iSOM-GSN: An Integrative Approach for Transforming Multi-omic Data into Gene Similarity Networks via Self-organizing Maps

Nazia Fatima

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

https://scholar.uwindsor.ca/etd/7807

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.
iSOM-GSN: An Integrative Approach for Transforming Multi-omic Data into Gene Similarity Networks via Self-organizing Maps

By

Nazia Fatima

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2019

©2019 Nazia Fatima
iSOM-GSN: An Integrative Approach for Transforming Multi-omic Data into Gene Similarity Networks via Self-organizing Maps

by

Nazia Fatima

APPROVED BY:

__________________________
S. Ananvoranich
Department of Chemistry & Biochemistry

__________________________
P. Moradian Zadeh
School of Computer Science

__________________________
L. Rueda, Advisor
School of Computer Science

September 12, 2019
DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office and that this thesis has not been submitted for a higher degree to any other University or Institution.
ABSTRACT

Deep learning models are currently applied in diverse domains, including image recognition, text generation, and event prediction. With the advent of new high-throughput sequencing technologies, a multitude of genomic data has been generated and made available. The representation of such data using deep neural networks, or for that matter, application of differential analysis has, however, not been able to match the growth of that data. One of the main challenges in applying convolutional neural networks on gene interaction data is the lack of understanding of the vector space domain to which they belong and also the inherent difficulties involved in representing those interactions on a significantly lower dimension viz Euclidean spaces. These challenges become more prevalent when dealing with various types of “omics” data with different forms.

In this regard, we introduce a systematic, and generalized method, called iSOM-GSN, used to transform multi-omic genomic data with higher-dimensions into a two-dimensional grid. Afterwards, we apply a convolutional neural network (CNN) to predict disease states of various types. Based on the idea of the Kohonen’s self-organizing map (SOM), we generate a two-dimensional grid for each sample for a given set of genes that represent a gene similarity network (GSN). The set of genes that are significantly highly mutated across the whole genome, are related to each other based on functional interactions. We then test the model to predict breast and prostate cancer stages using gene expression, DNA methylation, and copy number alteration, yielding accuracies in the 94-98% range for tumor stages of breast cancer and calculated Gleason scores of prostate cancer with just 14 input genes for both cases. To our knowledge, this is the first attempt to use self-organizing maps and convolutional neural networks on integrating high-dimensional multi-omics data. The scheme not only outputs nearly perfect classification accuracy, but also provides an enhanced scheme for visualization, dimensionality reduction, and interpretation of the results.

Keywords: Gene interaction networks, self-organizing maps, convolutional neural networks, multi-omics data.
DEDICATION

I would like to dedicate this thesis to my dear brother, father, and especially to my mother, who always persuaded me to pursue my Master’s studies and encouraged me in difficult situations.
ACKNOWLEDGMENT

I would like to express my deepest appreciation to my supervisor, Dr. Luis Rueda, for his perpetual support, patience, and inspiration during my Master’s studies at the University of Windsor. It has been such an honour for me to work under your supervision, and also be part of your research team. Thank you so much for giving me the opportunity to learn from you. Secondly, I would also like to express gratitude to my thesis committee members, Dr. Sirinart Ananvoranich and Dr. Pooya Moradian Zadeh, for their beneficial advice and suggestions to my thesis. I would also like to express my special thanks to my colleagues, specifically, Dr. Abed Alkhateeb, Osama Hamzeh and Johan Fernandes, for helping me in various aspects of the research and my life as a student, and to my friends at the University of Windsor for supporting me during past two years.

Finally, I would like to express my greatest appreciation to my family for all unconditional love, support, patience, encouragement, and kindness they gave me during my whole life.
# TABLE OF CONTENTS

DECLARATION OF ORIGINALITY ........................................... III
ABSTRACT ........................................................................ IV
DEDICATION ....................................................................... V
ACKNOWLEDGMENT ............................................................ VI
LIST OF TABLES .................................................................. IX
LIST OF FIGURES ............................................................... X

1 Introduction ................................................................... 1
  1.1 Terminology ............................................................. 4
  1.2 Problem Statement .................................................... 5
  1.3 Motivation ............................................................... 6
    1.3.1 Relationship with Machine Learning ......................... 6
  1.4 Contributions .......................................................... 7
  1.5 Chapters to Follow .................................................... 7

2 Literature Review .......................................................... 9
  2.1 Dimensionality Reduction ............................................ 9
  2.2 Supervised Learning .................................................. 11
  2.3 Unsupervised Learning ................................................. 11
  2.4 Multi-Omics Data Integration ....................................... 12
  2.5 Deep Learning ........................................................ 13

3 Neural Networks ............................................................ 15
  3.1 Artificial Neural Networks .......................................... 15
  3.2 Convolutional Neural Networks .................................. 18
    3.2.1 Convolution Layer ............................................. 20
    3.2.2 Pooling .......................................................... 23
    3.2.3 Fully Connected Layer ....................................... 24
  3.3 Kohonen’s Self-Organizing Maps ................................. 25
  3.4 Conclusion ............................................................. 26

4 Proposed Method .......................................................... 27
  4.1 Process ................................................................. 28
  4.2 Gene similarity Network ............................................ 30
  4.3 Integrating Multiple Data Types .................................. 32
  4.4 Classification ........................................................ 34
    4.4.1 Network Architecture ......................................... 35
    4.4.2 Network Training ............................................. 38
LIST OF TABLES

1  Distribution of number of samples per class in PRCA dataset. . . . . 39

2  Indexes for genes in both PRCA and BRCA datasets. . . . . . . . 53
LIST OF FIGURES

1. Analogy of a neural network. The biological neural network has been adopted from [88]. 16
2. Basic architecture of the perceptron algorithm without bias. 17
3. Block diagram of an artificial neural network. 18
4. Block diagram of the architecture of a typical convolutional neural network. 19
5. Sample image in matrix form. 21
6. Example of filter or kernel. 22
7. Sample of a feature map: new matrix after convolution. 22
8. Schematic view of the Max-pooling operation in a CNN. 24
9. Example of a self-organizing map. 26
10. Block diagram of the main components of iSOM-GSN. 28
11. Diagram depicting the step-by-step process of iSOM-GSN. 29
12. Sample gene similarity network for BRCA after 1500 epochs. 32
13. Sample image after integrating multiple data views for PRCA of 12 genes. 33
14. Schematic diagram of the CNN network architecture. 37
15. Interpretation of MutsigCV. Source:http://software.broadinstitute.org 40
16. Flowchart depicting the workflow followed by MutsigCV. 41
17. Confusion matrix for email classification. 43
18. Confusion matrix scheme for two classes. 44
19. Confusion matrix scheme for three classes. 44
20. Schematic view of k-fold cross validation. 47
21. GSN obtained after 5 epochs for PRCA. 50
22. GSN obtained after 100 epochs for PRCA. 50
23. GSN obtained after 200 epochs for PRCA. 50
CHAPTER 1

Introduction

Machine learning methods are approaches used to learn functional relationships from high dimensional data [13]. They can be broadly categorized into supervised and unsupervised algorithms. The goal of the algorithms in supervised learning is to predict either the label or response of a data point by providing a labeled training dataset, while in unsupervised learning the goal is to learn inherent patterns within data. The ultimate aim in all machine learning algorithms is to optimize the model performance not only on the training data but also on the generalized dataset. In computational biology, which is still a naive field with respect to data and process involved therewith, machine learning is a lucrative approach as it has the ability to predict without the need of assumptions about underlying processes and data.

Furthermore, deep learning approaches are a class of machine learning algorithms that can reduce dimensionality and perform prediction tasks [49]. Deep learning is an umbrella term which comprises of various approaches that together demonstrate breakthrough results against the best in class machine learning algorithms. It is currently one of the most active fields in machine learning and shown to improve performance in image recognition (nontextual data). Like machine learning, deep learning can also be categorized into supervised and unsupervised applications. Large neural networks, the main form of deep learning approaches are the building blocks of machine learning that can predict and perform dimensionality reduction.

A very well-known technique in machine learning is the self-organizing map (SOM) [1], which is an unconventional form of unsupervised artificial neural networks that are trained on unlabeled data. SOMs are used to reduce dimensions of the data,
while providing with helpful visualization of high-dimensional data. They differ from conventional neural networks as they apply competitive learning instead of error correction learning. Thus, they possess the ability to preserve the topological structure of the data too.

In addition, DNA is a molecule that holds blueprint or instructions to develop and direct activities of all living organisms. Each strand is made up of a sequence of nucleotide combination of nucleic acids A, T, C and G or what we call the “alphabet” of DNA. Gene expression [82] is a process by which the instructions in DNA are used to create protein molecules. Changes in the DNA sequence are referred to as (genetic) mutations. Although they may not be harmful always, occasionally, a change in a single nucleotide itself results in fatal diseases such as sickle cell disease, cystic fibrosis, Tay-Sachs disease and various types of cancer, to name a few.

