

**Internship Report:**  
Predicting Drug Activity on  
Cancer Cell-lines Based on  
Gene Expression

By: Aaron Mohammed

MS Bioinformatics & Computational Biology  
Department of Molecular Medicine  
College of Medicine

Supervisor: Dr. Feng Cheng  
Department of Pharmaceutical Science  
College of Pharmacy

University of South Florida

Summer 2022

# **Table of Contents**

I. Summary.....	Pg.1
II. Background.....	Pg.1
III. Tasks & Results	
1. Correlation Analysis.....	Pg.2
2. Biological Significance.....	Pg.4
3. WGCNA.....	Pg.7
4. Machine Learning.....	Pg.11
IV. Closing Remarks	
Citations.....	Pg.

## **I. Summary**

During the summer of 2022, I worked with Dr. Feng Cheng from the Department of Pharmaceutical Science at the University of South Florida. The purpose of our research project was to investigate whether gene expression could be used to predict drug activity on cancer cell lines from the NCI-60 panel. The R programming language was used to carry out this research. The drug activity data and gene expression datasets were obtained from the CellMiner database. The gene expression datasets included RNAseq, Affy HuEx, and Agilent mRNA data. Correlation coefficients were calculated between drug activity and each gene expression dataset. Based on the results, the gene expression dataset that resulted in the highest correlations was chosen to be used in machine learning models. The DAVID database and hierarchical clustering were used to gauge if the correlations had any biological relevance. The 'WGCNA' R package was used to cluster the genes into module eigengenes and then used as support vectors in machine learning models created using the 'e1071' R package, with drug activity as the response vector. In this report, I give detailed explanations of what my tasks were, the computational tools I used, the results I got, and what I learned.

## **II. Background**

During the late 1980's, the Developmental Therapeutics Program (DTP) of the National Cancer Institute (NCI) began the development of an anticancer drug screening called the NCI-60 cell line panel. The NCI-60 is made up of 60 cancer cell lines from various types of cancer like skin, colon, and breast cancer. So far, over 80,000 compounds have been screened *in vitro*. The NCI-60 panel plays a very important role in drug discovery, in fact, most FDA approved drugs were screened in the NCI-60 panel at least twice<sup>1</sup>. Since the cell-lines in this panel evolve over time, they sometimes gain resistance towards drugs they were once sensitive to and vice versa. Keeping track of their genomic changes over time can provide valuable data that aides in drug discovery. If the expression of certain genes in these cancer cell-lines can determine whether or not they are sensitive towards anti-cancer drugs, then maybe gene expression profiles that are similar across cell-lines can be used to predict if a drug will be active towards it. This is what we sought out to investigate in this project. Our approach involved using support vector machines (SVMs), which are supervised machine learning models that can be used for classification.

Gene expression data for each of these cell lines were downloaded from CellMiner, a database created by the DTP<sup>2</sup>. The NCI-60 drug activity data that was used for this project was also downloaded from CellMiner. The raw “Compound activity: DTP NCI-60” dataset was used. It contained over 50,000 different compounds, of which 158 were FDA approved drugs and 79 were in clinical trials. In this dataset, drug activity is expressed as the negative log of GI50<sup>3</sup>. GI50 is the concentration of the drug at which there is a 50% reduction in growth of the cancer cells<sup>2</sup>. This means that a high value indicates only a small concentration of the drug was needed to inhibit the proliferation of the cells by 50%. The compound activity value of 13 is the max value in this dataset, while -4 is the lowest. 3 gene expression datasets were downloaded from CellMiner and the dataset that had the best correlations with drug activity was used for machine learning. The RNAseq, Affy HuEx 1.0, and Agilent mRNA datasets were downloaded.

### **III. Tasks & Results**

#### **1. Correlation Analysis**

After downloading the drug activity and expression data, R was used to calculate correlation coefficients. This was done in order to determine which gene expression dataset would serve as the better predictor. The drug activity dataset contained over 80,000 rows since it contained multiple entries/experiments for most of the compounds. In order to reduce runtime, the size of the drug activity dataset was reduced by removing the drug activity data from past experiments and keeping the data from only the most recent experiments for each drug. This was accomplished by utilizing the `rev()` and `duplicate()` functions, and the `not, !`, operator to subset the data frame based on the column that contained the NSC IDs for each drug. This reduced the number of drugs from 83,680 to 56,461. To reduce the size further, the data frame was filtered and only the compounds that were FDA approved, had a max activity value  $\geq 8$ , and a standard deviation  $\geq 1$  were extracted. This reduced the number of drugs to 21.

