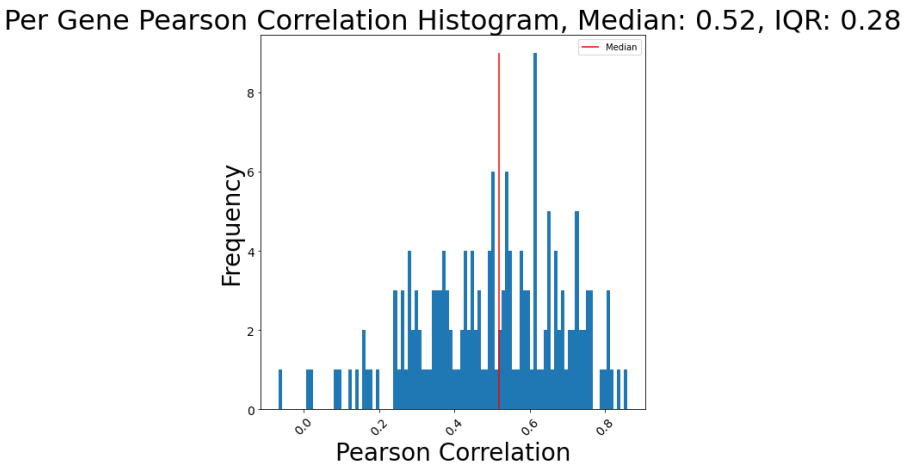
## 5.1 Comparing Informativeness of Different Input Types

Since RPPA data has information for only 168 genes, we compared the informativeness of Unthresholded CNA + Tumour Purity, Thresholded CNA + Tumour Purity, and RPPA using only these genes. Previously we observed that Unthresholded CNA + Tumour Purity is more informative than Thresholded CNA + Tumour Purity while predicting ZScore. Unfortunately, we could not reproduce the results of this experiment due to time and computational resource constraints. We also compared the informativeness of Unthresholded CNA + Tumour Purity and RPPA using the DLLinear model and DLPerGene model on BLCA samples. In these experiments, the regularization options are L1, L2, Dropout, L1 + Dropout, and L2 + Dropout.

- DLLinear Model

  As it can be seen from the plots below, RPPA is more informative than Unthresholded CNA when we use DLLinear model.

  - When RPPA is used as input



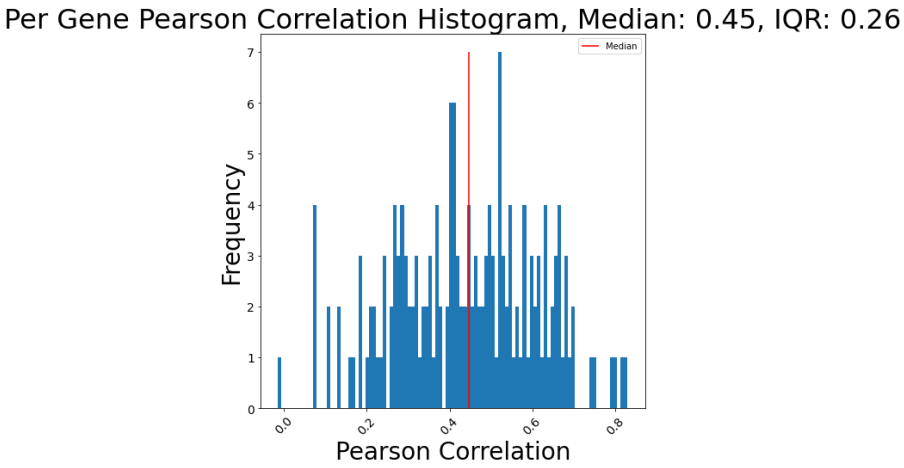

Initial Learning Rate: 0.01

Number of Parameters: 33 432

Regularization: Dropout (0.33)

MSE: 0.87

– When Unthresholded CNA + Tumour Purity is used as input




Initial Learning Rate: 0.01

Number of Parameters: 33 432

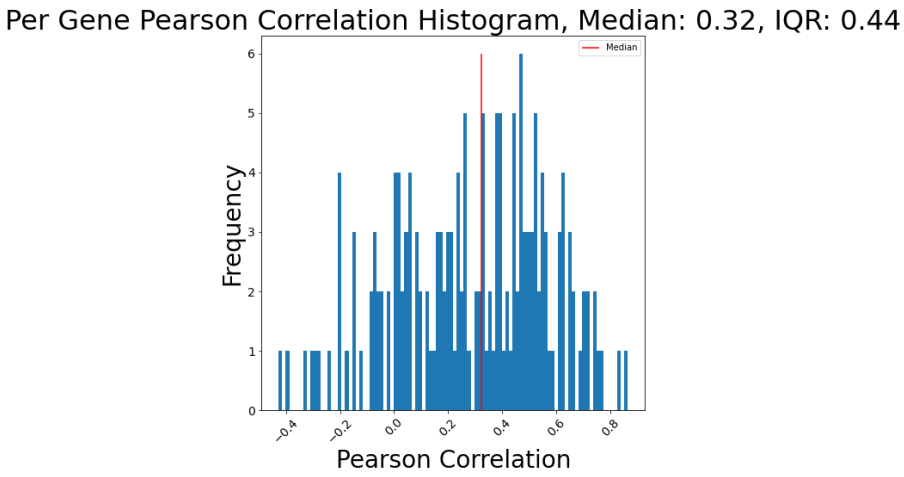
Regularization: L1 Reg (0.0001)

