Deep learning models for predicting phenotypic traits from omics data

by

Md. Mohaiminul Islam
A Thesis submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada

Abstract

Computational and statistical analysis of high throughput omics data, such as gene expressions, copy number alterations (CNAs), single nucleotide polymorphisms (SNPs) and DNA methylation (DNAm) has become very popular in cancer studies in recent decades because such analysis can be very helpful to predict whether a patient has certain disease or its subtypes. However, due to the high-dimensional nature of the data sets with hundreds of thousands of variables and very small numbers of samples, traditional machine learning approaches, such as Support Vector Machines (SVMs) and Random Forests (RFs), have limitations to analyze these data efficiently. In this thesis, we propose deep neural network (DNN) based models for classifying molecular subtypes of breast cancer and DNN-based regression models to account for interindividual variation in triglyceride concentrations measured at different visits of peripheral blood samples using epigenome-wide DNAm profiles.

We collect copy number alteration and gene expression data measured on the same breast cancer patients from the METABRIC (Molecular Taxonomy of Breast Cancer International Consortium). We propose multiple DNN models for predicting their molecular subtypes, which include the status of estrogen-receptor (ER): ER+ and ER-, and the status of PAM50 subtypes: luminal A, luminal B, HER-2 enriched and basal-like. In addition, we use epigenome-wide DNAm profiles of before and after medication interventions (called *pretreatment* and *posttreatment*, respectively) to predict triglyceride concentrations for peripheral blood draws at visit 2 (using pretreatment data) and at visit 4 (using both pretreatment and posttreatment data). Our experimental results show that DNN models can predict triglyceride concentrations for blood draws at visit 4 using pretreatment and posttreatment DNAm data more accurately than for blood draws at visit 2 using pretreatment DNAm data. Furthermore, we get the best prediction

results when we use pretreatment DNAm data to predict triglyceride concentrations for blood draws at visit 4, which suggests a long-term epigenetic effect on phenotypic traits. The performance of our proposed DNN models is compared with baseline models: SVM, RF, and the DNN model fine-tuned from deep belief network (DNN_DBN). We demonstrate that our proposed DNN models are superior to SVM, RF, and DNN_DBN in terms of prediction performance.

Our experimental results show that integration of multi-omics profiles into DNN-based learning methods can improve the prediction of the molecular subtypes of breast cancer. The proposed integrative DNN-based learning frameworks are not limited to integrate only copy number alteration and gene expression data and can be extended to include many more data sources, such as methylation data and clinical data. We also demonstrate the superiority of our proposed DNN models over the SVM model for predicting triglyceride concentrations. This study also suggests that the DNN approach has advantages over other traditional machine-learning methods to model high-dimensional epigenome-wide DNAm data and other genomic data.

Acknowledgement

Foremost, I am thankful to the almighty and merciful Allah for giving me the opportunity to come here in Canada and letting me complete the M.Sc. degree requirements. His blessings enabled the power to believe in my desire and pursue my dreams.

This thesis came in this form with the support and motivation of several people and I would like to express my indebtedness to all of them.

At the very outset, I express my deepest and sincere gratitude to my master's thesis advisors, Dr. Pingzhao Hu and Dr. Yang Wang for their support of my M.Sc. study and research completion. They continuously motivated and steered me in the right direction to gain utmost expertise in research and writing of this thesis. Their doors were always open whenever I need them and they never let me down. Their insightful comments and questions during my experiments and writing of this thesis boosted my knowledge a lot. Both are very hardworking and studious professors and I always wish them to be successful in every aspect of life.

I am grateful to my lab members at The Hu Lab: Rasif Ajwad, Rayhan Shikdar, Chen Chi, Jiaying You, Ye Tian, Yan Cheng, Qian You, Svetlana Frenkel, and Nikta Feizi, for their friendship and research discussions. I would also like to wholeheartedly thank all the other faculty members and staff at the Department of Computer Science and the Department of Biochemistry and Medical Genetics for their support during my M.Sc. degree completion in many ways.

I would like to dedicate this thesis to my family. My parents and sisters (Maria and Maha) are always there whenever I need mental support. They never let me feel alone in this journey by their endless love, support, and encouragement. Thank you all for giving me the strength to reach for the stars and pursue my dreams.

I also thank that the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) and the organizers of Genetics Analysis Workshop 20 (GAW20) for providing the data sets used in this study.

Publications and Contributions

1. **Islam MM**, Ajwad R, Chi C, Domaratzki M, Wang Y, Hu P. Somatic Copy Number Alteration-Based Prediction of Molecular Subtypes of Breast Cancer Using Deep Learning Model. In Canadian Conference on Artificial Intelligence 2017 May 16 (pp. 57-63). Springer, Cham. This paper is presented in **Chapter 3**.
2. **Islam MM**, Ajwad R, Chi C, Wang Y, Hu P. OmicsNet: integrative deep learning frameworks for classifying molecular subtypes of breast cancer. Submitted to Artificial Intelligence in Medicine 2017. This paper is presented in **Chapter 4**.
3. **Islam MM**, Tian Y, Cheng Y, Wang Y, Hu P. A deep neural network based regression model for triglyceride concentrations prediction using epigenome-wide DNA methylation profiles. In BMC Proceedings 2017. In press. This paper is presented in **Chapter 5**.