The gene expression data was filtered so that only highly expressed genes were kept. This was determined by calculating average expression for each gene across the cell lines and keeping the ones with average expression greater than or equal to a certain value. For the RNAseq dataset, genes that had average expression of at least 1 were kept, reducing the number of genes from 23,808 to 10,131. For Affy HuEx, the average expression cutoff was 6 and for Agilent mRNA, the

cutoff was 5. The number of genes for Affy HuEx were reduced from 1,048,575 to 268 and 41,090 to 845 for Agilent mRNA.

Once the datasets were filtered, it was then time to get the correlation coefficients between drug activity and gene expression. At first, a nested for loop containing the `cor.test()` function was used, but Dr. Cheng recommended that I get in the habit of using the “apply” functions since those functions are more efficient than for loops for very large datasets. I accomplished this by creating a functional or “nested function”, which is a function of functions. First, the datasets were all transposed so that the drug IDs and gene IDs became columns. The names of drug and gene IDs were stored in string vectors, called `drug_name` and `G` respectively, to be used as column indexes. Two `lapply()` functions and a `cor.test()` function were used, each nested within the other. That functional is shown below,

```
Corr <- lapply(drug_name, function(drug_name, drugs_df){
  lapply(gene_name, function(gene_name, genes_df){
    cor.test(as.numeric(drugs_df[[drug_name]]),
             as.numeric(genes_df[[gene_name]])) %>%
    tidy()}, genes_df) %>%
  bind_rows() %>%
  mutate(Gene = gene_name) %>%
  select(Gene, estimate, p.value) %>%
  as.data.frame()}, drugs_df)
```

where `Corr` is the data frame the output is stored in, `drugs_df` is the drug activity dataset, and `genes_df` is the gene expression dataset. In order to save the output from this functional, the ‘broom’ and ‘dplyr’ packages were used. The pipe operator, `%>%`, and the `tidy()` function from the ‘broom’ package were used to store the statistical output of `cor.test()` into a tibble, which is essentially a simple version of a data frame that minimizes memory. The `bind_rows()`, `mutate()`, and `select()` functions from the ‘dplyr’ package were used to manipulate the tibbles and prepare them for conversion to data frames. The output of this nested `lapply()` function is a list of data frames, so the `ldply()` function from the ‘plyr’ package was used to take values from each data frame stored in the list and place them into a single data frame.

The correlation coefficients and p values from the list were extracted and stored in separate data frames using for loops. Another data frame containing the max correlation values, standard deviation of activity, max activity, and names for each drug was created to be exported to excel.

This process was repeated for each gene expression dataset, using both the Pearson and Spearman methods. In total, there were 6 resulting datasets. The RNAseq dataset was chosen since it was more correlated with drug activity than the other gene expression datasets. The max correlation coefficients using the Pearson method is shown in Table 1. The drug with the highest Pearson correlation coefficient was Dabrafenib, 0.868 with a p-value of 1.093e-14.

ID	Drug Name	RNAseq	Affy HuEx	Agilent mRNA
NSC_764134	Dabrafenib	0.868444595	0.617700142	0.628898795
NSC_778304	Encorafenib	0.79637691	0.604124756	0.596412285
NSC_759877	Dasatinib (Salt)	0.65954711	0.477472315	0.667926976
NSC_764042	ARRY-162	0.613473369	0.395597683	0.49680274
NSC_768068	Cobimetinib (XL-518)	0.609095692	0.454204284	0.460534312
NSC_732517	Dasatinib	0.601511664	0.443693083	0.514676278
NSC_287459	Cytarabine	0.600924975	0.405522595	0.550401927
NSC_741078	Selumetinib	0.588182718	0.382604012	0.473002774
NSC_758246	Trametinib	0.565996502	0.431944273	0.447800518
NSC_606698	Rapamycin	0.553280038	0.363508897	0.381078898
NSC_733504	Everolimus	0.547620796	0.519850501	0.490325446
NSC_778590	Cobimetinib (GDC-0623)	0.546014427	0.363009064	0.444349829
NSC_226080	Rapamycin (Salt_2)	0.53671643	0.515807417	0.487754655
NSC_683864	Temsirolimus	0.534450908	0.359875414	0.412855524
NSC_758664	Rapamycin (Salt_1)	0.514088407	0.454547183	0.381471794
NSC_628503	Docetaxel	0.493749954	0.399446112	0.491897444
NSC_728073	Irinotecan	0.48115399	0.428024279	0.474562362
NSC_698037	Pemetrexed	0.465206531	0.358077451	0.48335166
NSC_90636	Vinblastine	0.461302367	0.496862542	0.550255982
NSC_726630	Belinostat	0.447092739	0.316985451	0.426815852
NSC_760087	Vinorelbine	0.322881507	0.369471162	0.45003084