MSE: 1.22

- DLPerGene Model

  When we use DLPerGene model, Unthresholded CNA + Tumour Purity is more informative than RPPA.

  – When RPPA is used as input



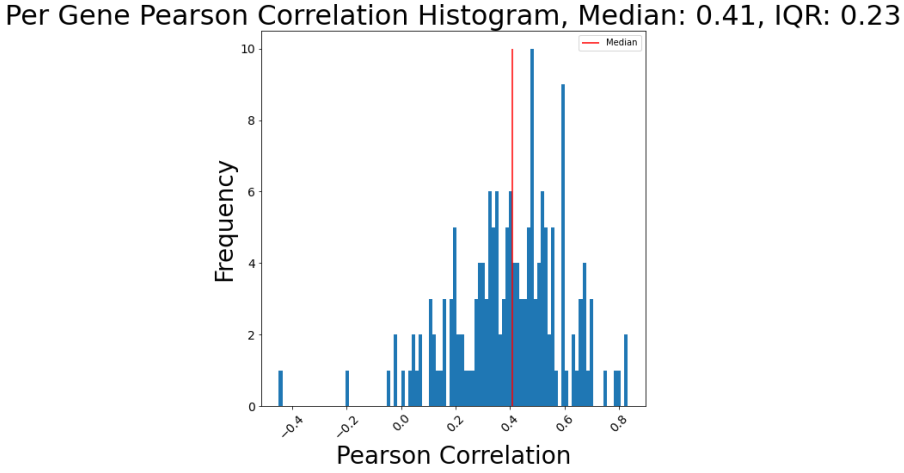

Initial Learning Rate: 0.001

Number of Parameters: 5 208

Regularization: Dropout (0.33) + L1 Reg (0.0001)

MSE: 1.06

– When Unthresholded CNA + Tumour Purity is used as input




Initial Learning Rate: 0.01

Number of Parameters: 5 376

Regularization: Dropout (0.25)

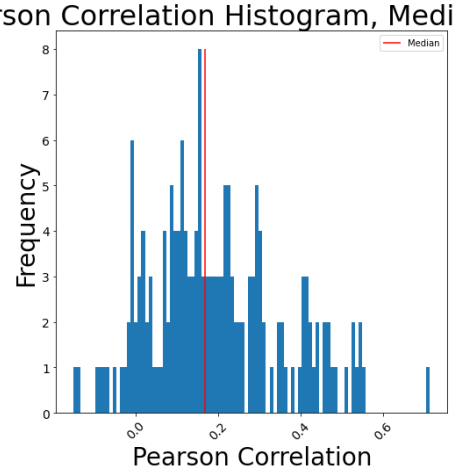
MSE: 1.24

## 5.2 Comparing Predictability of 168 Highly Expressed Genes and 168 Protein Coding Genes

In these experiments, the only regularization option is L1. For each model and for each gene type option, we share the results of the best-performing L1 regularization option. As it can be seen in the plots below, it is easier to predict 168 protein-coding genes compared to 168 highly expressed genes for both models.

- SklearnLinear Model
  - 168 Highly Expressed Genes

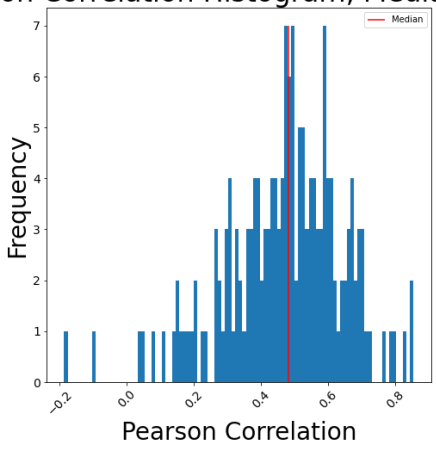
    Per Gene Pearson Correlation Histogram, Median: 0.17, IQR: 0.2

    Regularization: L1 (0.01)