On the other hand, cancer is a leading cause of death worldwide [38]. Its identification and prognosis is a major issue and to date, many algorithms have been devised that enable quantitative and qualitative detection. Also, the advent of next-generation sequencing data yields the possibility of diagnosing cancer more accurately, facilitating making more informed decisions in precise diagnosis, prognosis, and treatment of the disease.

In this regard, prostate cancer is a complex disease that begins in the cells of the prostate gland. Adenocarcinoma, urethral carcinoma, sarcoma, and small cell carcinoma are the various types of prostate cancer, in general. Adenocarcinoma is the most common type of prostate cancer [87]. More often than not, Gleason classification is used to grade prostate cancer. Gleason scoring looks at the pattern or arrangement of the cancer cells in the prostate and rates them based on how normal or abnormal the cell arrangement is. Gleason patterns 1 and 2 are considered as normal, and 3, 4 and 5 as cancerous. Since the pattern is not unanimous across the whole gland, pathologists usually report the two most common patterns observed predominantly. For example, if the pathologist found that most of the cells are following pattern 4 and some cells are following pattern 3, then the score of $4+3=7$ is reported. Gleason scores that range from 6 to 10 are considered as cancerous, whereas scores from 9 to
10 are considered as being very abnormal [65].

To introduce how the method for predicting prostate cancer works, we need to introduce a few biological terms and entities that we describe below for the sake of brevity.

- **Biomarkers**: These are biological conditions or substances that help us ascertain the state of a biological entity or classify biological properties based on that state. Since cancer is usually progressive in nature, biomarkers play a vital role in the study of cancer progression. In this thesis, we focus on biomarkers that are necessary to classify clinical aspects of prostate cancer and breast cancer [26].

- **Gene Expression**: This is the process by which information from a gene (DNA) is used to convert it into a functional product, i.e., proteins.

- **DNA Methylation**: It is a process in which a methyl group is added to a DNA molecule, which leads to repression of gene transcription that eventually inhibits protein formation [82].

- **Copy Number Alternation**: This is a phenomenon in which segments of the genome are duplicated and the number of repetitions varies. These somatic changes result in gaining or losing a DNA fragment; they represent the most common alterations of cancer and other diseases [63].

A model can predict cancer types from unknown data based on previously trained data of known and homogeneous classes. The fundamental limitation of using a single omic (e.g., only gene expression) based study is that it fails to obtain robust and highly predictive classifications across heterogeneous datasets. Besides, the model’s accuracy decreases when the classifier is trained with a dataset of one study and tested with a dataset from another study, i.e., when the study was conducted with data for the same class. This leads to the hypothesis that adding other types of omic data such as DNA methylation and copy number variation data (i.e., multi-omics data) can predict clinical features more accurately and enhance the prediction model toward
better interpretation of the results. On the other hand, a gene interaction network is a graph of genes that are known to have a functional relationship among each other, either by physical interactions or through their gene products, i.e., proteins.

Although attempts have been made to use both conventional machine learning approaches (such as classification and clustering), and deep neural networks such as conventional neural networks (CNN) on multi-omics, data integration is still in its infancy. Although former methods used CNN for classification, they lack adequacy to generalize it for specific clinical aspects. Also, this approach cannot be applied to data integrated from various classes and/or sources.

In this thesis, we propose a deep learning-based method which we call iSOM-GSN to predict disease states, which begins with integrating multi-omic data and then leveraging the unsupervised learning abilities of a self-organizing map (SOM) to identify gene similarity networks (by the use of gene expression data). This data is then fused with other genomic data (multi-omic features) to improve prediction accuracy and help visualization. To our knowledge, this the first deep learning model that uses both SOMs and CNNs for representation learning with multi-omics data as input.

1.1 Terminology

In current biomedical research, it is typical to have access to a large amount of data, including clinical information, genetic information, histories and therapeutics from a single patient. Omics data such as transcriptomics, proteomics, metabolomics and genomics data are all can be termed as multi-view data [48]; such data provide complementary information to characterize a biological object, system or aspects. It can be from different types and can originate from different sources. They can follow different statistical distributions and if not always but mostly represent different aspects of the same cohort. Specifically, four types of data are of our interest:

1. Multi-class: Data pertaining to different groups but with the same feature set.
1. INTRODUCTION

2. Multi-view data with the same cohort/groups but different aspects of the same samples, i.e., distinct feature sets.

3. Multi-view data of the same set of samples with same set of features but at a different condition.

4. Multi-view data with distinct features and distinct samples in the same domain.

Of the above, Items 2 and 4 are often referred to as multi-omics data.

Single-omics Data

On the other hand, single-omics data describes a biological process at a single specific molecular level. For example, RNA-seq alone can be used to capture gene expression values across the whole genome (exome), which can be used to identify alternative splicing or single nucleotide variations (SNV) that are responsible for mutations and eventually cancer. Other epigenetic modifications such as DNA Methylation and CNA can be captured and used to identify other abnormalities or epigenetic changes that may lead to cancer.

Differentiation between single-omics and multi-omics data

Single omics data at any given point of time exhibit the following characteristics viz, are high dimensional in nature, redundancy, high correlation and non-negative. When it comes to muti-omics data their characteristics are heterogeneity, causality and mutually complementary.

1.2 Problem Statement

In this thesis, we address the following problem. Given a cancer data-set of patients with various clinical features, the aim is to achieve the following with a high degree of accuracy:

1. Classify samples based on different clinical features, and
2. Identify potential biomarkers which are specific to these clinical features.

1.3 Motivation

Precision medicine is an emerging field in biomedical sciences, where medical decisions, treatment, and practices are tailored for individuals or specific groups of individuals [58]. The main approaches used in the field rely on identifying epigenetic changes that are specific to patients; these changes help detect how each patient or group of patients is unique and what factors make them vulnerable to certain diseases (or class of diseases). Deducing genomic markers using multi-omics data has played an important role in precision medicine in oncology and other chronic diseases such as asthma, diabetes, and mental disorders [50, 51, 52, 79, 80, 85]

Using precision medicine models, patients can be grouped based on their genetic variations and treatment can be tailored based on similar genetic characteristics. This creates a need for developing algorithms and/or models that make use of multi-omic data which can predict or categorize to a desired degree of accuracy.

1.3.1 Relationship with Machine Learning

Given that machine learning methods are highly dependent on the choice of data they are applied on, learning representations of the data makes it easier to extract useful information when building classifiers. A good representation is needed to understand the underlying relationships and use “that understanding” to yield abstract and ultimately more efficient representations. Among various methods of learning, we focus on deep learning methods that use non-linear transformations with the goal of yielding an abstract and useful representation at the classification stage.

Network-based methods take “the knowledge of relationships among genomic data” into account to generate more stable biomarkers than linear ones, and hence can accurately predict the outcome of cancer patients across different datasets [4, 6, 14, 17, 20, 83]

Further to this, studies have indicated that different protein and gene interactions
play an important role in cancer molecular mechanisms and hence network-based biomarkers with multi-omic data provide more knowledge (related information) for prediction of cancer stages.

These findings motivated us to detect biomarkers to effectively predict and diagnose patients of different cancer stages. These findings could potentially be used to investigate different treatments or by specific drugs and different types of therapies.

1.4 Contributions

In this thesis, we propose a novel deep learning approach to perform representation learning and classification of cancer patients and to identify relevant biomarkers for specific states of disease. The main contributions can be summarized as follows:

- A deep learning method for prediction of tumor aggressiveness and progression using SOM-CNN.
- An approach that integrates multi-omics data.
- A new strategy to derive gene similarity networks via self-organizing maps.
- Use of iSOM-GSNs to identify relevant biomarkers without handcrafted feature engineering.
- An enhanced scheme to interpret and visualize multi-dimensional, multi-omics data.
- An efficient model for graph embedding and representation learning.
- A novel deep learning method for classification of cancer by converting the initial data-set into an intermediate form.

1.5 Chapters to Follow

We have so far introduced basic concepts and related important terminology, discussed types and importance of multi-omics data, our motivation, problem statement, and
1. INTRODUCTION

contributions. The rest of the thesis is organized as follows:

- Chapter 2 presents various approaches and works done to date.
- Chapter 3 explains the concepts that we have used in this thesis.
- Chapter 4 introduces our approach and model at a high level.
- Chapter 5 presents the experimental results obtained from our approach.
- Chapter 6 discusses our method in detail and further discussion of the findings.
- Chapter 7 presents the conclusion and future work.
CHAPTER 2

Literature Review

Deep learning algorithms have revolutionized various fields in past few years; they emerged with image classification in 2012 [44], by breaking records to half the error rate in computer vision community along with speech recognition [31]. In addition, deep learning has been proved to surpass other conventional machine learning techniques in natural language processing [15], especially answering questions [10], analyzing sentiments and topic classification and translation of language [37].