**Table 1** – Max correlation coefficient values between drug activity and gene expression.

## 2. Biological Significance

Before moving forward, it was necessary to find out if the results of the correlation analysis could be explained by biological phenomena or just random chance. If the answer is the latter, then there would not be much point in moving forward with using this gene expression dataset for machine learning. If the results of this correlation analysis do make sense within the context of biology, then it would be likely that the highly expressed genes that are positively or negatively

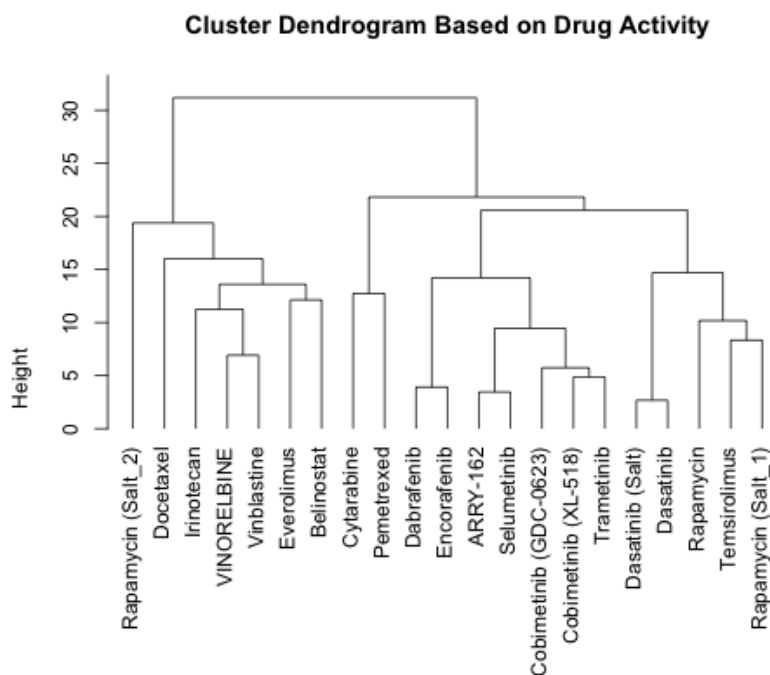
correlated with drug activity are involved with pathways related to cancer. It would also be likely that positively or negatively correlated genes are associated with metabolic pathways, since many of these drugs target proteins that are involved with cell signaling. A webserver called DAVID

ID	NAME	POSITIVE CORRELATION	NEGATIVE CORRELATION
287459	Cytarabine	Metabolic pathway	Pathways in cancer
698037	Pemetrexed	Metabolic pathway	Pathways in cancer
764134	Dabrafenib	Metabolic pathway	Pathways in cancer
778304	Encorafenib	Metabolic pathway	Pathways in cancer
764042	ARRY-162	Metabolic pathway	Pathways in cancer
741078	Selumetinib	Metabolic pathway	Pathways in cancer
778590	Cobimetinib (GDC-062)	Metabolic pathway	Pathways in cancer
768068	Cobimetinib (XL-518)	Metabolic pathway	Pathways in cancer
758246	Trametinib	Metabolic pathway	Neither
759877	Dasatinib (Salt)	Pathways in cancer	Metabolic pathway
732517	Dasatinib	Pathways in cancer	Metabolic pathway
758664	Rapamycin (Salt_1)	Pathways in cancer	Neither
683864	Temsirolimus	Pathways in cancer	Metabolic pathway
606698	Rapamycin	Pathways in cancer	Metabolic pathway
760087	Vinorelbine	Metabolic pathway	Pathways in cancer
90636	Vinblastine	Metabolic pathway	Pathways in cancer
728073	Irinotecan	Metabolic pathway	Pathways in cancer
726630	Belinostat	Metabolic pathway	Pathways in cancer
733504	Everolimus	Pathways in cancer	Metabolic pathway
628503	Docetaxel	Metabolic pathway	Pathways in cancer
226080	Rapamycin (Salt_2)	Pathways in cancer	Metabolic pathway

**Table 2** – This table displays results of the pathway analysis and shows whether any of the positively or negatively correlated genes are related to pathways in cancer or metabolic pathways.