    MSE: 0.91
  - 168 Protein Coding Genes

    Per Gene Pearson Correlation Histogram, Median: 0.48, IQR: 0.21

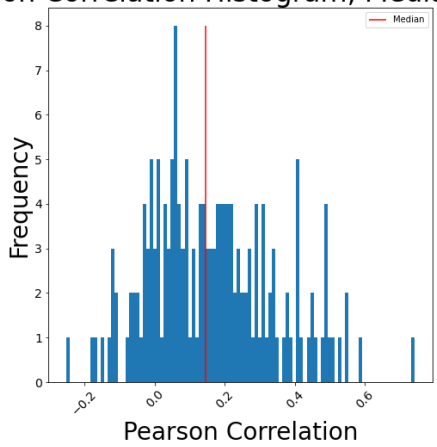
    
    

    Regularization: L1 (0.001)

    MSE: 0.75
- SklearnPerGene Model

- 168 Highly Expressed Genes

  Per Gene Pearson Correlation Histogram, Median: 0.15, IQR: 0.25

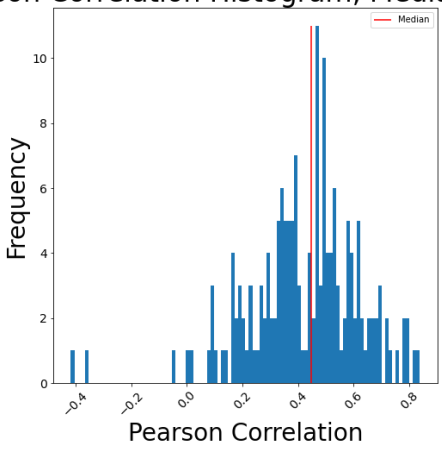



  Regularization: L1 (0.001)

  MSE: 0.92

- 168 Protein Coding Genes

  Per Gene Pearson Correlation Histogram, Median: 0.45, IQR: 0.22




  Regularization: L1 (0.001)

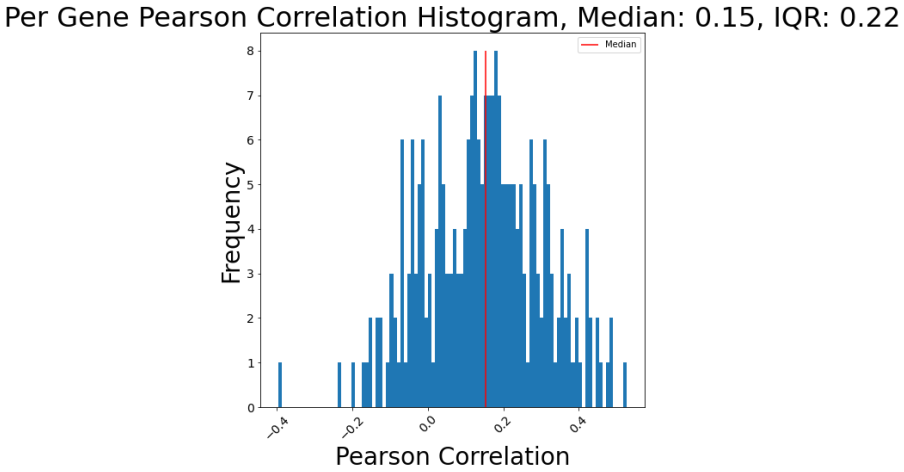
  MSE: 0.79

## 5.3 Comparing Predictability for Varying Number of Genes

In these experiments, the regularization options are L1, L2, Dropout, L1 + Dropout, and L2 + Dropout. For the same model, the median per-gene Pearson Correlation is similar between 250 highly expressed genes and 1000 highly expressed genes. However, MSE is higher for 1000 highly expressed genes.

- DLPerGene Model

- 250 Highly Expressed Genes



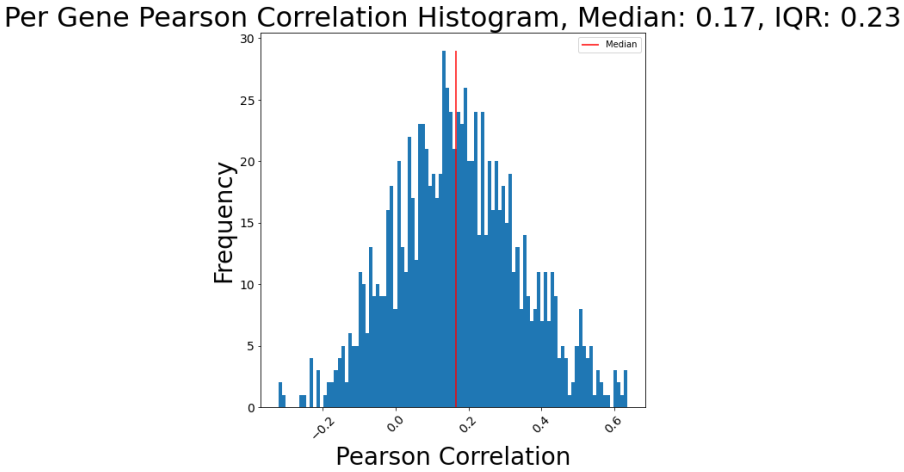

Initial Learning Rate: 0.01

Number of Parameters: 8 000

Regularization: Dropout (0.33)