2.1 Dimensionality Reduction

For dimensionality reduction various methods have been devised; the following are some of the works that have been done in this regard. In 2018, a method proposed by Lyu et al. [55] applies a heatmap as a dimensionality reduction scheme on gene expression data to deduce biological insights and classify. However, the accuracy using this method limits to 97% with the Pan-Cancer dataset and lacks benefits of integrated multi-omics data. As part of their method, they used normalized level3 RNA-Seq data of 33 tumor types of the Pan-Cancer Atlas [61]. The data was again normalized using a log transformation since a few gene expression values were less than one while others were enormously large. Then, variance threshold was applied to remove gene values that remain almost unchanged across the whole dataset. The data was then embedded into 2D images, first based on chromosome number and then padded by zeros to form a square matrix. Afterward, all images are normalized to be in the range \([0,255]\).
Then, based on the “Guided Grad-cam” idea, they let the training images go through the CNN. Activation maps and gradient maps are recorded in the training phase, followed by generating a heatmap for each sample using guided grad-cam. These images are then normalized to obtain class-specific heatmaps. As a result, in the final classification, higher intensity in the heat map represents a higher contribution to final classification.

Principal component analysis (PCA) is a dimensionality reduction technique, which is often used to transform a dataset with of high dimension into a smaller one while keeping the information preserved in the lower dimension as much as possible.

In 2009, a study by Shen et al. proposed algorithms iCluster [68] and iCluster+ [69], which make use of the latent variable model and PCA.

iCluster is a joint latent variable model-based clustering method. Its main idea is to jointly estimate the tumor subtype based on multi-omic data such as mRNA expression data, DNA Methylation data, and DNA copy number data.

The integrative model is described mathematically as follows:

\[ X_m = W_mZ + e_m, \]  

where \( m \) is the number of genomic types available, \( Z \) is the latent component that connects the various types of data, \( e \) is the error term, and \( W \) is the coefficient matrix.

They derive a likelihood based solution of Equation (2.1) by using a latent continuous parameterization. The log-likelihood function of the data is then applied. The expectation maximization (EM) algorithm is subsequently applied to obtain maximum likelihood estimates of \( W \).

This is considered a much more efficient approach than directly maximizing the marginal likelihood. Afterward, a sparse solution of \( W \) is achieved by penalizing the complete data log-likelihood.

Finally, iCluster generates a single integrated cluster based on inferences from multi-omic data types. The procedure is applied to both breast and lung cancer data.
2. LITERATURE REVIEW

types and potentially novel subtypes are generated.

On the other hand, iCluster+ uses a two-step approach. In the first step, given \( k \) latent variables, it estimates the LASSO (least absolute shrinkage and selection operator) penalty parameter that minimizes the Bayesian information criterion and then chooses the best \( k \) based on deviation ratio.

2.2 Supervised Learning

In supervised learning, the model aims to learn from a set of training samples, and the goal is to predict the class label; i.e., classification or response. The ultimate goal is to optimize general performance and not just for the dataset at hand, but a generalized prediction. With this aim, the data is randomly split into three subsets i.e., training, test and validation sets.

The training set is used for learning, the validation set is used to select the best model and the test set is used at the end to estimate the generalization performance of the model. The aim to find a perfect balance between model flexibility and the amount of training data to avoid overfitting.

Many applications of supervised learning are available in computational biology, one among many is to predict the viability of cancer cell lines when exposed to selected drugs [59].

In 2012 a study by [60], makes use of multi-omic data and aims to cluster cancer data into subtypes. Even though the model performs well, it does not solve the classification problem using multi-omics data.

To the best of our knowledge, there is no work done by using multi-omic data and classification.

2.3 Unsupervised Learning

In unsupervised learning, the goal is to learn inherent patterns within the data without pre-existing labels. One of the works is Mol2vec [36], which is an unsupervised
approach to learn vector representation of molecular structures. The model is inspired by natural learning processing techniques and resulted in high-quality embeddings of molecules.

Another work is clustering of single-particle cryo-electron microscopy (cryo-EM) data, which applies hierarchical clustering to yield subtle structural differences among classes and high-resolution visualization [84].

In 2015, trans-species learning [12] was proposed and used deep belief networks (DBN) to predict phosphorylation states of humans, where the training and test sets contained rat cells. Model organisms are usually used in biomedical research and drug discovery, i.e., tests are made on model organisms such as rats and the response is learned to predict simultaneous response in humans. The model significantly outperforms state-of-the-art classification algorithms with AUROC of 0.93.

2.4 Multi-Omics Data Integration

It is a challenge for machine learning experts and data scientists to wisely use multi-view data for specific needs. In accordance with when these data are incorporated into machine learning, fusion techniques have emerged.

Integrative methods can be conducted in various ways. The following are some of the methods that have been proposed:

- Feature Concatenation: It is the simplest approach, and consists of concatenating all features into a single vector and then applying feature selection. The approach may improve performance as demonstrated in [16, 53, 76].

- Ensemble Learning: It is an optimized version of a decision tree that avoids overfitting by bagging [11] and boosting [40]. However, this approach is costly with multi-view data [19, 57]. Random forest resolves this issue by picking features randomly. It has been proved that random forests are resistant to outliers [27].

- Network-based Methods: Multi-view data can be integrated using network fusion, which is essentially a non-linear model. An example is similarity network
2. LITERATURE REVIEW

fusion (SNF) [78], which integrates mRNA, DNA methylation and miRNA data to classify cancer types and predict survivability. SNF constructs a network of samples for each data type and then efficiently fuses them into one network which represents the whole spectrum of the underlying data.

- Multi-view Matrix: The main idea of this method is to extract new features from each data view and then incorporate these features to perform classification and clustering. Some examples are principal component analysis (PCA) [39, 77], factor analysis (FA) [42, 81], and non-negative matrix factorization (NMF) [23, 45, 46, 47].

- Multi-modal Learning: The basic idea of this method is to select a sub-network for each view and integrate the outputs of individual networks to a higher layer. Examples of these are deep belief net (DBN) [32], and the deep Boltzmann machine (DBM) [66].

2.5 Deep Learning

Deep learning is one of the most active fields in machine learning and has been shown to improve performance in image recognition [33], natural language processing [8], and most recently in computational biology.

More recently, deep learning has shown promising results in predicting activity of potential drugs [56], and reconstructing brain circuits [30]. Other applications of deep learning are in computational chemistry [25], dermatology [21] and in high-energy physics, among others [9].

Deep learning has also opened the door for opportunities to model biological processes that can eventually predict how these processes are disrupted, with significant progress in genomics and imaging. Gene expression technologies have given rise to abundant transcriptomic data that can represent the underlying state of the system and can be used to study how the system reacts, deduce similarity and subtypes which can then be used for drug fabrication and treatment [67].
Many approaches have already applied deep learning with various aims on gene expression data. Denoising autoencoders have been used to pre-processing gene expression data by use of deep networks, which are then used to cluster gene data into known modules in cell cycle processes [28]. Another application of denoising autoencoders is in recognizing patterns from genomic data of microbial species-specific to Pseudomonas aeruginosa [74, 75]. These unsupervised approaches obtain insights which would otherwise be overlooked or difficult to discover.

DeepChrome [70] uses a CNN to predict gene expression values from histone modifications data and which is accurate and but also reveal novel epigenetic interactions. AttentiveChome [71] adds an attention-based deep learning approach to the former making it more accurate.

Large scale projects such as the Cancer Genome Atlas (TCGA) contains a plethora of multidimensional data obtained by applying high-resolution microarrays and next-generation sequencing. With the accumulation of these diverse multidimensional data, the need for methods to integrate and analyze data arises.

An example of it is deepDriver [54], which predicts candidate driver genes based on mutation-based feature and similarity networks. That method achieves an accuracy of 98% for predicting driver gene. However, there is no work done on integrating multi-omic data using deep learning-based methods.