was used to investigate the pathways associated with genes that were positively and negatively correlated with drug activity. DAVID is an abbreviation for Database for Annotation, Visualization, and Integrated Discovery, it allows a user to access tools and databases for functional annotation analysis<sup>4</sup>. Gene IDs for the top 500 positively correlated and lowest 500 negatively correlated genes for each drug were inputted to DAVID. Each search check to see if the gene IDs are included in any of the databases contained in DAVID. If they are, a data tables displaying lists of pathways associated with the genes can be accessed. The KEGG pathway database was selected for this analysis. KEGG stands for Kyoto Encyclopedia of Genes and Genomes, and it contains many annotations of gene functions<sup>5</sup>. A table was created to keep track

of whether the genes were associated with either metabolic pathways or pathways in cancer. If both pathways appeared in the results, the one that had the most gene counts was chosen. The results of this analysis are shown in Table 2 and includes the genes involved with the mechanism of action for each drug that were included in the NCI-60 drug activity dataset. It was found that 14 drugs had positively correlated genes that were primarily associated with metabolic pathways and 7 had positively correlated genes that were primarily associated with pathways in cancer. 13 drugs had negatively correlated genes that were primarily associated with pathways in cancer and 6 that had negatively correlated genes that were primarily associated with metabolic pathways. There were two drugs who's lowest 500 correlated genes were not associated with metabolic pathways or pathways in cancer.



**Figure 1** – This map displays how similar each drug's activity is with each other. Similarity increases as the height gets smaller.

Hierarchal clustering of drug activity was used to cluster the drugs based on drug activity. This was done to find out if the drugs cluster together based on the results of the pathway analysis. Would the drugs that were positively correlated with genes associated with metabolic pathways and negatively with cancer pathways cluster together and vice versa? A cluster dendrogram of



drug activity was created in R and is shown in Figure 1. The first step in making this plot was calculating the distances between all possible pairs of drug activity values between all drugs and placing them in a distance matrix. This was done by using the `dist()` function. The distance matrix was then inputted into the `hclust()` function which uses complete linkage clustering to form the dendrogram. Although the drugs were not completely separated based on pathways in metabolism and cancer, most did come together that way in clusters that have a small height. The colored sets shown in Table 3 represent clusters with a height less than 10. There are some exceptions, but almost 70% of them clustered this way. This means that the correlations between gene expression and drug activity appear to make sense biologically. This becomes clearer once the genes associated with the mechanisms of action for each drug are included in Table 3. They were obtained from the original NCI-60 drug activity dataset.

ID	Name	Mechanism	Positively Correlated Genes	Negatively Correlated Genes
287459	Cytarabine	Ds	Metabolic pathway	Pathways in cancer
698037	Pemetrexed	Df AM GARTF DHFR	Metabolic pathway	Pathways in cancer
764134	Dabrafenib	PK:BRF	Metabolic pathway	Pathways in cancer
778304	Encorafenib	PK:BRF	Metabolic pathway	Pathways in cancer
764042	ARRY-162	PK:STK,MAP2K,MAP2K1,MAP2K2	Metabolic pathway	Pathways in cancer
741078	Selumetinib	PK:STK,YK,MAP2K,MAP2K1,MAP2K2	Metabolic pathway	Pathways in cancer
778590	Cobimetinib (GDC-0623)	PK:MAP2K,MAP2K1	Metabolic pathway	Pathways in cancer
768068	Cobimetinib (XL-518)	PK:MAP2K,MAP2K1	Metabolic pathway	Pathways in cancer
758246	Trametinib	PK:STK,MAP2K,MAP2K1,MAP2K2	Metabolic pathway	Neither
759877	Dasatinib (Salt)	BCR-ABL PK:YK,PDGFR,KIT	Pathways in cancer	Metabolic pathway
732517	Dasatinib	BCR-ABL PK:YK,PDGFR,KIT	Pathways in cancer	Metabolic pathway
758664	Rapamycin (Salt_1)	PK:STK,MTOR	Pathways in cancer	Neither
683864	Temsirolimus	PK:STK,MTOR	Pathways in cancer	Metabolic pathway
606698	Rapamycin	PK:STK,MTOR	Pathways in cancer	Metabolic pathway
760087	Vinorelbine	TUBB Tu-frag	Metabolic pathway	Pathways in cancer
90636	Vinblastine	TUBB Tu-frag	Metabolic pathway	Pathways in cancer
728073	Irinotecan	TOP1	Metabolic pathway	Pathways in cancer
726630	Belinostat	HDAC	Metabolic pathway	Pathways in cancer
733504	Everolimus	PK:STK,MTOR	Pathways in cancer	Metabolic pathway
628503	Docetaxel	TUBB Tu-stab	Metabolic pathway	Pathways in cancer
226080	Rapamycin (Salt_2)	PK:STK,MTOR	Pathways in cancer	Metabolic pathway

**Table 3** – The blue and red boxes represent the 2 clusters with the largest distance. The colored sets are drugs inside clusters that have a height less than 10.