MSE: 0.91

- 1000 Highly Expressed Genes



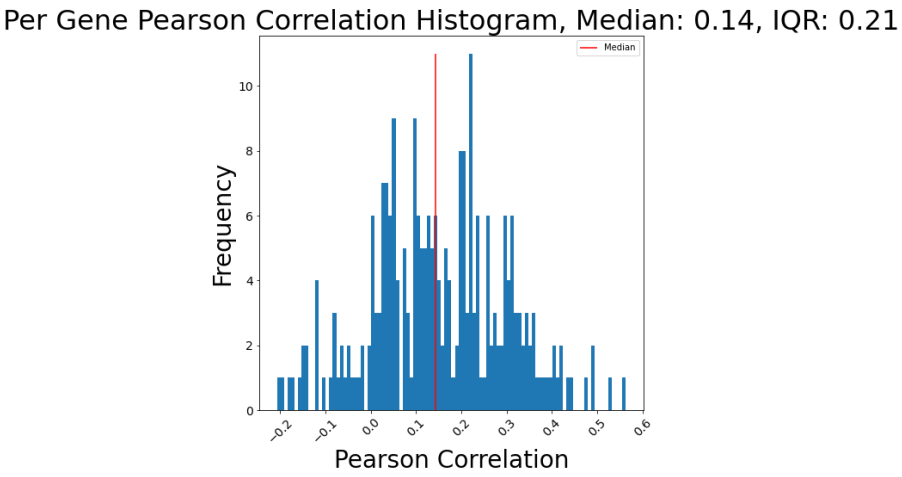

Initial Learning Rate: 0.01

Number of Parameters: 32 000

Regularization: Dropout (0.5)

MSE: 1.99

- DLLinear Model
  - 250 Highly Expressed Genes



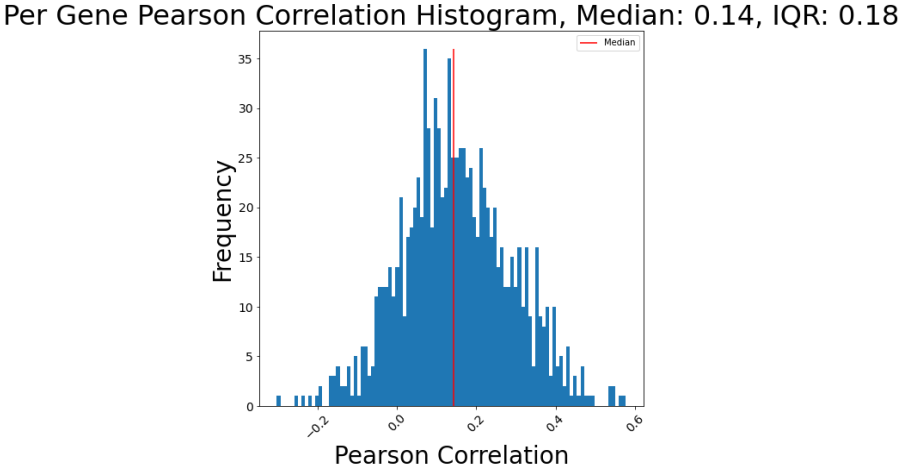

Initial Learning Rate: 0.01

Number of Parameters: 70 250

Regularization: Dropout (0.5) + L2 Reg (0.0001)

MSE: 0.94

– 1000 Highly Expressed Genes




Initial Learning Rate: 0.001

Number of Parameters: 1 031 000

Regularization: Dropout (0.33)

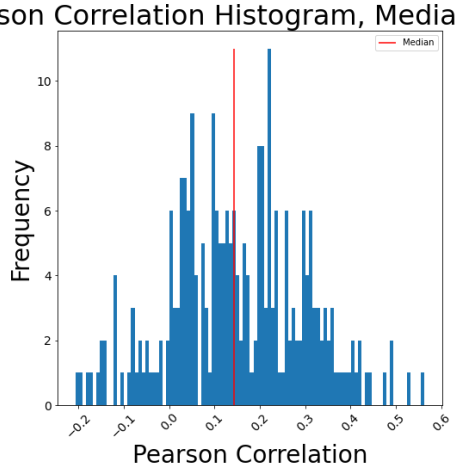
MSE: 2.16

## 5.4 Comparing Single Models and Per-Chromosome Models

We observed that using a per-chromosome model does not always yield better results. We provide one example where using a per-chromosome model improves the results and one example where it worsens the results. In these experiments, the regularization options are L1, L2, Dropout, L1 + Dropout, and L2 + Dropout.