I, Md. Mohaiminul Islam, designed and implemented the experiments and drafted the manuscripts. Rasif Ajwad generated the CNA data. Chen Chi and Dr. Yan Cheng helped write part of the background section related to biology. Ye Tian discussed the experiments of "A deep neural network based regression model for triglyceride concentrations prediction using epigenome-wide DNA methylation profiles" and presented the results at the GAW20 conference Dr. Yang Wang provided supportive suggestions during the course of designing and implementing the experiments. Dr. Pingzhao Hu supervised and monitored the whole project.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AUC	Area Under the Curve
CF	Convolutional Feature
CNV	Copy Number Variation
CNA	Copy Number Alteration
DBN	Deep Belief Network
DCIS	Ductal Carcinoma in Situ
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DNA	Deoxyribonucleic Acid
DNAm	DNA methylation
DNN	Deep Neural Network
ER	Estrogen Receptor
FNR	False Negative Rate
FPR	False Positive Rate
GAW20	Genetics Analysis Workshop 20
METABRIC	Molecular Taxonomy of Breast Cancer International Consortium
mRNA	Messenger Ribonucleic Acid
PCA	Principle Component Analysis
PCR	Polymerase Chain Reaction
TSS-seq	Transcription Start Site sequencing
SNP	Single Nucleotide Polymorphism
SVM	Support Vector Machine
ReLU	Rectified Linear Units
RF	Random Forest
RNA	Ribonucleic Acid
ROC	Receiver Operating Characteristics

Chapter 1

Background and Introduction

1.1. Omics data

Omics refers to use high-throughput experimental technologies to examine genomics, transcriptomics, metabolomics and proteomics for understanding biological and disease mechanisms. The omics data generated from these technologies are high-dimensional and correlated. Different computational and statistical analyses of these data can be used to identify risk factors for different diseases or to build autonomous diseases prediction models. The technological development allows researchers to have a huge amount of high-dimensional biological data. The omics technologies generate such high-throughput data by detecting numerous alterations in molecular components [1]. These technologies also generate additional biological data to comprehend different types of correlations and dependencies among the molecular components. Bioinformatics is a discipline which emerges to perform computational analysis with the high-throughput biological data. Bioinformatics offers tools and methodologies for analyzing different omics data to understand the underlying information about different diseases. Such analyses will help physicians to provide early and patient-specific treatment. Schneider and Orchard [2] list the state-of-the-art available technologies to generate omics data. They also provide the list of different available bioinformatics resources to analyze omics data and discuss the bioinformatics challenges to handle the high-throughput data.

1.2. Phenotypic traits prediction

A living biological organism can show a number of observable characteristics, such as the morphology, growth and the behavior of the organism. Phenotypes are the product of different genetical expressions of an organism. These expressions are known as the genotype of that organism. However, phenotypic traits are the alternatives of a phenotype of a particular organism. For example, hair is a phenotype but different hair colors are the phenotypic traits. Study of phenotypic traits prediction is very important as it gives us the knowledge about how genotype impacts upon an individual's diseases or traits. Lippert et al. [3] use the whole-genome sequencing data to identify individuals by predicting their biometric traits. Genome sequencing data were also used by Chen et al. [4] to build a probabilistic Bayesian model to predict dichotomous traits (e.g. Glaucoma, Corn's disease, Prostate cancer). This model incorporates annotated information about different variant genotypes and genes, which are associated with diseases. There are other phenotypic trait prediction models such as eye color [5,6], skin color [6] or facial structures [7].

1.2.1 Breast cancer and its molecular subtypes

Cancer is a disease that is characterized by uncontrolled cell growth in an organ, i.e. the site the cells originate from. Breast cancer begins in the breast tissue and may start in the duct or lobe of the breast. When the "controls" in breast cells are not working correctly, they divide continually and a lump or tumor is formed. It is a complex, heterogeneous disease at both the cellular level and molecular level, with differing prognostic and clinical outcomes. In clinical practice, breast cancer is classified based upon receptor expression. It is called estrogen-receptor-

positive (ER+) if the cancer cells, like normal breast cells, have receptors for the hormone estrogen, in which they rely on in order to promote their growth. Statistics show that approximately 67% of breast cancers test positive for hormone receptors [8]. Testing whether a patient is hormone receptor positive or negative is important in clinical diagnosis as the results help physicians to determine whether the cancer is more likely to respond to hormonal treatments or chemotherapy.

A study done in 2000 has emerged a new genomic paradigm [9] in discovering the intrinsic subtypes of breast cancer. When they looked at the gene expression profiles of breast cancers, they found that the cancers segregated into 5 clusters: luminal A and B, Normal, Basal like group and the HER-2 enriched. It started with genome-wide gene expression profiling using microarray data, and developed into a PCR (polymerase chain reaction)-based test with a curated list of 50 genes known as the PAM50 signature. The PAM50 signature measures the expression levels of these 50 genes in tumor samples, which can classify breast cancers into one of the four intrinsic subtypes (Luminal A, Luminal B, HER-2 enriched and Basal-like). This classification has been shown to be prognostically independent of clinicopathologic factors and can determine the sub-group of patients who are more likely to benefit from adjuvant chemotherapy [10].

1.2.2 Prediction of triglyceride concentration in blood

Triglyceride is a type of fat in the human blood. Having a high concentration of triglycerides in human blood can increase our risk of heart diseases, stroke, and other disorders. Many genetic loci have been identified by genome-wide association studies, but only a small proportion of interindividual variability of triglycerides has been explained by the genetic determinants. It is known that the level of triglycerides is heritable. Consequently, the

development of new high-throughput genomic technologies makes it natural to extend these phenotypic prediction models to complex traits, such as triglyceride. Using DNAm profiles to predict disease phenotypic courses has not yet been explored in detail.