### 3. WGCNA

Since the RNAseq dataset had thousands of genes, the ‘WGCNA’ R package was used to cluster the genes and form module eigengenes. This was necessary because SVMs are not suitable for large datasets. Instead of having to use every gene in a dataset, a module eigengene can be used to represent a large number of genes that have similar patterns of expression. Module eigengenes

can be created by carrying out weighted correlation network analysis or WGCNA. This type of analysis is based on the correlations between multiple genes and how their gene expression changes through different samples. If their gene expression is co-expressed i.e. follows similar patterns across sample, they form a connection in the network. Unlike regular gene co-expression networks (GCNs) that assign binary values to represent connected and unconnected, weighted co-expression networks assignments are weighted based on the degree of connectedness<sup>6</sup>. This is what is referred to as “soft-thresholding”. The clusters that form in these networks are call modules and it is these modules that form eigengenes.

Before using the ‘WGCNA’ package, input variables needed to be prepared. The RNAseq gene expression data was imported into R along with a file that contained phenotype data for each cell-line. That phenotype data was manually created in excel and consisted of the name of each cell-line and type of cancer they originate from. The first 6 columns of data that contained identifiers and locations of each gene were extracted and stored in a .csv file. This file would be used as a gene map later on. Mitochondria and ribosomal genes were removed because they were not important for this study. The gene expression values were normalized and scaled according to this formula:  $\log_2(\text{FPKM}+1)$ . Genes with low expression were removed by calculating the average expression for each gene and only keeping the ones that had an average greater than 1. Once filtered, coefficients of variation (CV) were calculated for each gene by dividing standard deviation by average expression and the genes that had a value less than 0.2 were removed. This was done to select genes with high variance. The gene expression dataset was now left with 4,725 genes.

Once the input variables were prepared, it was time to move forward and begin using the ‘WGCNA’ package. The first step was to analyze the network topology resulting from the gene expression data frame and different values of soft-thresholding power,  $\beta$ , in what is called an adjacency function. This function is what determines how the co-expression network is constructed. This step is necessary for deciding the right thresholding power for the co-expression network to have “scale-free topology”, which means the degree distribution follows a power law distribution,

$$p(k) \sim k^{-\gamma}$$

where  $p(k)$  is degree distribution,  $k$  is degree, and  $\gamma$  is a constant. Degree refers to the number of connections a node or gene in the network has with others. The reason why the network should follow this topology is because nature appears to have many scale-free networks, at least approximate ones<sup>7</sup>. In a gene co-expression network that is scale free, the degree distribution decays as  $k$  increases. As mentioned earlier, weighted correlation networks use soft-thresholding to classify connections. The adjacency function is represented by,