- DLLinear Model (250 Highly Expressed Genes)
  - Single Model

    Per Gene Pearson Correlation Histogram, Median: 0.14, IQR: 0.21

    Initial Learning Rate: 0.01

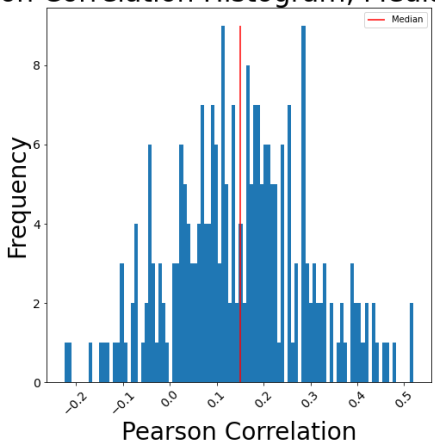
    Number of Parameters: 70 250

    Regularization: Dropout (0.5) + L2 Reg (0.0001)

    MSE: 0.94

  - Per-Chromosome Model

    Per Gene Pearson Correlation Histogram, Median: 0.15, IQR: 0.19

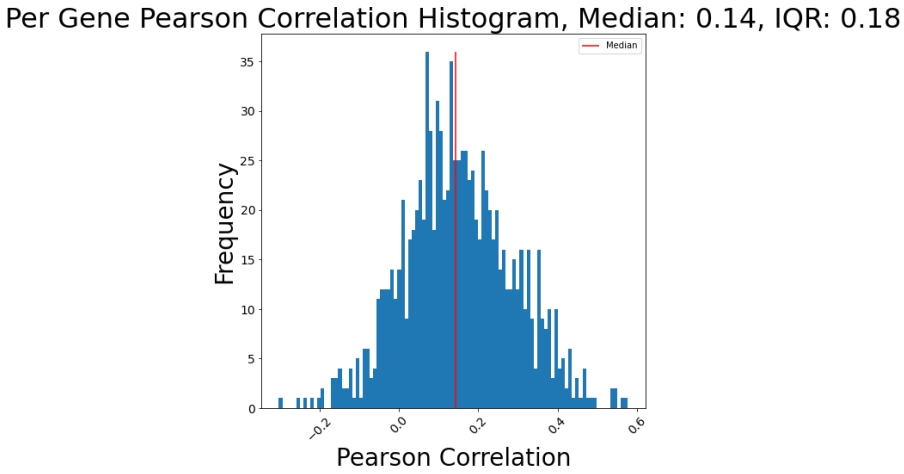
    Initial Learning Rate: 0.01

Number of Parameters: 11 488

Regularization: Dropout (0.5)

MSE: 0.91

- DLLinear Model (1000 Highly Expressed Genes)
  - Single Model



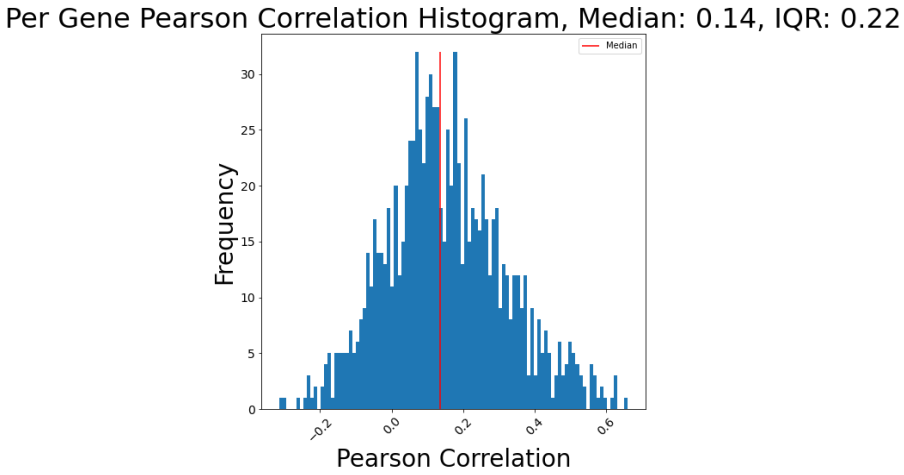

Initial Learning Rate: 0.001

Number of Parameters: 1 031 000

Regularization: Dropout (0.33)