1.3. Machine learning in bioinformatics

Artificial intelligence (AI) is an area of computer science which demonstrates its necessity in our everyday life by machine learning (ML) methods. ML methods can automate the data analysis and can find the hidden intrinsic patterns from big data which is impossible for a human being. ML methods use these patterns to build predictive models without any explicit programming. These predictive ML models are improving our daily life in various ways such as recommendations of different products during online shopping based on our searches of products, stock price prediction, classification of different objects from images, real-time language translation etc.

Traditional machine learning methods, such as Support Vector Machine (SVM), Random Forest (RF), Bayesian Network (BN) etc., are dependent on the well-defined, engineered and robust hand tuned features (or feature vectors) as inputs from the raw input data to make reasonable predictions. A domain human expertise is required to develop these engineered features. However, real-time biomedical data are often high-dimensional and noisy. These conventional ML methods are not capable enough to provide suitable techniques to handle such natural raw data (i.e. normalized gene expression data).

Machine learning approaches have previously been applied to identify these molecular subtypes (such as PAM50 subtypes) of breast cancer using microarray-based gene expression

profiles [11]. But, a new class of ML methods called deep learning (DL) can handle such high-dimensional, noisy and natural raw data by following representation learning or hierarchical data-driven approaches.

1.4. What is Deep learning ?

DL is a family of artificial neural network (ANN) based ML methods which have been inspired by the working principles of a human brain. In a DL network architecture, a series of hidden layers are connected in a cascade fashion between input and output of the network. Each of these layers takes input from its previous layer and transforms the data into a more abstract form. Non-linear layers allow DL methods to model complex relations between input and output of the network like shallow ANNs. DL is a representation learning method which means it can be fed with raw data and then it will automatically extract necessary representation for predictions. A DL network provides representations at different levels. The output of each of hidden layers is considered as the representation at that level. The higher layers the data belong, the higher-level abstraction we get these representations of the data. In different studies, these higher-level representations of raw data prove to be very effective for classification or detection problems. The most important thing here is that these representations, alternatively called feature vectors, are learned not by human engineering rather from the raw input data directly.

Unlike other ML methods, DL methods have been shown to efficiently handle high-dimensional and noise data in many domains, such as computer vision, language processing. These qualities of DL attract biomedical researchers to use DL instead of conventional ML methods because biomedical data (e.g. omics data) often suffer from high-dimensionality and noisiness.

1.5. How Deep learning evolved ?

With the improvement of GPU hardware and availability of massive training datasets, Krizhevsky et al. [12] have rekindled interest in deep learning models such as Convolutional Neural Networks (CNNs) by achieving a significant gain using CNN over existing methods in image classification on the ImageNet challenge. In computer vision, there has always been a growing need to train visual recognition systems more generically so that a system trained on one visual recognition task (e.g. classification) could be easily adapted to another task (e.g. detection). To handle such a challenge of adapting the source domain to different target domains, many domain adaptation methods have been proposed [13,14,15]. For example, Donahue et al. [16] extracted “deep features” from a deep neural network trained on one computer vision task and shown the state-of-the-art performance on a variety of other tasks. Razavian et al. [17] also adapted deep CNN (DCNN) features to build a pre-trained CNN called OverFeat and achieved remarkable performance gain simply by applying the model to a variety of visual recognition tasks that OverFeat was not trained for [18]. After having feature descriptors extracted from the first fully connected layer of OverFeat, they applied a linear SVM classifier to these features for image classification, scene recognition, fine-grained recognition and attribute detection on different datasets. A pre-trained CNN is usually followed by domain-specific fine-tuning on data from the target domains, especially when training data is scarce. Following this approach, Girshick et al. [19] fine-tuned a CNN pre-trained on ILSVRC2012 classification dataset and achieved substantially better object detection performance on PASCAL VOC as compared to the standard models based on simple hand-engineered features.

Hinton et al. [20] introduced a technique called Dropout as a form of regularization by selecting a random set of activations during training in order to set their weights as zero within each layer. The output is an averaged result of predictions of several other grouped models. Wan et al. [21] proposed DropConnect to generalize the Dropout model and it achieved state-of-the-art performance on some benchmark datasets comparing to Dropout by training a large model without any overfitting problem.

Another DL architecture is deep belief network (DBN). This is a probabilistic model and generates random observed data values with hidden parameters. DBNs can be trained in a layer-by-layer approach in which these layers are made of restricted Boltzmann machines (RBMs). Hinton [22] proposed an approach called contrastive divergence to learn the weights of RBM using maximum likelihood method.

A DNN can be pre-trained using a DBN. A DBN network is first trained and the learned weights from this pre-trained DBN are then used to initialize the weights of the DNN. This is useful when the training data is small because the random initialization of weights can significantly hamper the performance of the learned model. Since the learned DBN weights are already close to the optimal value of the best performing model. This approach can not only improve the performance of the model but also minimize the duration of fine-tuning [23]. Stacked autoencoder is another variant of DL-based approach to produce a good representation of input data. This network can capture the ordered grouping of the input in an unsupervised fashion. Vincent et al. [24] proposed this idea to produce a robust representation of the corrupted input data to recover the corresponding input data. This also refers to feature extraction for the representation of the input data. A DNN can be built to stack an autoencoder on the top of

another. They demonstrated that this approach can improve classification performance in many applications.

DL based methods have already achieved state-of-the-art prediction performances in diverse fields such as image classification [25], object detection [12], speech recognition [27] etc. However, DL methods also allow us to build state-of-the-art prediction models for sequential data [26, 28].