So far We have seen how machine learning and deep learning were used in various domains and also in genomics. In the next chapter, we discuss concepts in detail which are used in our method.
3.1 Artificial Neural Networks

An artificial neural network (ANN) is a processing algorithm, which is initially inspired by the neuronal structure of the human brain on a smaller scale. The human nervous system contains cells which are referred to as neurons. These neurons are connected to others by dendrites and axons as shown in Figure 1. The ANN is a collected set of simple connected processors called neurons, each producing a significant activation, as shown in Figure 1. Input neurons are activated via sensors perceiving the environment, while the others are activated through connections from other active neurons. The depth of a neural network is the number of hidden layers and width in one of its layers. With the advent of high-performance hardware and technology, it has become possible to train networks with a large number of hidden layers. This has caused that ANNs have been re-branded as deep networks or deep neural networks.
The basic or simple instantiation of a neural network is called *perceptron*. This neural network contains an input layer and an output node as shown in Figure 2, where the input are training variables or features and the output contains the value of an observed variable. The edges to the input variables consist of weights for each input with which the features are multiplied and added to the output variable. Then, the *activation function* is applied to convert real values to either $-1$ or $+1$, which is appropriate for binary classification. We can use different types of activation functions to simulate different types of models. It is to be noted that even though the input variables in the first layer are not involved in any computation, they are still considered as a layer, and hence a simple perceptron is considered as a single layer neural network.
Contrary to the single layer, a multilayer perceptron or multilayer neural network contains more than one computational layer. In this case, the network contains input and output layers as default, although the only computation-performing layer is the output layer. Multilayer neural networks, as the name suggests, contain multiple computational layers. All layers between the input and output layers are called as hidden layers (shown in Figure 3) because the computations performed are not visible to the user. Unlike single-layer perceptrons, in this network the loss function is a complex function of the weights in previous layers. The gradient is computed using the backpropagation algorithm, which conforms a two-step process. The first one is the forward phase in which the output is computed based on the input values and current weights for each input variable. The actual output value is then compared to the predicted value and a derivative of the loss need to be propagated back through all the layers. In the backward phase, the gradient of the loss function is learned with respect to different weights by using the chain rule of differential calculus; the
following equations are used to update the weights.

\[ w_{ij}(t + 1) = w_{ij}(t) - \eta \frac{\partial C}{\partial w_{ij}} + \xi(t) \]  

(3.1)

where \( \eta \) is the learning rate, \( C \) is the loss function and \( \xi(t) \) a stochastic term.

In the learning phase of an ANN, the predicted label is compared to the actual label to compute the loss. It then uses the backward propagation algorithm to modify the weights to reduce the loss to the minimum. The loss function is optimized by the gradient descent algorithm. In each step, the weights are adjusted accordingly.

![Block diagram of an artificial neural network.](image)

**Figure. 3:** Block diagram of an artificial neural network.

### 3.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are inspired by cognitive neuroscience. Hubel and Wiesel’s originally worked on the cat’s visual cortex and discovered that they have simple neurons which respond to small motifs in the visual field and complex ones that respond to bigger ones [86]. The CNN framework is designed to work with grid-structured inputs, i.e., the one that has strong spatial dependencies. The most common grid data are images that can model input data as multidimensional arrays, by considering a two-dimensional image with three channels as RGB colors or one-dimensional sequence data. Since the dimensionality increases from three to 100s in high-resolution images, training a neural network becomes challenging. To
tackle this problem, CNNs make some assumptions on the structure of the network, thereby reducing the number of parameters to learn. A convolution layer consists of maps of multiple neurons, known as feature maps or filters, whose size is equal to the input image. Each feature map detects different features of the image, for example, detecting edges or orientation of the image. The activity of the neurons is achieved by applying discrete convolution of its receptive field, i.e., by computing the weighted sum of the input neurons and applying an activation function.

CNNs are a type of neural networks that are proven to be very effective in image classification and sequence prediction problems. In many applications, position and frequency of features are not relevant for learning and recognizing objects. Thus, the following assumption is used: the pooling layer summarizes neurons by computing, for example, by averaging or maximizing their activity resulting in better representation of feature activities. By applying the same pooling concept to smaller regions, the image is downsampled, which leads to reduced model parameters. Typically, a CNN consists of multiple convolution and pooling layers alternatively, which allow learning more abstract features. Finally, a fully connected layer follows the last pooling layer which is used for the final probabilistic prediction of the class. The number of convolution and pooling layers is decided based on the validation dataset, and are known as hyperparameters of the model. In the following, we describe each of the different layers and the way these layers are interlinked in a CNN.

![Figure. 4: Block diagram of the architecture of a typical convolutional neural network.](image-url)

The architecture of CNN is divided into three basic blocks:
• Convolution Layer

• Pooling

• Fully connected layer

3.2.1 Convolution Layer

The main purpose of the convolution layer is to extract spatial and temporal features from an image. An image is nothing but a collection of pixel values. Eventually, every image can be represented in the form of a matrix. An image from a standard camera comprises three layers - using the RGB (red, green, blue) coloring scheme stacked one over each other, i.e., their volume or depth, where each pixel is valued typically in the range 0 to 255. Convolution preserves the spatial relationships by using small squares of input data. For example, let us consider only black and white images: i.e, each pixel as a value 0 or 1. Let us also consider a 5 \times 5 image whose pixels are 0 and 1 such as the one depicted in Figure 5.
In a CNN, the parameters are organized using structural units known as *filters* or *kernels*. A filter is usually a square matrix (an example shown in Figure 6), which is usually much smaller than the actual image to which it is applied. While the depth of the filter remains the same as the image’s depth, i.e., one in our case. The convolution operation places the filter at each possible position on the image so that the filter overlaps with the image and a dot product is calculated between the image and the filter. The resulting matrix is called *feature map*, as shown in Figure 7.
3. NEURAL NETWORKS

The number of filters used in each layer depicts the capacity to control the model parameters. Each filter captures a particular type of spatial pattern in a small region of the image. Clearly, a large number of filters are needed to capture a huge variety of possible shapes (for example edges, curves) that are combined to create the final image.

A more formal definition of the convolution operation follows. If the $p$th filter in the $q$th layer has parameters denoted by the three-dimensional tensor $W^{(p,q)} = \{w_{ijk}^{(p,q)}\}$, the feature maps in the $q$th layer are represented by the three-dimensional...
3. NEURAL NETWORKS

tensor \( H^{(q)} = \{ h^{(q)}_{ijk} \} \) and indices \( i, j, k \) indicate height, width and depth of the filter, respectively. The convolution operation from the \( q \)th layer to the \((q + 1)\)th layer is defined as follows:

\[
h^{(q+1)}_{ijp} = \sum_{r=1}^{p_q} \sum_{s=1}^{p_q} \sum_{k=1}^{d_q} w^{(p,q)}_{r,s,k} h^{(q)}_{i+r-1,j+s-1,k}
\]  

(3.2)

In practice, the CNN learns the filters on its own during the training phase, though we need to specify parameters such as filter size, number of filters, and others. The convolution layer is usually interleaved with pooling and ReLU operations. Its purpose is the same as that of the activation function in traditional neural networks; i.e., in order to introduce non-linearity, ReLU is added after every convolution layer. It is an element-wise operation applied per pixel, which replaces all negative pixel values by zero.

3.2.2 Pooling

Similar to the convolution layer, pooling aims to reduce the spatial size of convolved features. It is used to decrease the computational power needed to process the data. It also reduces the dimension and contributes to extracting dominant features, which are translation (positional and rotational) independent, and hence, effectively training the model without overfitting. Max pooling takes the largest element from the rectified feature map within that window, while average pooling takes the average of all the elements in that window. Unlike convolution operations, pooling is done after every activation map. Pooling independently operates on each feature map to produce another feature map. Thus, it does not change the number of feature maps after pooling. In other words, the depth after the pooling layer is the same as the input layer. In real-life problems, max-pooling has been shown to work better than average pooling or sub-sampling, though it depends on the actual application.

Figure 8 depicts an example of a Max-pooling operation, where a \( 2 \times 2 \) matrix is the window which covers part of the actual matrix of image and considers only the largest elements, and hence, taking only the highlighting features.
So far, we have discussed the basic building blocks of a CNN. For these to work, we have to replicate and join them. Together, these layers extract useful features, then introduce non-linearity and reduce dimensionality, while learning features to scale and translation.

### 3.2.3 Fully Connected Layer

This layer functions as a traditional feed-forward network. In this layer, as the term implies, each neuron from the previous layer is connected to the next layer, and so on. The output from convolution and pooling layers depicts high-level features of the input image. These are then fed to fully connected layers for effective classification. It is also a way to learn the non-linear combination of features learned from previous layers, similar to what kernelized classification methods, such as support vector machines do. The fully connected layer then gives the probability of all classes, which always equals one. The probability of output is then converted into one of the classes. This is achieved by using an activation function as the output of a fully connected layer. The most common activation functions are Softmax, ReLU, and TANH.
3.3 Kohonen’s Self-Organizing Maps

We have seen neural network methods based on updating the weights in-order to correct errors. Competitive learning is a completely different paradigm whose goal is not to map inputs to outputs.

The Kohonen’s self-organizing map (SOM) is a variation of the competitive learning approach. It is an unsupervised model that does not need a target output, though all samples in a given dataset that follow a similar pattern are grouped together in the training phase. Training a SOM starts with initializing the weights of each node. Assume \( S_i \) is set of input vectors of size \( n \), then \( n \) random vectors are initialized. A sample vector is chosen from the training data, and every node is then examined to calculate which one’s weight is the most likely. This winning node is called the best matching unit (BMU), where the BMU value is calculated by using Euclidean distance as follows:

\[
d_i = \sqrt{\sum_{i=0}^{n} (S_i - W_i)^2}
\]  

where \( S_i \) is sample vector and \( W_i \) is weight vector.

Then, the radius of the neighborhood of the BMU is calculated, and the weights of the nodes in the neighborhood are altered to keep all similar nodes together. At time \( t \) the weight vector is adjusted by using the following equation:

\[
W(t + 1) = W(t) + \theta(t)L(t)(V(t) - W(t))
\]  

where \( L(t) \) is a predetermined learning rate and \( \theta \) is the neighborhood function.