MSE: 2.16

  - Per-Chromosome Model




Initial Learning Rate: 0.01

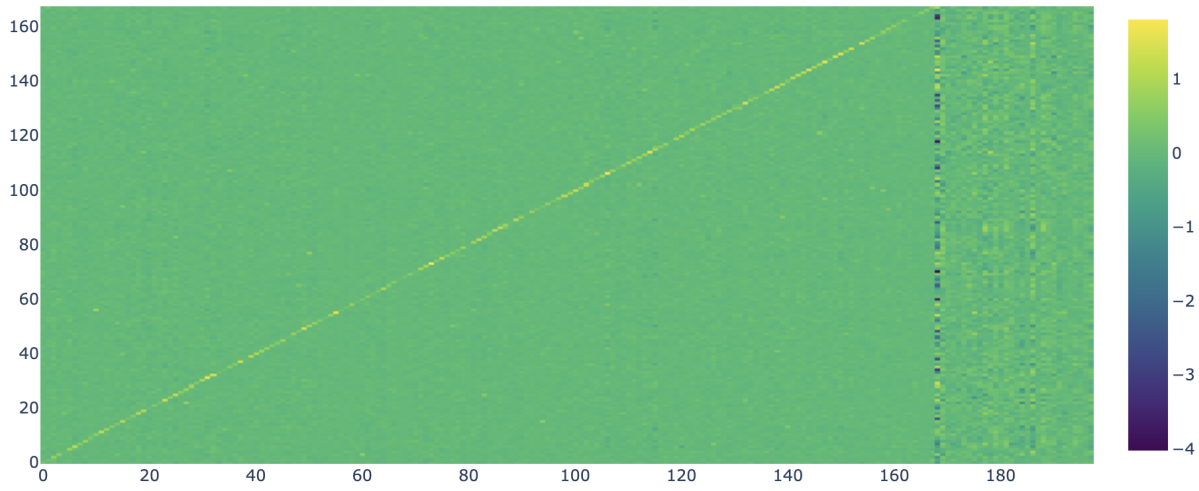
Number of Parameters: 87 790

Regularization: Dropout (0.5)

MSE: 2.01

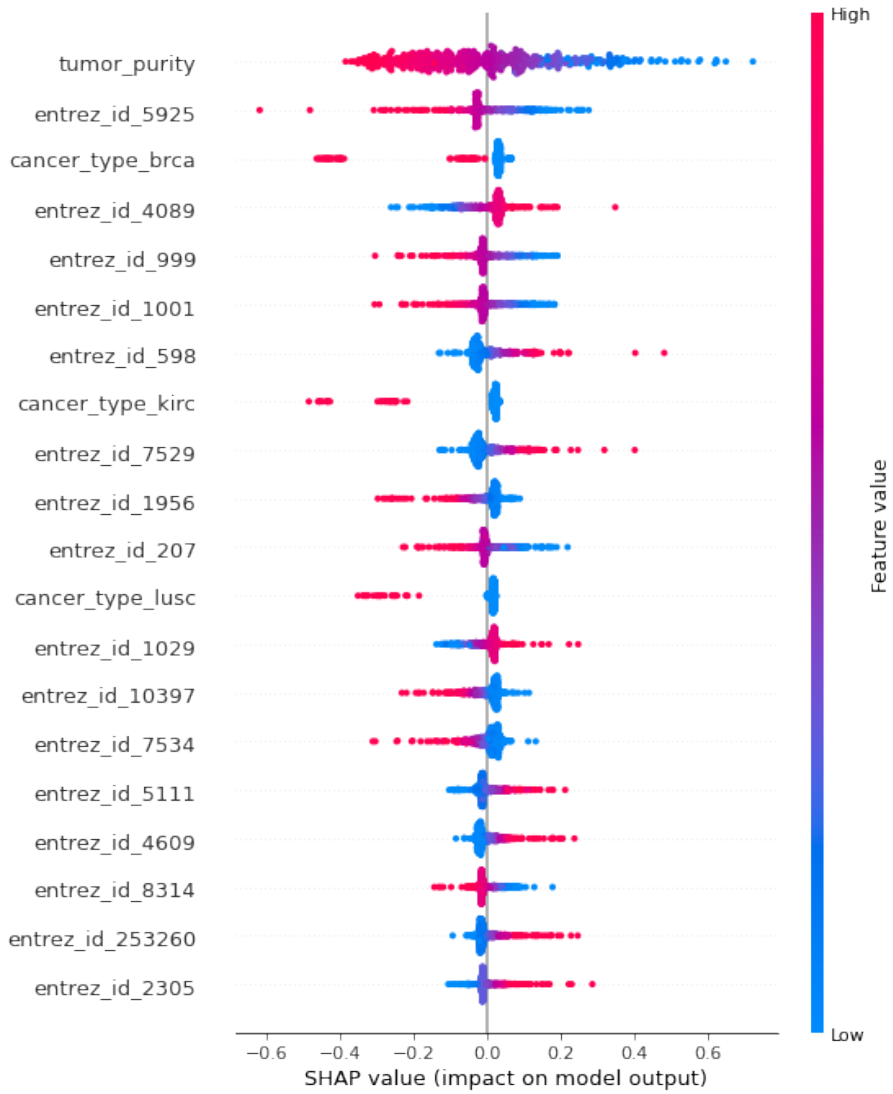
## 5.5 Interpretation of Linear Model Predictions