1.6. Application of deep learning to solving different bioinformatics applications

Analyzing gene expression data is very important in discovering tumor-specific biomarkers and clinical diagnosis [29], but high-dimensionality and the noisiness in the gene expression data pose a great challenge to biologists for cancer detection using traditional machine learning methods. Dananee et al. [30] proposed a deep learning based approach which implements a Stacked Denoising Autoencoder (SADE) [31] to analyze high dimensional gene expression data. This SADE network condenses the high-dimensional gene expression data into a lower dimension and produces a new eloquent illustration of its input. Connectivity matrices of this SADE allow them to identify a set of gene regulatory targets. These targets should be studied further as they have the potential to be very useful in cancer diagnosis. Somatic point mutation based cancer classification (SMCC) is very important to know the patient-specific cancer conditions so that specified personal therapy can be provided. SMCC becomes attractive research problem as DNA sequencing technology allows to have a huge volume of sequencing data. However, existing SMCC methods do not generate satisfactory cancer type or subtype classification result because of high data sparsity and small sample size. Yuan et al. [32]

proposed a new DNN-based model called DeepGene to eliminate these issues in SMCC. This model first filters the gene data by mutation rate to remove irrelevant genes from them. Then it indexes the gene data by their non-zero elements which let DeepGene overcome the data sparsity problem. Finally, the outputs of these two steps are fed into a DNN which performs automatic extraction of features for SMCC. DeepGene achieves ~67% prediction accuracy which is much better than the prediction performances of most baseline classifiers (i.e. SVM ~67% , k-Nearest Neighbors (KNN) ~42% and Naïve Bayes (NB) ~9%) . Liang et al. [33] also proposed a model which uses DBN for the purpose of clustering cancer patients by integrating multimodal data. They integrate gene expression data and clinical data (e.g. survival time) and feed the output into the DBN model. This model can capture intra- and cross-modality correlations (i.e. correlation among genomic data from different platforms) and learn a unified representation of the input. As a result, this model outperforms existing methods in clustering cancer patients. In addition, this model can predict missing values in the data and identify key target genes of miRNAs responsible for different cancer subtypes. Furthermore, preliminary clinical screening of a patient with skin disease usually begins with a visual diagnosis by a dermatologist. Since this is a very common malignancy in a human being [34, 35], an automatic system to classify skin diseases will be very helpful for the clinical purpose. Esteva [36] collected 129,450 clinical images of skin diseases and built a DCNN model to classify them. This model achieves better prediction performance than the real-life dermatologists. Furthermore, it can be deployed on a mobile device because of its scalability and fast performance speed. Nonetheless, a large number of nuclei and the variability in their sizes in histopathological images of breast cancer pose a great difficulty to build an automated system for nucleus detection. Xu et al. [37] overcame this

challenge by using a deep learning approach called Stacked Sparse Autoencoder (SSAE). This model outperforms nine previous state-of-the-art nuclear detection methods.

Conventional machine learning approaches have been applied to analyze high-content microscopy data to protein subcellular localization from yeast cell images [38]. However, these approaches were not able to perform such analysis without human expert's intervention and yet did not provide accurate classification. Kraus et al. [39] came up with a model called DeepLoc which is a DCNN based approach to overcome these limitations. DeepLoc outperforms the model ensLOC [38] by 71.4% according to mean average precision using fewer number of images. However, ensLOC uses binary SVM ensemble approach to assign single cells to subcellular compartment classes. Kraus et al. [39] also investigated the reason behind their success over ensLOC by performing 2D visualization of their network's components. They found out that DeepLoc generates a unique signal for different inputs. The structure of a protein and its functions can be studied further by protein contact map prediction from sequences. Wang et al. [40] treated this problem as a pixel-level labeling by considering a protein contact as an image. They proposed a novel deep learning based protein contact map prediction model with extremely unbalanced positive and negative labels. Their model integrates two evolutionary couplings (EC) and sequence conservation information into their network. Their model gives the state-of-the-art performance result in protein contact map prediction. Furthermore, the predicted proteins contacts by this model can generate an improved 3D structure model than previous best models: CCMpred [41] and MetaPSICOV [42]. Besides, many biological processes such as signal transduction and cellular organization can be affected by different protein-protein interactions (PPI). Hence, it is very important to build a PPI prediction model in order to provide a better design for the therapy of a disease. Sun et al. [43] are the first one to build a deep

learning based model that is a stacked autoencoder for the sequence-based PPI prediction. They achieved an accuracy of 97.19% with 10-fold cross-validation which is better than any existing PPI predictors.

Genomics becomes rich with many different types of functional genomic data because of latest sequencing technology. Eser et al. [44] proposed a new integrative framework called FIDDLE which integrates multiple types of genomic data to predict yeast Transcription Start Site sequencing (TSS-seq) [45]. FIDDLE confirms that TSS-seq data can be predicted using only one dataset as well as by integrating multiple datasets (e.g. RNA-seq, DNA sequence) as input. However, FIDDLE gives improved prediction performances when its input is the integration of multiple datasets (i.e. RNA-seq and DNA sequence) instead of only one dataset (i.e. RNA-seq or DNA sequence).

Chen et al. [46] proposed a deep learning system (D-GEX) which takes a gene's expression profile as input and infers the expression profile of a target gene. D-GEX has the ability to show cross-platform generalization. This model archives 15.33% improvement in gene expression prediction than a linear regression approach. D-GEX proves its cross-platform generalization when the learned D-GEX is used in RNA-Seq-based database for gene expression prediction for each target gene and still outperforms LR by 6.57%.