The above step is repeated to predefined iterations or until desired convergence, i.e., the weights remain unchanged or the change is less than a threshold.

As an example, suppose that an array of values has been presented to a SOM as a three-dimensional vector, one for each dimension, i.e., RGB, and the network has learned to represent them in a two-dimensional space. Then, the output would be as shown in Figure 9. As can be observed in the image, regions of similar colors are
grouped together, i.e., regions having similar properties are grouped together.

Self-organizing maps are also known as a dimensionality reduction technique, because they provide a discretized representation of the input space on a lower dimension making it feasible to visualize. When applied to three-dimensional color data, the visual inspection often shows colors of similar grayscale are close to each other, as depicted in Figure 9.

3.4 Conclusion

So far, we have discussed how various artificial neural networks algorithms are implemented, in general. In the next chapter, we discuss in more detail how we used SOMs to obtain gene similarity networks and how we have used them as inputs to the CNN.
CHAPTER 4

Proposed Method

Over the past years, various types of data collection approaches from a variety of fields have been incorporated and have led to multiple data types in multi-omics. These interdisciplinary fields have led to precision diagnosis and treatment. Thus, our work presents an integrative approach, which can include many types of data. Although we use three different types of data here, the model can be extended to multiple types.

We consider the problem of integrating multiple types of omics data. For this purpose, we propose a three-step approach, which we call iSOM-GSN, and whose high-level steps are depicted in Figure 10. First, we create a GSN by extracting features from one data view, in our case, gene expression data. Then, for each sample, we integrate all data types by considering features extracted from the first step. Finally, we apply a CNN to perform classification and five-fold cross-validation to test the model.
4. PROPOSED METHOD

Figure. 10: Block diagram of the main components of iSOM-GSN.

4.1 Process

Our method is divided into eight steps. Each of the steps is discussed in detail in the following sections and a high-level process is described as follows (Figure 11 depicts these steps):
4. PROPOSED METHOD

Figure. 11: Diagram depicting the step-by-step process of iSOM-GSN.

1. Pre-process the data: More details on pre-processing is described in Section 4.5.

2. Select Features: As part of this step, we reduce the feature count from 16,000+ to 20, these are referred as rows and columns in the image. More details about this step are given in Section 4.6.

3. Template Creation: After reducing the number of features (in our case, number
of genes), we apply a SOM to create a template image. More details of this step are explained in Section 4.2.

4. Data normalization: We normalize the data between $[0-1]$ such that it is ready to be applied in the RGB color scheme. More details of this step are given in Section 4.3.

5. Image creation: Once we obtain the template ready, we color the template for each patient and create an image for each patient.

6. Splitting the dataset: The set of images are then divided into training and test sets.

7. Train the model: The training set is used to train the CNN model.

8. Test the model: Once the model is trained, we use the test set of the images to check the accuracy of the model. More details of this step are discussed in Section 4.4.

We assume the input data is a set of matrices $S = \{s^{(i)}_{oj}\}$, where $i = \{1, 2, 3, \ldots, n\}$ represents the samples, $j = \{1, 2, 3, \ldots, m\}$ represents the genes, and $o = \{1, 2, 3, \ldots, p\}$ represents types of data (omics). Here, $n$ is the number of samples, $m$ is the number of genes, and $p$ is the number of types of omics.

4.2 Gene similarity Network

The first step consists of creating a gene similarity network (GSN) by applying a SOM learning algorithm. In this step, we consider only one type of data, i.e., gene expression. Let $S_1 = \{s^{(i)}_{1j}\}_{i,j=1}^{n,m}$ denote one omic data where $j = \{1, 2, 3, \ldots, m\}$ represents the set of genes and $i = \{1, 2, 3, \ldots, n\}$ represents the set of samples. $S_1$ is the input to the SOM.

A SOM is a lower-dimensional representation of complex, higher-dimensional data in such a way that distances among vectors in the original space are preserved in the
new representation. A SOM is learned via an unsupervised clustering algorithm, which takes sample vectors as inputs, and groups them based on the similarities derived from the features. In our case, the input vectors to the SOM are the samples with gene expression values of all samples as features. The following are the main steps used to construct a SOM.

1. Initialize $m$ neurons, with random weights assigned to each neuron $c_k$, where $k = 1, 2, \ldots, m$.

2. Calculate the Euclidean distance between each gene $g_j$ and its neuron $c_k$, and identify the winning neuron, i.e., the neuron that has the smallest distance to its respective neuron. The Euclidean distance is calculated as follows:

$$d_j = \sqrt{\sum_{i=0}^{i=n} (s_{ij} - c_{ji})^2}, \quad (4.1)$$

where $s_{ij} = g_j$ represents the gene vector for the $i^{th}$ sample and $c_{ji}$ represents the neuron vector.

3. Suppose that $c_k$ is the winning neuron, i.e., it is the closest to gene $g_j$. Then, we update the weight of $c_k$ using Equation (4.2). The winning neuron is also known as the best matching unit (BMU).

4. Update the weights of the neurons that are in proximity to the BMU, $c_k$. To account for this, we use a neighborhood function that is defined by Equation (4.3).

5. Repeat steps 2 - 4 for $e$ iterations or until desired convergence (i.e., the weights remain unchanged or the change is less than a threshold).

6. Finally, obtain $c_m$ neurons, which represent $g_m$ genes in the two-dimensional space, represented by Equation (4.5).

$$c_k(t+1) = c_k(t) + \theta_j(t)L(t)(s_{1j}(t) - c_k(t)),$$  \quad (4.2)
where $L(t)$ is the learning rate regulation function defined in Equation (4.4).

$$\Theta(t) = \exp \left( \frac{d_j^2}{2\sigma^2(t)} \right) t = 1, 2, 3 \ldots e$$ \hspace{1cm} (4.3)

where $d_j$ is Euclidean distance from step 2

$$L(t) = L_0 \exp \left( \frac{-t}{\lambda} \right) t = 1, 2, 3 \ldots e$$ \hspace{1cm} (4.4)

where $L_0$ is initial learning rate.

$$X = (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m),$$ \hspace{1cm} (4.5)

where $(x_j, y_j)$ represents the coordinates of $g_j$.

As a result of running the training algorithm, a SOM is obtained in which the genes are organized based on their similarity, representing a GSN. This network is represented as a two-dimensional lattice whose coordinates are denoted as in Equation (4.5):

![Figure 12: Sample gene similarity network for BRCA after 1500 epochs.](image)

### 4.3 Integrating Multiple Data Types

The second step of iSOM-GSN is to integrate multiple data types. We use the GSN generated in the first step as a template image and color it by considering each data
view by following the RGB color scheme, where Red is represented by gene expression, Green by DNA methylation and Blue by CNA, as shown in Figure 10.

In our case, $S_{1j}^{(i)}$ represents gene expression, $S_{2j}^{(i)}$ DNA methylation, and $S_{3j}^{(i)}$ copy number alteration (CNA).

For each sample $s^{(i)}$, gene $g_j$ is colored as in the RGB palette, by considering the rule of Equation (4.6):

$$x_{pq} = \begin{cases} 
RGB_j^{(i)} & \text{if point } (p, q) \text{ is within certain radius of } g_j, \\
0 & \text{otherwise}
\end{cases} \quad (4.6)$$

where $R_j^{(i)} = S_{1j}^{(i)}$, $G_j^{(i)} = S_{1j}^{(i)}$ and $B_j^{(i)} = S_{3j}^{(i)}$

As a result, we obtain a set of matrices, one per each sample, defined as follows:

$$X^{(i)} = \{x_{pq}^{(i)}\} \quad (4.7)$$

where $i$ represents the number of genes.

Figure 13 represents a sample image created after integrating multiple omics for the PRCA dataset. As can be observed, various shades of colors for different genes
represent their values with respect to the three different types of omic data.

4.4 Classification

The last step of iSOM-GSN is to feed the images generated in the previous step to the CNN, to predict the state of the disease as the final output. The architecture of the CNN is proven to be the most effective method in learning visual representations. The CNN is also known to perform better than the human eye in many visual processing problems. The usage of the CNN in any method is just a variation in how the convolution and pooling layers are combined, and how the network is trained.

Ideally, the input to the CNN are images that can be represented on a two-dimensional grid structure of size $L \times B$, which are the length and breadth of an image. Each element in the input matrix corresponds to the spatial location of the pixel within the image. In a color image, a three-dimensional array of size $L \times B \times d$ is used, where $d$ is the depth, corresponding to the RGB color channels. As part of the transformation applied in the previous step, we create a three-dimensional grid by using Equation (4.7), which is ready to be input to the CNN.

In the CNN, the parameters are organized into sets of three-dimensional structural units, known as filters or kernels. Filters are usually square in dimension and the depth of the filter is always the same as the initial input. Assume that the dimensions of this filter are $F_q \times F_q \times d_q$.