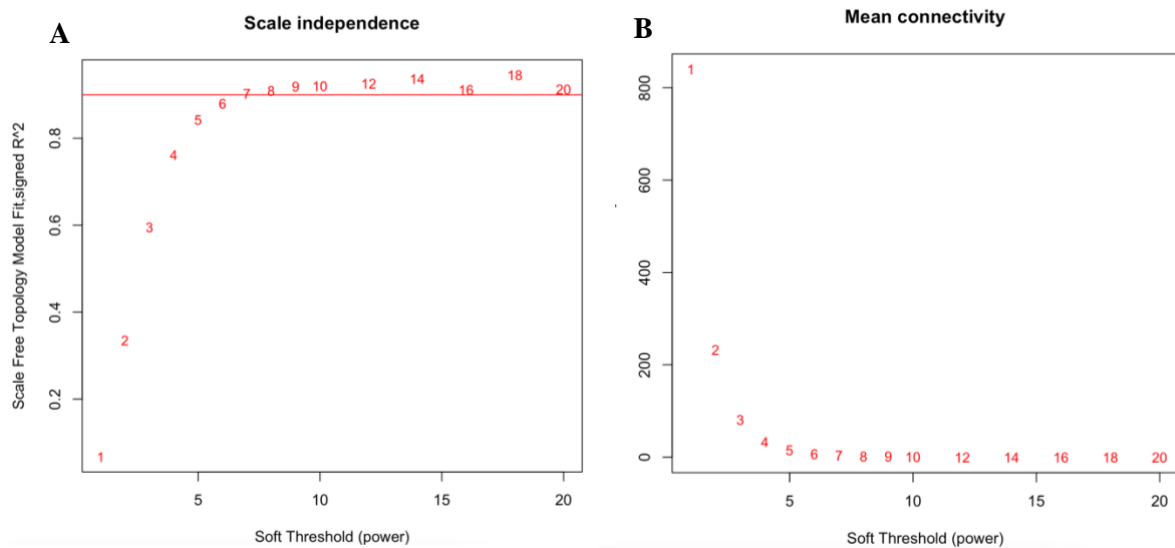
$$a_{ij} = |s_{ij}|^\beta$$

where  $a_{ij}$  is an element in an adjacency matrix, and  $s_{ij}$  is an element in a similarity matrix. Similarity is a measure of how much gene expression profiles correlate with one another and is given by the following equation,

$$s_{ij} = |cor(i, j)|$$

where  $cor(i, j)$  is a function for the Pearson correlation and  $i$  and  $j$  represent a pair of genes<sup>6</sup>. The elements of the adjacency matrix represent adjacency between a single pair of genes i.e. they represent whether a pair of genes are connected or co-expressed. In “hard-thresholding”, used by GCNs, these elements are 1’s and 0’s. As one can see from the similarity function, similarity in soft-thresholding can take on a range of values between 0 and 1. This is how the networks are weighted.

The function `pickSoftThreshold()` is the function used for network topology analysis. The inputs were the gene expression dataset, vector of power values, and level of verbosity, how much a computer tells the user what it is doing in the console. The output of this function is used to make two plots shown in Figure 2. These plots are used to choose the value of soft-thresholding power. In the scale independence plot of Figure 2A, the red line represents the threshold level for a network to be scale free. In order for the networks to be approximately scale-free, the value of power chosen should be above or close to the red line and should have a low value of mean connectivity. The plots displayed in Figure 2 were created using the gene expression dataset, so the value for power chosen was 7.



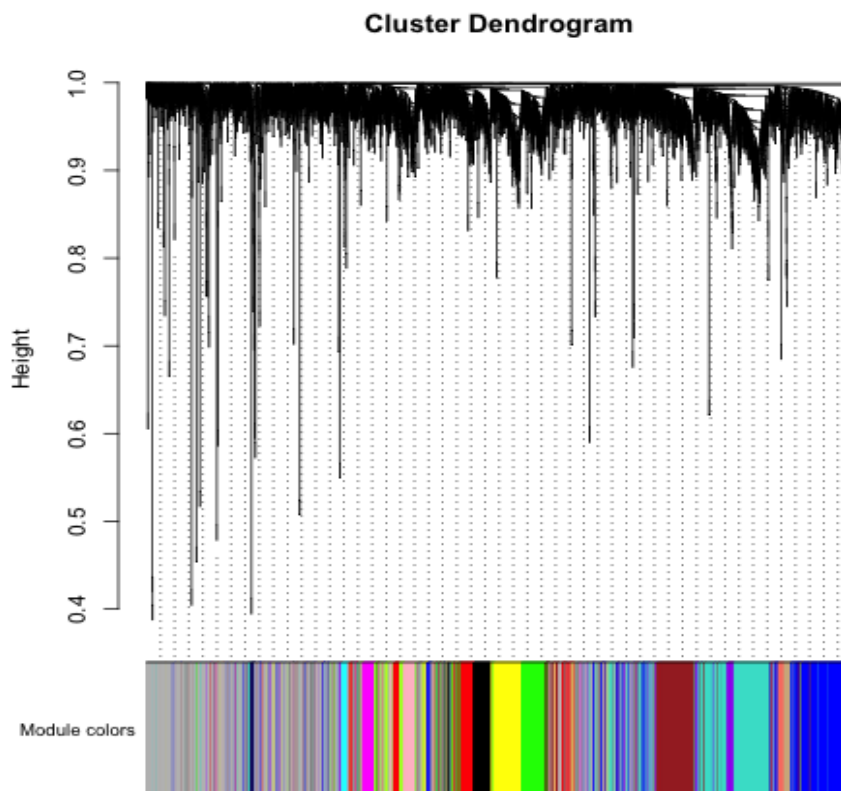
**Figure 2** – A: Plot displaying how much the co-expression network with this gene expression dataset fits a scale-free topology for different values of power. B: Plot displaying the average connectivity between genes in this dataset using different values of power.