Existing methods for classification of cellular phenotypes from cellular images consist of multiple steps. Each of these steps is required with manual modifications and the tuning of different parameter settings. Godinez et al. [47] introduced a new multi-scale CNN (M-CNN) network which uses microscopic images to classify them into phenotypes. The prediction performances of the M-CNN in terms of accuracy over eight benchmark datasets are significantly higher than the previous state-of-the-art methods including CNN based approaches.

Gene expression can be regulated using transcription factors (TFs). So, the cell-specific TF binding predictions using gold standard Chip-seq data is very important. Qin and Feng [48] introduced a DNN model termed TFImpute to achieve the above goal. TFImpute can determine whether a specific TF would bind to a given DNA sequence in a specific cell line. The prediction performance of TFImpute proves its superiority from the comparison with another latest DNN-based approach called DeepBind [49]. Therefore, biologists can use TFImpute to understand how TF binding can be included by a specific cell line.

Zhou et al. [50] are the first to propose a DCNN based approach to predict the effects of noncoding-variants from large-scale chromatin-profiling data and achieved state-of-the-art predictive performance. They call their method as deep learning–based sequence analyzer (DeepSea). Experimental results show that DeepSea can also precisely predict the consequence of specific SNPs on TF binding.

Obtaining precise knowledge about a patient's health condition is crucial to provide early and better treatment. Discovery of good imaging biomarkers can lead clinical research into achieving this goal. Oakden-Rayner et al. [51] provided proof-of-concept research which proves that computer-based cross-sectional chest CT image analysis is able to predict 5-year mortality in adult (age >60 years) person. Their framework includes deep learning model and the predictive performances of this model are better than those who use human-generated features. Besides, visualization of different components of this deep learning based model can provide an explanation about the better prediction performances [52].

Gene expression can be controlled by enhancer elements and cis-acting DNA regulatory elements [53]. However, existing enhancer predictors face a challenge, that is, the lack of availability of huge and experimentally confirmed enhancers for humans or other species. Yang

et al. [54] developed a DNN-based hybrid architecture termed as BiRen which takes only DNA sequence as input to predict enhancers. Experimental results proved that BiRen can predict common enhances more accurately than previous state-of-the-art methods, which are based on DNA sequence only.

Analysis of high-dimensional single-cell RNA-seq data is very important to answer several biological questions such as the amount of heterogeneity of cells in a population, the discovery of a biomarker for explicit cells and retrieving analogous cell types. Lin et al. [55] introduced an NN based model to address all these queries without integrating any prior knowledge into the model. This method can deduce cell type more properly using a database of tens of thousands of single cell profiles than any existing methods.

Although the significant advancements have been made in applying DNN models to different bioinformatics applications as described above, no studies have been performed to use DNN models built for molecular subtypes of breast cancer classification either by CNA profiles or gene expression profiles or by integrating both. Furthermore, there is no DNN-based regression model to predict the triglyceride concentrations in the human blood using epigenome-wide DNAm data. In this thesis, we have proposed several DNN-based classification frameworks which take either CNA profiles or gene expression profiles or both of them as input for the prediction of molecular subtypes of breast cancer. In addition, we also proposed a DNN-based regression model which takes high-dimensional DNAm data as input to predict triglyceride concentrations (before and after treatment) in the human blood.

Chapter 2

Motivation and Research Objectives

2.1. Motivation

Omics datasets need to be efficiently analyzed for providing useful insight about phenotypic traits. Such kind of insights can be further used for patient stratification. This may lead to identify right therapies to provide patient-specific treatment. However, as we know that omics data are quite high-dimensional and there exists a high correlation among the different elements in a data set e.g. genes in CNA or gene expression profiles. These characteristics make us difficult in building models to handle the data using conventional machine learning methods, since these methods often suffer from overfitting problem when such high-dimensional and correlated data goes as an input to the models directly. To overcome these limitations, this proposed thesis aims to develop DNN-based predictive models to handle high-dimensional and correlated omics data to predict complex phenotypic traits.

2.2. Hypothesis

We hypothesize that high-dimensional and highly correlated omics data can be efficiently modeled through multi-layer deep neural network. The phenotypic traits can be more accurately predicted using the proposed models than traditional machine learning models. Furthermore, the proposed DNN-based frameworks can be efficiently used to integrate multiple omics data sources to predict phenotypic traits.

2.3. Research Objectives

The research objectives of this thesis are to use genomic data to predict molecular subtypes of breast cancer and to predict triglyceride concentrations measured at different visits of peripheral blood samples. The molecular subtypes of breast cancer we aim to predict include the status of estrogen-receptor (ER positive and ER negative) and the PAM50 subtypes (Luminal A, Luminal B, HER-2 enriched and Basal-like). We have three specific aims:

A: Develop DCNN models to predict the molecular subtypes of breast cancer using gene expression and CNA data, respectively;

B: Develop novel DCNN models to integrate multiple genomic data to predict the molecular subtypes of breast cancer;

C: Develop DNN regression models for the prediction of triglyceride concentrations from multiple peripheral bloods draws using epigenome-wide DNAm profiles.

Chapter 3

Classifying molecular subtypes of breast cancer using single data source

3.1. Introduction

Rather than being a single disease, breast cancer is a collection of diseases with multiple subtypes. Breast cancer can be classified into estrogen-receptor-positive (ER+) and estrogen-receptor-negative (ER-). A patient has ER+ breast cancer if her cancer cells have receptors for the hormone estrogen. Classifying patients into hormone receptor positive or negative is important for physicians because they need to determine whether the patients need hormonal treatments or chemotherapy. With the advent of technologies researchers were able to use gene expression profiles to identify four intrinsic molecular subtypes of breast cancer (i.e., PAM50 subtypes): Luminal A, Luminal B, HER-2 enriched and Basal-like [9].