The first step of the CNN is convolution, which places the filter in each possible position in the image so that it fully overlaps with the image, and performs a dot product resulting in a total of $L_{q+1} \times B_{q+1}$ possible dot products. The resulting matrix of this operation is known as feature map. Each filter captures a unique aspect of the image such as curves or edges. Therefore, a broad variety of filters are required to capture enough features to create the final image, which is achieved with a series of convolution operations, i.e., layers. Thus, the formal convolution operation is simply a dot product over the entire volume of the filter, which is repeated over all valid
4. PROPOSED METHOD

spatial positions and filters as follows:

\[
h^{(q+1)}_{ijp} = \sum_{r=1}^{F_q} \sum_{s=1}^{F_q} \sum_{k=1}^{d_q} w_{r,s,k}^{(p,q)} h^{(q)}_{i+r-1,j+s-1,k},
\]

(4.8)

where \( w^{(p,q)} \) is the \( p^{th} \) filter in the \( q^{th} \) layer. Here, \( h^1 = X^i \) is an input image.

The convolution operation is followed by pooling, which in essence implies a dimensionality reduction scheme applied to a feature map while preserving the spatial features of the image. Pooling can be of various types; in our experiments, we use Max-pooling. After various repetitions of convolution and pooling operations, the output is a fully connected layer, which is designed depending on the nature of the application (e.g., classification or regression). In our method, we use Softmax to classify the images, or equivalently, samples of patients that correspond to specific states of disease.

4.4.1 Network Architecture

The proposed network architecture used throughout our experiments for cancer stage classification is illustrated in Figure 14. The network comprises of only two convolutional layers and two fully-connected layers with a small number of neurons. Our choice of a smaller network design is motivated both from our desire to reduce the risk of over-fitting as well as to simplify the nature of the classification stage. The subsequent convolutional layers are then defined as follows:

- 32 filters of size 3 \( \times \) 3 pixels are applied to the input in the first convolutional layer, followed by a rectified linear operator (ReLU), a max-pooling layer taking the maximal value of 2 \( \times \) 2 regions with two-pixel strides and a local response normalization layer.

- The output of the previous layer is then processed by the second convolutional layer, containing 32 filters of size 3 \( \times \) 3 pixels. Again, this is followed by ReLU, a max-pooling layer and a local response normalization layer with the same hyperparameters as before.
4. PROPOSED METHOD

The following fully connected layers are then defined by:

- A first fully connected layer that receives the output of the second convolutional layer and contains 128 neurons, followed by a ReLU and a dropout layer.

- A second fully connected layer that receives the output of the first fully connected layer and again contains an output of three neurons, followed by a ReLU and a dropout layer.

Finally, the output of the last fully-connected layer is fed to a Soft-max layer that assigns a probability to each class. The prediction is made by considering the class with the maximal probability for a given image.
Figure. 14: Schematic diagram of the CNN network architecture.
4.4.2 Network Training

Aside from our use of a lean network architecture (i.e. fewer layers), we apply two additional methods to further limit the risk of over-fitting. First, we apply dropout learning. The network includes three dropout layers with a dropout ratio of 0.5 (50% chance of setting a neuron’s output value to zero). Then, we use data augmentation by taking a random input image and scale and mirror it in each forward-backward training pass. Training is done using Adam optimizer [41].

4.5 Dataset

The reference dataset of 499 patients is downloaded from the cBioPortal [62]. The dataset has approximately 60,000 features for gene expression data alone. A variance threshold of 0.2% is applied to this data, which removes all features have all zero’s or >80% as zero; this step reduced the feature set size to 16,000. Our aim is to classify patients based on Gleason [5, 29] score for PRCA and tumor stage for BRCA [5, 24, 72, 73].

The data were then normalized on a common scale for all omics, including DNA methylation and CNA data. The gene names were preserved in HUGO format and the names considered irrelevant by HUGO were removed. All the three datasets were then combined, based on patient ID; this left us with 484 patients data that have all three required omic data.

MultisigCV [43] was used to further process this data. MutsigCV identified significantly mutated genes by building a patient-specific mutation model based on gene expression and DNA methylation data. It considered the whole genome or exome sequence as input and identified genes that are mutated more often. The top 1,000 mutated genes from Multisig were considered for the rest of the experiment. More details about MutsigCV are given in the next section.

It is worthwhile to note that samples with Gleason score 7 were considered as two different classes, i.e., 34 and 43, as these two are clinically different.
4. PROPOSED METHOD

Table 1: Distribution of number of samples per class in PRCA dataset.

<table>
<thead>
<tr>
<th>Gleason Score</th>
<th>Number of Samples</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>3+4</td>
<td>147</td>
<td>34</td>
</tr>
<tr>
<td>4+3</td>
<td>101</td>
<td>43</td>
</tr>
<tr>
<td>4+5,5+4</td>
<td>139</td>
<td>9</td>
</tr>
</tbody>
</table>

Although the current work focuses on three types of data, it can be extended to include more types of data. We consider the following:

- Gene Expression
- DNA Methylation
- CNA (Copy Number Alteration)

4.6 MultisgCV

Since multi-omic data is heterogeneous, which means that it can not be combined and processed together. Also, since each omic data type has more than sixty thousand features, it is invariant to filter it out by performing feature selection. We use Mut-Sigcv [43] as a way to limit the list of genes per omic, because it detects significantly mutated genes based on genome-wide background mutation rate. A score is calculated for each gene based on the observed number of mutations and covered bases. Then, the \( p \)-value is calculated by convoluting the background distribution of all mutation types and determining the probability of meeting the score by background mutation score. The algorithmic procedure is depicted in Figure 16.
4. PROPOSED METHOD

Figure. 15: Interpretation of MutsigCV.
Source: http://software.broadinstitute.org

The algorithm takes as input three files shown below and outputs the list of significantly mutated genes ordered by \( p \)-value.

- **Mutation file**: A mutation annotated tab-delimited file that lists mutations.
- **Coverage file**: This file has information about sequencing coverage for each patient and gene.
- **Covariates file**: This file contains covariate data for each gene.
4. PROPOSED METHOD

Figure 16: Flowchart depicting the workflow followed by MutsigCV.
CHAPTER 5

Results

Evaluating any machine learning model is an essential part, validation accuracy is not the only performance metric which gives a correct view of the model performance. Thus, to gauge the performance of the model we first discuss some metrics which we have used to evaluate the proposed model.

5.1 Equipment Specifications

The study was carried out on a workstation with the following specifications:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Intel i7 4790</td>
</tr>
<tr>
<td>Frequency</td>
<td>3.6 Ghz</td>
</tr>
<tr>
<td>Memory</td>
<td>32 G</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 10 Enterprise</td>
</tr>
</tbody>
</table>

The algorithm was implemented using SimpSOM library for self-Organising maps and using Keras with TensorFlow as the backend engine for Convolutional Neural networks. A reference implementation is available at GitHub [22].
5. RESULTS

5.2 Classification and Evaluation

To evaluate our model we use various evaluation metrics to gauge the performance of the model such as categorical Accuracy, Precision, Recall, F1-score and mean absolute error.

Before understanding the definition of various performance metrics, the basic building block which is known as the confusion matrix is defined as follows.

5.2.1 Confusion Matrix

As the name suggests, it is a matrix that gives insight into model performance on a set of test data for which true values are known.

A confusion matrix is usually constructed for classification problems where output can be two or more classes. It is a table with four different outcomes as shown in Figure 18. Let us consider the scenario when the output is two classes i.e when there are only two predictive outcomes. One class is set as positive and another class as negative. When a positive sample is predicted as a positive then its called as True Positive (TP). Similarly, when a negative sample is classified as negative, it is called True Negative (TN), indicating that they are predicted as their true intrinsic value. An example of such problem is when we classify the email as Spam and Non-Spam, as shown in Figure 17. Let us assume the spam class is positive and non-spam is negative. When an email is spam and we classify it as spam it is considered as TP and when an email is non-spam and we classify as non-spam then its TN.
5. RESULTS

Similarly, when a positive sample is predicted as negative, it is called False Positive (FP), and vice-versa for False Negative (FN). In our example, spam is classified as non-spam and non-spam classified as spam, respectively. These terms, True Positive, True Negative, False Positive and False Negative, are the basis for calculating various performance metrics.
5. RESULTS

5.2.2 Accuracy

Accuracy is calculated as the total number of correctly classified samples with respect to the total samples in the subset, for multi-class classification the formula is as follows:

\[
\text{Accuracy} = \frac{TP_i}{n_i}, \quad (5.1)
\]

where \( TP_i \) is the number of samples classified correctly for each class and \( n_i \) is total number of samples for class \( i \).