After choosing 7 as the value for soft-thresholding power, it was time for construction of the network and detection of the modules. In the ‘WGCNA’ package, this could all be done by using just one function, `blockwiseModules()`. It takes in the gene expression data frame and power as input along with other parameters shown here,

```
output = blockwiseModules(datExpr, power = 7,
                           TOMType = "unsigned", minModuleSize = 30,
                           reassignThreshold = 0, mergeCutHeight = 0.25,
                           numericLabels = TRUE, pamRespectsDendro = FALSE,
                           saveTOMs = TRUE,
                           saveTOMFileBase = "Celltreatment",
                           verbose = 3)
```

where `TOMType` stands for Topological Overlap Matrices (TOMs), `minModuleSize` sets the minimum size of modules, `reassignThreshold` is a threshold in the form of a p-value used for reassigning genes between modules, `mergeCutHeight` is a value for cutting the height of the dendrogram, `numericLabels` is a logical where `FALSE` will lead to the modules being labeled with colors and `TRUE` with numbers, `pamRespectsDendro` is a logical that controls the behavior of

Partitioning Around Medoids (PAM), a method for clustering genes, saveTOMs saves the TOMs to the computer, saveTOMFileBase is a character string that serves as the file name base for files containing topological overlaps, and verbose dictates how many messages are displayed while the code is running. When this function is executed, a number of actions take place. First, the adjacencies between the genes are calculated and those values are used to make a TOM. If two genes have topological overlap, it means that they share similarity based on the other genes they are connected too<sup>8</sup>. In other words, they are a part of the same module. The TOMs are then scaled so that the data within them are comparable across samples and then the consensus between the matrices is calculated. That result is used to create a dendrogram in order to identify the modules. The modules are then combined if their expression profiles are very similar<sup>9</sup>. The final dendrogram can then be plotted, the one created for our gene expression dataset is shown in Figure 3. The module eigengenes can be extracted from the list that the output is stored in and is located in a data frame called “MEs”. In total, 16 module eigengenes were formed from this dataset.



**Figure 3** – A dendrogram of the genes after clustering. Each color corresponds to a module.

#### 4. Machine Learning

After generating the module eigengenes, the ‘e1071’ package was used to train and test SVMs. SVM stands for support-vector machine and it’s a supervised machine learning model that can be used for classification. During training, a model takes in “support vectors” and a “response” vector. The response vector is what the SVM is intended to predict and for binary classification, it contains 1’s and 0’s. The support vectors contain the data used to carry out regression and it requires at least 2 vectors. The rows of each element in these vectors are lined up with an element of the response vector. During training, the elements of the support vectors get plotted and the machine attempts to find “hyperplane”

NAME	ID	MEs	Accuracy	Sensitivity	Specificity	PPV	NPV
Cytarabine	287459	ME9_ME11	0.936	0.656	1.000	0.824	0.924
Dasatinib	732517	ME9_ME15	0.935	0.645	1.000	0.812	0.923
Encorafenib	778304	ME7_ME9	0.935	0.649	1.000	0.815	0.923
ARRY-162	764042	ME9_ME12	0.935	0.648	1.000	0.815	0.923
Dasatinib (Salt)	759877	ME3_ME9	0.934	0.650	1.000	0.821	0.922
Cobimetinib (XL-518)	768068	ME4_ME9	0.934	0.650	1.000	0.819	0.922
Dabrafenib	764134	ME1_ME9	0.934	0.649	1.000	0.820	0.922
Rapamycin (Salt_1)	758664	ME7_ME9	0.863	0.487	0.977	0.748	0.863
Cobimetinib (GDC-062)	778590	ME0_ME10	0.845	1.000	0.000	0.845	0.000
Everolimus	733504	ME1_ME8	0.845	0.000	1.000	0.000	0.845
Temsirolimus	683864	ME0_ME3	0.826	1.000	0.000	0.826	0.000
Rapamycin (Salt_2)	226080	ME7_ME11	0.744	0.862	0.549	0.789	0.665
Vinorelbine	760087	ME4_ME7	0.742	0.679	0.813	0.774	0.727
Selumetinib	741078	ME4_ME7	0.715	0.622	0.813	0.758	0.698
Trametinib	758246	ME0_ME8	0.697	0.856	0.482	0.712	0.652
Rapamycin	606698	ME7_ME15	0.695	0.712	0.688	0.746	0.642
Docetaxel	628503	ME6_ME8	0.682	0.737	0.654	0.764	0.570
Irinotecan	728073	ME0_ME7	0.649	0.563	0.760	0.709	0.628
Pemetrexed	698037	ME7_ME15	0.643	0.635	0.663	0.625	0.672
Belinostat	726630	ME4_ME15	0.635	0.375	0.900	0.752	0.616
Vinblastine	90636	ME7_ME15	0.617	0.581	0.674	0.626	0.626