We fitted a SklearnLinear model using the 168 protein-coding genes of the UnthresholdedCNA + Tumour Purity dataset. In this experiment, we used an L1 regularization of 0.001. We visualized the weight matrix below. The rows correspond to the output variables (ZScore of 168 genes), and the columns correspond to the input variables (Unthresholded CNA of 168 genes, tumour purity and one-hot encoded cancer type). As it can be seen in the weight matrix below, the diagonal elements are generally close to 1, indicating that most of the genes are cis-regulated. The 169th column has values for tumour purity. The columns after the 169th column correspond to the one-hot encoding of the cancer type. As it can be seen in the plot below, tumour purity (the proportion of cancer cells in the tumour sample) is an important feature while predicting ZScore. This is because, in a given tumour sample, only the cancer cells are affected by copy number aberration.




## 5.6 Post-Hoc Explanation of Non-Linear Model Predictions

We plotted the Shap values for DLMLP model below. Since generating Shap values takes a long time, we analyzed only one gene that is strongly cis-regulated (BCL2L11, whose diagonal weight is 0.94 in the linear model's weight matrix).




Although BCL2L11 is a strongly cis-regulated gene, this effect is not obvious in the Shap values plot (since 10018 which is the entrezgene ID of BCL2L11 is not in the first 20 most important features). We can also see that tumour purity is selected as the most important feature.

# Future Work

## 6.1 Simulated Dataset

A simulated dataset can be created with artificial CNA-ZScore links. Then one can suggest a new algorithm that can identify these links and then compare it with TreeLasso and Sparse Canonical Correlation Analysis on the simulated dataset.

## 6.2 Optimal Transport

The main motivation behind optimal transport is to find a mapping between two probability measures such that the total transportation cost is minimized [Huesmann and Sturm 2013]. In our problem, we can use optimal transport to map copy number aberration information to the Z Score of a given tumor sample. Both the input and the output vectors are high-dimensional in our case, therefore we found a method ([Korotin et al. 2021], [iamalexkorotin 2021]) that can do optimal transport in the high-dimensional setting. One can try mapping CNA information to Z Score using this method.

# Conclusion

In this work, we provided a literature review of existing methods which predict GEX/Z Score from CNA information. We also provided the analysis of the relationship between input (RPPA, CNA) and output (Z Score) variables using per-gene Pearson correlation histograms. We implemented several algorithms which can predict Z Score from the given inputs. We compared the informativeness of different input types (RPPA, CNA), predictability of different sets of genes (168 highly expressed genes, 168 protein-coding genes), predictability of varying number of genes (250 highly expressed genes, 1000 highly expressed genes), and performance of single models and per-chromosome models. We also showed that a linear model's weight matrix can be used to identify CNA-Z Score links, however, one should take into account the fact that when the input variables have CNAs of genes that are closely located, a CNA-Z Score link's corresponding weight might be less than it should be. This problem is already handled with previously proposed algorithms TreeLasso and sCCA. Finally, we showed that using Shap values on a non-linear model's predictions might not be able to reveal some of the true CNA-Z Score links.

# Bibliography

ALONSO, M., AUSSÓ, S., LOPEZ-DORIGA, A., CORDERO, D., GUINÓ, E., SOLÉ, X., BARENYS, M., OCA, J., CAPELLÁ, G., SALAZAR, R., SANZ-PAMPLONA, R., AND MORENO, V. 2017. Comprehensive analysis of copy number aberrations in microsatellite stable colon cancer in view of stromal component. *British Journal of Cancer 117* (07).

CHEN, X., LIN, Q., KIM, S., CARBONELL, J. G., AND XING, E. P. 2012. Smoothing proximal gradient method for general structured sparse learning. *CoRR abs/1202.3708*.

CHENG, L., WANG, P., YANG, S., YANG, Y., ZHANG, Q., ZHANG, W., XIAO, H.-S., GAO, H., AND ZHANG, Q. 2012. Identification of genes with a correlation between copy number and expression in gastric cancer. *BMC medical genomics 5* (05), 14.

CHEUNG, V., SPIELMAN, R., EWENS, K., WEBER, T., MORLEY, M., AND BURDICK, J. 2005. Cheung vg, spielman rs, ewens kg, weber tm, morley m, burdick jt. mapping determinants of human gene expression by regional and genome-wide association. nature 437: 1365-1369. *Nature 437* (11), 1365–9.