5.2.3 Precision

Precision describes how accurate or precise the model is by calculating the ratio of True Positives over all cases that were predicted as positive. A lower Precision value denotes that there is a higher False Positive value. In other words, negative cases are being predicted as positive. Thus, a higher Precision value would indicate a clear classification between negative and positive cases.

\[
\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}. \quad (5.2)
\]

5.2.4 Recall

Recall records how many of the actual positives our model captures. Thus, a lower recall value indicates that the model labeled a few actual positives as negatives, i.e., False Negatives. A higher recall value denotes that the majority of the positive classes have been predicted correctly.

\[
\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}. \quad (5.3)
\]
5. RESULTS

5.2.5 F1-Score
F1-score is a function that provides the Harmonic mean of Precision and Recall and seeks a balance between them. This balance of the two ratios (Precision and Recall) helps pick the correct classifier when we evaluate multiple classifiers. Thus, when we can pick a classifier that has a good F1-score, it will by nature achieve a good Precision and recall as well. Unlike Accuracy, it works well when there is an imbalance in class distribution.

\[ F1Score_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i}. \]  (5.4)

5.2.6 Mean Absolute Error
The mean absolute error (MAE) provides a quantifiable measurement of how far off the predictions of the model are. It calculates the average of differences between actual and predicted values. Each difference is provided as an absolute value so that no error is unnoticed. To obtain the average of all error values provided by adding the absolute predicted error values we simply divide the sum by the total number of data points/cases being considered. This value gives us an estimate of how far off we are in terms of predicting the actual values in the data points. The MAE is calculated as follows:

\[ \sum_{i=1}^{n} |y_i - x_i| \]

\[ \frac{n}{n}. \]  (5.5)

5.2.7 k-Fold Cross Validation
To evaluate the performance of a model we need a good measurement and a resampling procedure to derive that measurement that will be unbiased towards a sample of the data. Using k-fold cross-validation, we split the data into a ratio of testing and training data; k is a placeholder where the user decides the number of times the data needs to be passed over. This method ensures that all patterns available in the training data are identified by the model.
5. RESULTS

Figure 20: Schematic view of $k$-fold cross validation.

For instance, we can split the data into four training subsets and one testing set. This testing set is held out while the model is fit onto the complete training set that is made up of the four training subsets, as shown in Figure 20. The model is evaluated using the test set, the scores are noted down and the model is scrapped. The same process is repeated until all data points have been considered part of the test set. Thus, each data point is part of the training set for $k - 1$ times and part of the testing set once.

Once all samples have been processed, the evaluation scores from each cycle summed and an average score is produced, which represents an unbiased and accurate evaluation of the model’s classification performance.

5.2.8 Area Under the Curve

Area under the curve (AUC) is one of most widely used metrics for evaluation. Receiver operating characteristic (ROC) is a probability curve that is plotted on a graph, where the $y$-axis is the True Positive rate (Sensitivity) and the $x$-axis is the False Positive rate (FPR). The area under the ROC curve represents the degree of separation that the model has achieved between two classes.

The higher the AUC is the better the classification performance of the model is. For instance, an AUC of 0.5 indicates that the model cannot distinguish between the classes; it is equivalent to a random classification. A lower AUC value, say 0, means
that the model has inverted the prediction values, i.e., positives predicted as negatives and vice versa. An AUC of 0.99 would be one of the most desirable, since it indicates that the classes are easily distinguished by the model.

In this scenario, as the model has multiple classes to classify using the one-vs-all technique, we can produce multiple ROCs for each one-vs-all case. The values used to play the graph are based on the confusion matrix of a two-class problem.

The following is the definition of the True Positive Rate and False Positive Rate. Both TRP and FPR have values in range of 0 to 1.

**True Positive Rate or Sensitivity**

True Positive rate (TPR) is the proportion of positive cases that are correctly identified as positive by the model among all values that are actually positive. Another name for this measurement is “Sensitivity”, and is defined as follows:

\[
Sensitivity = \frac{TP_i}{TP_i + FN_i}.
\]

(5.6)

**False Positive Rate**

False Positive rate (FPR) is the proportion of negative cases that are incorrectly identified as positive by the model among all values that are actually negative. Another name for this measurement is “Fall-out rate”. It can also be derived from \(1 - \text{True Negative Rate} \), i.e., Specificity, and calculated as follows:

\[
Specificity = \frac{FP_i}{FP_i + TN_i}.
\]

(5.7)

\[
AUC = \frac{2}{c(c-1)} \sum_{i,j=1}^{c} AUC_{ij}
\]

(5.8)

where \(AUC_{ij}\) is the AUC for classes \(i\) and \(j\)
5.3 Comparison with Other Approaches

A practical challenge for validating this framework is the unavailability of independent datasets with all data types. An intrinsic question arises as to what extent a single data type (e.g., gene expression) is effective in classifying, based on our framework. To that end, we conducted an experiment on one omic data at a time using the same approach. We discovered that for the PRCA dataset gene expression data alone yielded 90% accuracy, whereas DNA methylation and CNA yielded 87% and 89% respectively.

In general, validation is required to assess the degree to which single omic data alone is efficient. Then, it may not be necessary to collect all types of data. We must note that the proposed method can also be applied to single omic data with a different state of disease or clinical variable.

5.4 Results

As discussed in Chapter 4, iSOM-GSN is divided into three steps. In this chapter, we present the results obtained after each step. To test our approach, we considered the two datasets introduced earlier, PRCA and BRCA; each individually utilized 14 genes to create SOMs as images for each sample and then classify them. We discuss the results obtained in each step in the following sections.

5.4.1 Gene Similarity Network

We consider Gene Expression data in this step, to create a gene similarity network which acts as a template that is used in subsequent steps. Figures 21, 22, 23, 24 and 25 shows GSNs generated after running the SOM learning algorithm on the PRCA dataset for different numbers of epochs. We observe that with the increasing number of epochs the arrangement of genes takes a more sparse alignment creating a network of genes that are related based on their similarity to each other.
5. RESULTS

Figure. 21: GSN obtained after 5 epochs for PRCA.

Figure. 22: GSN obtained after 100 epochs for PRCA.

Figure. 23: GSN obtained after 200 epochs for PRCA.
Figure. 24: GSN obtained after 500 epochs for PRCA.

Figure. 25: GSN obtained after 1,000 epochs for PRCA.

The different gene arrangements after 5 epochs and 1500 epochs can be clearly observed in the figures. As can be observed, initially (i.e., after 5 epochs) they are all clustered but after 1,000 epochs the genes are organized more sparsely, based on their gene expression similarity. To bring in additional insight into the genes found by iSOM-GSN, we have investigated some of them in the literature. It has been reported that genes FOXA1 and NKX31 co-expressed together to increase prostate cancer cells survival by aiding in the progression of the prostate tumour [90].
Similarly, we have investigated the behavior of the SOM learning algorithm to create GSNs for the BRCA dataset. Figures 26 and 27 show the gene arrangements after 5 epochs and 1500 epochs. More details on the validation of these genes are given in Section 5.5.

The 14 genes which are considered in this step are indexed as per the following table:
5. RESULTS

Figure 28: Comparison of PRCA and BRCA genes.

Table 2: Indexes for genes in both PRCA and BRCA datasets.
5.4.2 Multi-omics Integration

We considered all types of omics in this step to create images which will then be used for classification. Figures 30, 31, and 32 present images created for each class for the PRCA dataset after integrating all omics, while Figure 29 represents a cumulative image for all three classes. Similarly, Figures 33, 34 and 35 represent images for the BRCA dataset after integrating all the omics.

If we observe the figures closely, from the set of images for each class, we notice that each class follows a distinct shade of color. This trend shows that each Gleason score in PRCA and tumor stage in BRCA follow similar patterns, which is very helpful to classify them. These observations demonstrate the power of the proposed scheme for integrating different types of omic data for visualization and interpretation.

Figure. 29: Cumulative training images for the PRCA dataset for all classes.

Figure. 30: Training images for the PRCA dataset for class 34.
5. **RESULTS**

Figure. 31: Training images for the PRCA dataset for class 43.

Figure. 32: Training images for the PRCA dataset for class 45.

Figure. 33: Training images for the BRCA dataset for class 2A.
5. RESULTS

Figure 34: Training images for the BRCA dataset for class 2B.

Figure 35: Training images for the BRCA dataset for class 3A.

5.4.3 Convolutional Neural Network

Once the images are created via the SOM learning algorithm, the last step of our method is to pass them through the CNN for classification. Figures 36 and 38 depict the plots that represent the graph of evaluation metrics generated for the PRCA dataset after running the algorithm for various epochs, until it converges, and the ROC curve for the final model.
5. RESULTS

Figures 36 and 37 show values of accuracy and other metrics, and how they vary through different epochs, until the network converges. We have analyzed the performance of iSOM-GSN on two multi-omic datasets, namely PRCA and BRCA. The evaluation of the performance of the classifier as per the results obtained are shown in Figure 36 for the PRCA dataset, and in Figure 37 for the BRCA dataset.