**Table 4** – Performance of SVMs when there were 2 support vectors, the active threshold was >7, percent data used for training was 80%, and the number of cycles was 10,000

NAME	ID	MEs	Accuracy	Sensitivity	Specificity	PPV	NPV
Dabrafenib	764134	ME4_ME9_ME12	0.941	0.676	1.000	0.836	0.930
Encorafenib	778304	ME1_ME9_ME14	0.943	0.668	1.000	0.824	0.933
Dasatinib (Salt)	759877	ME9_ME13_ME15	0.941	0.663	1.000	0.818	0.930
ARRY-162	764042	ME1_ME9_ME13	0.940	0.662	0.999	0.802	0.931
Cobimetinib (XL-518)	768068	ME1_ME4_ME9	0.944	0.674	1.000	0.826	0.933
Dasatinib	732517	ME1_ME4_ME6	0.939	0.644	0.999	0.801	0.928
Cytarabine	287459	ME2_ME9_ME13	0.941	0.681	1.000	0.836	0.930
Rapamycin (Salt_1)	758664	ME7_ME14_ME15	0.650	0.650	0.665	0.643	0.667
Cobimetinib (GDC-062)	778590	ME4_ME5_ME7	0.733	0.650	0.822	0.759	0.724
Everolimus	733504	ME5_ME8_ME11	0.778	0.935	0.476	0.782	0.734
Temsirolimus	683864	ME4_ME7_ME10	0.634	0.683	0.607	0.617	0.674
Rapamycin (Salt_2)	226080	ME4_ME5_ME6	0.715	0.853	0.545	0.713	0.735
Vinorelbine	760087	ME2_ME6_ME10	0.837	1.000	0.000	0.837	0.000
Selumetinib	741078	ME0_ME2_ME7	0.873	0.685	0.930	0.719	0.910
Trametinib	758246	ME4_ME7_ME10	0.763	0.680	0.847	0.814	0.733
Rapamycin	606698	ME1_ME3_ME7	0.704	0.729	0.697	0.702	0.729
Docetaxel	628503	ME4_ME7_ME13	0.855	1.000	0.000	0.855	0.000
Irinotecan	728073	ME0_ME8_ME14	0.728	0.866	0.546	0.741	0.717
Pemetrexed	698037	ME0_ME8_ME12	0.856	0.000	1.000	0.000	0.856
Vinblastine	90636	ME6_ME8_ME13	0.690	0.709	0.704	0.775	0.592
Belinostat	726630	ME4_ME14_ME15	0.652	0.533	0.815	0.747	0.661

**Table 5** – Performance of SVMs when there were 3 support vectors, the active threshold was >7, percent data used for training was 80%, and the number of cycles was 10,000

#### **IV. Closing Remarks**

#### **Citations**

1. Holbeck, S. L., Collins, J. M. & Doroshow, J. H. Analysis of Food and Drug Administration–Approved Anticancer Agents in the NCI60 Panel of Human Tumor Cell Lines. *Mol. Cancer Ther.* **9**, 1451–1460 (2010).
2. Reinhold, W. C. *et al.* CellMiner: A Web-Based Suite of Genomic and Pharmacologic Tools to Explore Transcript and Drug Patterns in the NCI-60 Cell Line Set. *Cancer Res.* **72**, 3499–3511 (2012).

3. CellMiner - Datasets. <https://discover.nci.nih.gov/cellminer/datasets.do>.
4. Sherman, B. T. *et al.* DAVID: a web server for functional enrichment analysis and functional annotation of gene lists (2021 update). *Nucleic Acids Res.* gkac194 (2022)  
doi:10.1093/nar/gkac194.
5. Kanehisa, M. & Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **28**, 27–30 (2000).
6. Zhang, B. & Horvath, S. A general framework for weighted gene co-expression network analysis. *Stat. Appl. Genet. Mol. Biol.* **4**, Article17 (2005).
7. Albert, R. Scale-free networks in cell biology. *J. Cell Sci.* **118**, 4947–4957 (2005).
8. Yip, A. M. & Horvath, S. Gene network interconnectedness and the generalized topological overlap measure. *BMC Bioinformatics* **8**, 22 (2007).
9. Langfelder, P. & Horvath, S. Tutorial for the WGCNA package for R II. Consensus network analysis of liver expression data, female and male mice. (2014).