CNAs represent the somatic changes of copy numbers in a DNA sequence. According to Beroukhim et al. [56], CNAs are predominant in a different type of cancers. It is expected that this data type can also be used to predict different molecular subtypes (such as ER status and PAM50 subtypes) of breast cancer using patient-specific CNA profiles. Previously, machine learning models were built to predict these subtypes [57]. CNA profile data is a high-throughput data and traditional machine learning methods, such as SVMs and RFs, can be easily overfitted if such high-throughput data is used directly as an input into these learners.

Deep convolutional neural network (DCNN) based models do not use any hand-crafted features, rather they use the raw information about training samples and produce a complex form

of generic features to represent the input data. Unlike SVMs and RFs, these deep models are able to take an input vector of any length. To avoid the overfitting problem, deep learning provides a useful technique known as dropout [58]. Deep learning has achieved many state-of-the-art results in different computer vision fields such image classification [25]. Currently, deep learning methods are used to solve different problems in bioinformatics. For example, Denas et al. proposed a DCNN model for binding site prediction [59].

In this experiment, we propose to build a DCNN based model using CNA profile-based data to predict molecular subtypes of breast cancer: the status of estrogen-receptor (ER+ and ER-) and the PAM50 subtypes (Luminal A, Luminal B, HER-2 enriched and Basal-like). The former is a standard supervised binary classification problem while the latter is a supervised multi-class classification problem.

3.2 Deep learning model for the prediction of molecular subtypes of breast cancer

Specifically, we propose to use a deep convolutional neural network (DCNN) for the prediction of molecular subtypes of breast cancer (**Figure 1**). Our network receives a single vector (X) as an input to the input layer of the DCNN, which is followed by convolutional layers. Each neuron of a convolutional layer receives some input and performs a dot product operation. These convolutional layers are considered a strong pattern detector of local features.

The two convolutional layers (**Figure 1**) are followed by a one-dimensional pooling layer. The outputs of the convolution layers are considered low-level features. Pooling over these features creates higher-level features that can help smooth the noisiness in training data. This pooling layer partitions the outputs from the convolutional layers into a set of sub-regions and

for each of those regions the maximum value is taken as an output. The pooling layer reduces the size of its input vector to decrease the large number of parameters to be estimated, which is also useful to avoid potential overfit-ting and to make model invariant to input features.

In our experiment, there is a complex non-linear relationship between the response variable (such as the prediction score assigned to a patient for a specific molecular subtype of breast cancer) and the predictors (such as the gene-specific CNA profiles). Therefore, we use the Relu (Rectified Linear Units) layer after the pooling layer to model this non-linear relationship. Relu performs a threshold operation as:

$$f(t) = \begin{cases} t, t \geq 0 \\ 0, t < 0 \end{cases} \quad (1)$$

Here, t represents the input to a neuron.

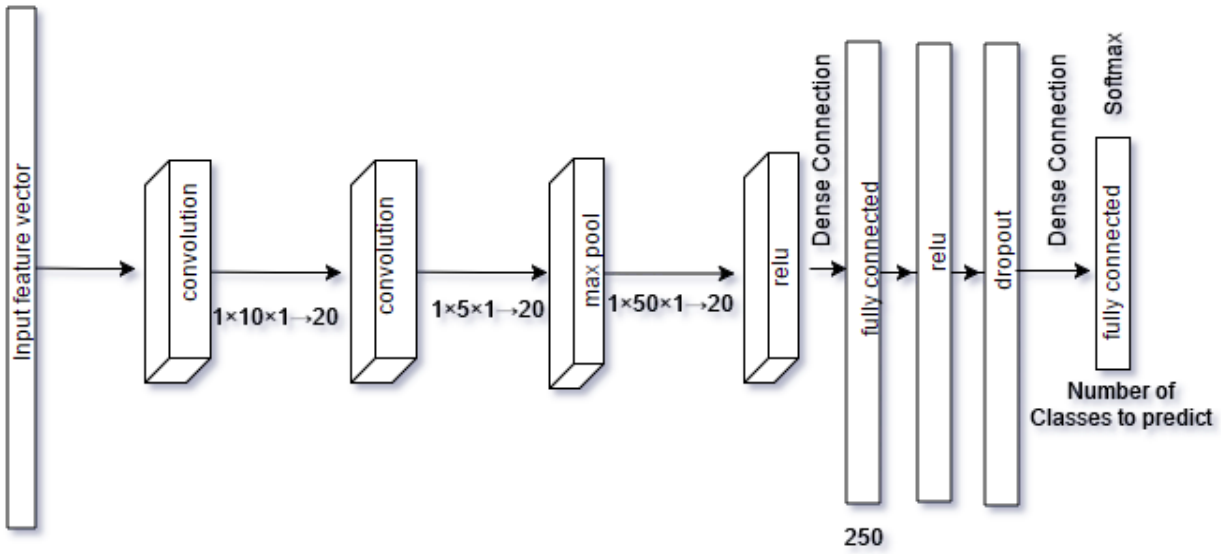


Figure 1 - Proposed architecture of DCNN. The number and size of different filters that must be learned is shown in the figure. Here, $1\times10\times1 \rightarrow 20$ means the kernel size is 1×10 and this is a one-dimensional feature while the total number of convolutional feature maps is 20. The stride

size for the convolutional layer is 1×1 and for the pooling layer is 1×2. The number of outputs of the first fully connected layer is 250. The size of the output of the last fully connected layer will be two for binary classification (ER status prediction) and four for multiclass classification (PAM50 subtype prediction).