CURTIS, C., SHAH, S., CHIN, S.-F., TURASHVILI, G., RUEDA, O., DUNNING, M., SPEED, D., LYNCH, A., SAMARAJIWA, S., YUAN, Y., GRÄF, S., HA, G., HAFFARI, G., BASHASHATI, A., RUSSELL, R., MCKINNEY, S., CALDAS, C., APARICIO, S., BRENTON, J., AND BØRRESEN-DALE, A.-L. 2012. The genomic and transcriptomic architecture of 2,000 breast tumors reveals novel subgroups. *Nature 486* (04), –.

DUTTA, D., SEN, A., AND SATAGOPAN, J., 2021. Sparse canonical correlation to identify breast cancer related genes regulated by copy number aberrations, 09.

GOLDMAN, M., CRAFT, B., HASTIE, M., REPEČKA, K., MCDADE, F., KAMATH, A., BANERJEE, A., LUO, Y., ROGERS, D., BROOKS, A., ZHU, J., AND HAUSSLER, D. 2020. Visualizing and interpreting cancer genomics data via the xena platform. *Nature Biotechnology 38* (05).

HUESMANN, M., AND STURM, K.-T. 2013. Optimal transport from Lebesgue to Poisson. *The Annals of Probability 41*, 4, 2426 – 2478.

IAMALEXKOROTIN, 2021. https://github.com/iamalexkorotin/wasserstein2benchmark/blob/main/notebooks/w2_train_hd_benchmark.ipynb.

KIM, S., AND XING, E. 2009. Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eqtl mapping. *The Annals of Applied Statistics 6* (09).

KOROTIN, A., LI, L., GENEVAY, A., SOLOMON, J., FILIPPOV, A., AND BURNAEV, E. 2021. Do neural optimal transport solvers work? A continuous wasserstein-2 benchmark. *CoRR abs/2106.01954*.

LUNDBERG, S. M., AND LEE, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett,

Eds. Curran Associates, Inc., 4765–4774.

PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N., ANTIGA, L., DESMAISON, A., KOPF, A., YANG, E., DEVITO, Z., RAISON, M., TEJANI, A., CHILAMKURTHY, S., STEINER, B., FANG, L., BAI, J., AND CHINTALA, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., 8024–8035.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

PEREIRA, B., CHIN, S.-F., RUEDA, O., VOLLAN, H.-K., PROVENZANO, E., BARDWELL, H., PUGH, M., JONES, L., RUSSELL, R., SAMMUT, S., TSUI, D., LIU, B., DAWSON, S.-J., ABRAHAM, J., NORTHEN, H., PEDEN, J., MUKHERJEE, A., TURASHVILI, G., GREEN, A., AND CALDAS, C. 2016. Erratum: The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nature Communications 7* (06), 11908.

SEAL, D. B., DAS, V., GOSWAMI, S., AND DE, R. K. 2020. Estimating gene expression from dna methylation and copy number variation: A deep learning regression model for multi-omics integration. *Genomics 112*, 4, 2833–2841.

SHAO, X., LV, N., LIAO, J., LONG, J., XUE, R., AI, N., XU, D., AND FAN, X. 2019. Copy number variation is highly correlated with differential gene expression: A pan-cancer study. *BMC Medical Genetics 20* (11).

SRIHARI, S., KALIMUTHO, M., LAL, S., SINGLA, J., PATEL, D., SIMPSON, P., KHANNA, K., AND RAGAN, M. 2016. Understanding the functional impact of copy number alterations in breast cancer using a network modeling approach. *Molecular BioSystems 12*, 3 (Mar.), 963–972. Funding Information: This work is supported by an Australian National Health amp; Medical Research Council (NHMRC) grant 1028742 to PTS and MAR. MK is supported by a Cancer Council Queensland (CCQ) Project Grant (1087363), KKK is a NHMRC Senior Principal Research Fellow (ID 613638). We thank METABRIC (Molecular Taxonomy of Breast Cancer International Consortium) for granting us access to the breast cancer dataset. We thank Dr Alison Anderson (UQ) for useful discussions.

STRANGER, B., FORREST, M., CLARK, A., MINICHIELLO, M., DEUTSCH, S., LYLE, R., HUNT, S., KAHL, B., ANTONARAKIS, S., TAVARÉ, S., DELOUKAS, P., AND DERMITZAKIS, E. 2006. Genome-wide associations of gene expression variation in humans. *PLoS genetics 1* (01), e78.

WU, H.-T., HAJIRASOULIHA, I., AND RAPHAEL, B. J. 2014. Detecting independent and recurrent copy number aberrations using interval graphs. *Bioinformatics 30*, 12 (06), i195–i203.

ZHU, J., ZHANG, B., SMITH, E., DREES, B., BREM, R., KRUGLYAK, L., BUMGARNER, R., AND SCHADT, E. 2008. Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature genetics 40* (08), 854–61.