As can be observed, accuracy for PRCA and BRCA fluctuates for initial epochs.
and eventually reaches the highest peak after a number of iterations. Similar behavior is observed for precision and F1-score.

Apart from the above metrics we also calculated ROC and AUC values once the model is converged. Figures 38 and 39 shows the ROC curve for datasets PRCA and BRCA, respectively, which are plotted with TPR against FPR, where TPR is on $y$-axis and FPR is on the $x$-axis.

Figures 38 and 39 depict the graphs of ROC curve after running iSOM-GSN for both PRCA and BRCA datasets respectively.
5. **RESULTS**

![Receiver operating characteristic example](image)

Figure. 39: Plot of ROC curve after running iSOM-GSN for BRCA dataset.

### 5.5 Biological Validation

Integrative genomic studies facilitate multi-omic data with the power to classify, predict and characterize the underlying data. This work presents a framework that conducts data integration, dimensionality reduction, feature selection and classification simultaneously to harness the full potential of integrated high dimensional large scale cancer genomic data. We illustrate the ability of the proposed method to discover and visualize patterns of genomic interactions in a biological comprehensive context for classification.

When the goal is to identify potential biomarkers and factors that characterize biological and clinical aspects, the proposed approach comes to the rescue. As part of a feature selection step, we narrowed down the relevant features to just 14 genes, which are sufficient for classification and achieve 96% accuracy. These genes are listed in Table 40. All genes are identified either as tumor suppressor genes or oncogenes for known pathways as shown in Table 40. TP53 and PTEN are the common genes that are highly mutated in both PRCA and BRCA datasets. These genes are tumor suppressor genes, which are well known to express high gene expression values and are considered as biomarkers for cancer in general [35]. On the other hand, SALL1
and PITPNM2 are genes that are not known as cancer-related genes.

While validating the genes for both the datasets, for BRCA, we have found in that genes MAP3K1 and MAP2K4 are predictors of MEK inhibitors, which are frequently found in breast, prostate and colon cancers. These can be potential targets for effective drugs as reported in [89].

In another study, we have found co-occurring mutations of PIK3CA and MAP3K1 are functionally significant in breast cancer and MAP3K1 mutational status may be considered as a predictive biomarker for efficacy in PI3K pathway inhibitor trials [7].

We have also found that the expression of genes GATA-3 and FOXA1 are sufficient to differentiate breast carcinoma from others, and hence are excellent bio-markers [18]. This fact is also supported in paper [34], which claims that these two genes are associated with a less aggressive phenotype and they give better prognosis in patients with HR-positive or HER2-negative breast cancer. As a result, we confirm that the GSNs formed are helpful in finding potential novel biomarkers for breast cancer progression.

<table>
<thead>
<tr>
<th>PRCA</th>
<th>Pathways</th>
<th>BRCA</th>
<th>Pathways</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPOP</td>
<td>Signaling by Hedgehog and Hedgehog Pathway</td>
<td>RUNX1</td>
<td>Gliona and Development Dopamine D2 receptor transactivation of EGFR</td>
</tr>
<tr>
<td>TP53</td>
<td>Apoptosis Modulation and Signaling and Gliona.</td>
<td>PIK3CA</td>
<td>Apoptosis Modulation and Signaling and Gliona.</td>
</tr>
<tr>
<td>FOXA1</td>
<td>Embryonic and Induced Pluripotent Stem Cell Pathways and Lineage-specific Markers</td>
<td>TP53</td>
<td>Activated PKH1 stimulates transcription of AR (androgen receptor) regulated genes KLK2 and KLK3 and mRNA Splicing - Major Pathway</td>
</tr>
<tr>
<td>CTNNB1</td>
<td>Beta-Adrenergic Signaling and Blood-Brain Barrier Pathway: Anatomy</td>
<td>SF3B1</td>
<td>Apoptosis Modulation and Signaling and Gliona.</td>
</tr>
<tr>
<td>MED12</td>
<td>Gene Expression and RNA Polymerase II Transcription Initiation And Promoter Clearance</td>
<td>PTEN</td>
<td>Gliona and Metabolism of proteins</td>
</tr>
<tr>
<td>PTPN2</td>
<td>Glycrophospholipid biosynthesis and Metabolism</td>
<td>CBFB</td>
<td>Regulation of nuclear SMAD2/3 signaling and ATF-2 transcription factor network</td>
</tr>
<tr>
<td>PTEN</td>
<td>Gliona and Metabolism of proteins</td>
<td>CDH1</td>
<td>Akt6 trafficking events and Integrated Breast Cancer Pathway</td>
</tr>
<tr>
<td>ATM</td>
<td>Apoptotic Pathways in Synovial Fibroblasts and Integrated Cancer Pathway</td>
<td>MAP2K1</td>
<td>Apoptosis Modulation and Signaling and Tocodilimus/Cyclosporine Pathway, Pharmacodynamics</td>
</tr>
<tr>
<td>NCOX1-1</td>
<td>Endometrial cancer and Pathways in cancer</td>
<td>MAP3K1</td>
<td>Apoptosis Modulation and Signaling and Tocodilimus/Cyclosporine Pathway, Pharmacodynamics</td>
</tr>
<tr>
<td>ZMYM3</td>
<td>Diseases associated with ZMYM3 include Dystonia 3, Torsion, X-Linked and Myasthenic Syndrome, Congenital 6, Presynaptic.</td>
<td>NGOR1</td>
<td>Signaling by NOTCH1 and Transcriptional activity of SMAD2/SMAD3-SMAD4 heterotrimer</td>
</tr>
<tr>
<td>SALL1</td>
<td>Transcriptional regulation of pluripotent stem cells and Developmental Biology</td>
<td>CDKNN1</td>
<td>CDK-mediated phosphorylation and removal of Cc6b and PI3K-AKT-mTOR signaling pathway and therapeutic opportunities</td>
</tr>
</tbody>
</table>

Figure. 40: Most relevant genes found to predict Gleason groups of PRCA and stages of BRCA.
CHAPTER 6

Conclusion and Future Work

This thesis presents a new deep learning method which is used to predict disease states by integrating multi-omic data. The method, which we call iSOM-GSN, leverages the power of self-organizing maps (SOM) to transform multi-omic data into a gene similarity network (GSN) by the use of gene expression data. Such data is then combined with other genomic features to improve prediction accuracy and help visualization. To our knowledge, this the first deep learning model that uses SOMs to transform multi-omic data into a GSN for representation learning, and uses CNNs for classification of disease states or other clinical features. The main contributions and future work are discussed below.

6.1 Main Contributions

This work presents a framework that uses a self-organizing map and a convolutional neural network used to conduct data integration, dimensionality reduction, feature selection and classification simultaneously to harness the full potential of integrated high dimensional large scale cancer genomic data.

We have introduced a new way to create gene similarity networks, which can unravel new gene interactions. We have also provided a scheme to visualize high-dimensional, multi-omics data onto a two-dimensional grid. In addition, we have devised an approach that could be also used to integrate other types of multi-omic data and predict any clinical aspects or states of diseases, such as laterality of the tumor, survivability, or cancer subtypes, just to mention a few.
6. CONCLUSION AND FUTURE WORK

The main contributions of this work can be summarized as follows:

- A deep learning method for prediction of tumor aggressiveness and progression using SOM-CNN.
- A new strategy to derive gene similarity networks via self-organizing maps.
- Use of SOM-CNN to identify relevant biomarkers without handcrafted feature engineering.
- An enhanced scheme to interpret and visualize multi-dimensional, multi-omics data.
- An efficient model for graph embedding and representation learning.

6.2 Future Work

This work can be extended to classify Pan-cancer data [61]. Omics can be considered as a vector and more than three types of data can be incorporated for classification. Apart from integrating multi-omics data, the proposed approach can be considered as an unsupervised clustering algorithm, because of the competitive learning nature of SOMs. Applications of iSOM-GSN can also be drug response, prediction of passenger or oncogenes, and other prediction tasks. In addition, we can also apply iSOM-GSN on other domains, such as predicting music genre’s for users based on their music preference, discover hidden relationships among publications in citation networks, and classify protein complexes based on three-dimensional structure, among others.
REFERENCES


REFERENCES


REFERENCES


[90] Y. Zhao, D. J. Tindall, and H. Huang. Modulation of androgen receptor by foxa1 and foxo1 factors in prostate cancer. *International journal of biological...
VITA AUCTORIS

<table>
<thead>
<tr>
<th>NAME:</th>
<th>Nazia Fatima</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLACE OF BIRTH:</td>
<td>Hyderabad, Telangana, India</td>
</tr>
<tr>
<td>EDUCATION:</td>
<td>Bachelor of Engineering in Computer science, Chaitanya Bharathi Institute of Technology affiliated to Osmania University, Hyderabad, Telangana, India, 2008.</td>
</tr>
<tr>
<td></td>
<td>Master of Science in Computer Science, University of Windsor, Windsor, Ontario, Canada.</td>
</tr>
</tbody>
</table>