To complete the higher-level reasoning of our network we use the fully connected layer ($F1$), as used in a traditional neural network, and the output of this layer can be calculated as a matrix multiplication tailed via a bias offset. So, we pass the output of $F1$ to another fully connected layer ($F2$) using a Relu layer and a dropout layer as medium. This helps our model overcome the potential overfitting problem and provides generalizability.

The output of $F2$ is a K-dimensional vector (a) that provides the prediction scores of test samples assigned to each of the classes. We use a softmax classification layer to transform the prediction scores into probability scores. This layer implements the softmax function using the prediction scores (a) and estimated parameters (e.g., w) from $F2$ to produce k-th probability scores for test samples assigned to each of the classes. Therefore, the probabilities that the test samples are assigned to the i-th class can be calculated as follows:

$$P(b = i|a) = \frac{e^{a^T w_i}}{\sum_{k=1}^{K} e^{a^T w_k}} \tag{2}$$

Here, $a^T w$ represents the inner product of a and w and K represents the number of classes. We train our network using the backpropagation approach and we use softmax loss to allow us to explain the prediction results as probabilities.

3.3 Experiments

3.3.1 Dataset

Our copy number alteration data is from the METABRIC (Molecular Taxonomy of Breast Cancer International Consortium) project [60]. For binary classification, we have 991 samples in the training set (794 samples and 197 samples for ER positive and ER negative classes respectively) and 984 samples in the test set. For the multiclass classification we have 935 samples for the training set (for Luminal A, Luminal B, HER-2 enriched and Basal-like classes we have 464, 268, 87 and 116 samples, respectively) and 842 samples for the test set. We have three discrete copy number calls: −1= copy number loss, 0= diploid, 1= copy number gain in our CNA mutation matrix (patients-by-genes).

3.3.2. Informative feature selection for DCNN

In total, we retrieved 18,305 genes. Since different genes have different numbers of CNAs across all patients, the genes are more informative if they have more somatic CNAs. We calculated the CNA frequency for each of the 18,305 genes as follows:

$$f_{CNA} = \frac{N_{CNA}}{M} \tag{3}$$

Here, f_{CNA} means CNA frequency of a gene, N_{CNA} means the number of copy number gains and losses of a gene and M represents the total number of samples (i.e., patients). We selected few cutoffs (0.0101, 0.0492, 0.0685, 0.1102 and 0.1283) based on the five-number summary statistics (minimum, first quartile, median, third quartile and maximum) and mean of the CNA frequency.

3.3.3. Construction of DCNN model

To implement the DCNN model (Fig. 1) for both the binary and multiclass classification tasks, we used publicly available C++ based deep learning library called CAFFE [61]. For each of the tasks, we trained several DCNN models using different CNA frequency cutoffs based on **Equation 3**: 0.0101, 0.0492, 0.0685, 0.1102 and 0.1283. The number of genes or features selected by the unsupervised approach at these cutoffs is 18305, 13476, 8857, 5192 and 4377, respectively. We used learning rate 0.001, batch size 64 and dropout ratio 0.5 to train our network.

3.3.4. Performance evaluation metrics and baseline models

We use overall accuracy and Receiver Operating Characteristics (ROC) curves to evaluate the performance of our DCNN classifiers. We use the area under the ROC curve (AUC) as the quantitative measure of the ROC curve. To compare the performance of our DCNN models, we use two state-of-the-art supervised classification models: SVM and RF, as our baseline models. We use two R packages known as e1071[62] and randomForest[63] to build SVM and RF models, respectively. For each sample, we have more than 18,000 genes while we have only ~1000 samples. SVM and RF are not able to use such high-throughput data as input vectors. Using such input will result in these models being overfitted. So, we performed nonparametric supervised Chi-square ($\chi 2$) test based on the number of samples in each CNV category: copy number loss, diploid and copy number gain and in each of the 4 breast cancer subtypes to calculate the significance of each of the genes. Then we selected the top (most significant) hundreds of genes to build our baseline models.

3.4 Results

Prediction accuracies and AUCs based on our DCNN models using different numbers of features selected at different CNA frequencies are shown in **Figure 2** and **Figure 3**.

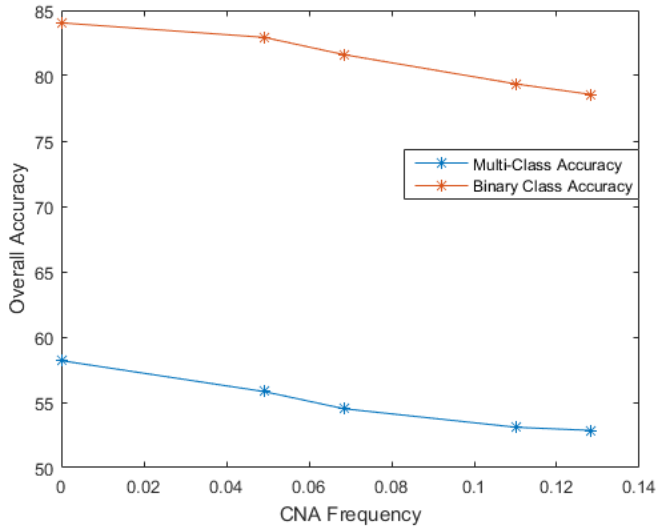


Figure 2 - Overall accuracy (%) of the proposed DCNN model at different CNA frequencies

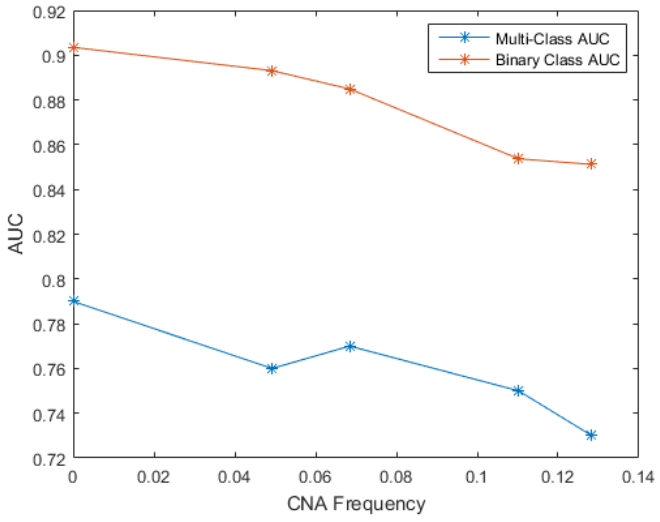


Figure 3 - AUC of the proposed DCNN model at different CNA frequencies

In general, the somatic CNA-based profiles have much larger power to predict ER status (binary classification) than PAM50 subtypes (multiclass classification). The models have highest prediction performance when all the features are used. Their performance decreases when the number of features used in the model's decreases.

The prediction results of our baseline models are shown in **Table 1** (accuracies) and **Table 2** (AUCs) for binary classification (B_SVM, B_RF) and multiclass classification (M_SVM, M_RF). The numbers in bold color mean that it is the best result among different numbers of the selected top genes. We use an R function called multiclass.roc [64] to generate multiple ROC curves to compute the multiclass AUC.

Table 1 - Overall accuracy (%)

Classifiers	Top selected genes				
	100	250	350	400	500
B_SVM	**76.5**	76.4	**76.5**	75.8	76.0
B_RF	81.5	**82.7**	**82.7**	82.3	81.7
M_SVM	42.7	43.7	43.8	**45.0**	43.7
M_RF	44.8	47.0	48.6	**49.5**	48.6

Table 2 - Area Under the Curve (AUC)

Classifiers	Top selected genes				
	100	250	350	400	500
B_SVM	0.693	0.686	**0.702**	**0.702**	0.693
B_RF	0.764	0.798	0.804	0.815	**0.817**
M_SVM	**0.780**	0.703	0.702	0.708	0.707
M_RF	**0.729**	0.725	0.723	0.715	0.725

Comparisons of the results (**Table 3** and **Table 4**) of our proposed DCNN models with our baseline models clearly confirms that our DCNN models outperform the results of SVM and RF. **Table 3** and **Table 4** show the best result among the binary and multiclass classifiers from **Figure 2**, **Figure 3**, **Table 1** and **Table 2**.

Table 3 - Comparison of the results for binary classification.

Classifier	Accuracy	AUC
DCNN	**84.1**	**0.904**
SVM	76.5	0.702
RF	82.7	0.817

Table 4 - Comparison of the results for multiclass classification.

Classifier	Accuracy	AUC
DCNN	**58.19**	**0.790**
SVM	45.0	0.780
RF	49.5	0.729

3.5 Conclusion and discussion

In this experiment, we showed that the proposed DCNN models achieve much better results than SVMs and RFs for both binary and multiclass classification tasks. We also demonstrated that the DCNN models can work well for data sets with larger numbers of features than samples, which often results in overfitting in SVM- or RF-based models. Although there are great advances using traditional machine learning models in different bioinformatics

applications, recent research including this paper shows that deep convolutional neural networks have significant advantages over them.

We use DCNN model rather than deep belief network (DBN) because DCNN models are more invariant to the translation of the data. DCNN can also provide a model which is more robust to the unwanted noisiness in the data than DBN. In our future work, we will incorporate DBN network into our experiments for both binary and multiclass classification.

Chapter 4

Classifying molecular subtypes of breast cancer by integration of multiple heterogeneous data sources

4.1. Background

Cancer progression is impelled by the accumulation of somatic genetic mutations, which consist of single nucleotide substitutions, translocations and copy number alterations (CNA) [65]. CNAs are somatic changes in the copy numbers of a DNA sequence that arise during the process of cancer development. This results in changes to the chromosome structure in the form of gain or loss in copies of DNA segments. This has been found to be prevalent in many types of cancer [56]. Genes in the CNA regions, if mutated, can create abnormal proteins with different functions than a normal protein, which can lead to the uncontrollable growth of cancer cells. Therefore, it will be useful to explore the possibility to predict the molecular subtypes of breast cancer by integrating both patient-specific CNA profiles and gene expression profiles.

Generally speaking, both of the CNA profile- and the gene expression profile-based feature vector for supervised machine learning algorithms includes the majority of the genes in the human genome; that is, each sample is represented by almost twenty thousand of genes. Supervised machine learning methods, such as support vector machine (SVM) and random forest (RF), work well to draw a decision boundary between two classes or the decision boundaries among multiple classes, but this becomes hard when the size of the feature vector is much larger than the number of training samples in many bioinformatics applications. Yeung and Ruzzo [66] used a classical technique known as Principle Component Analysis (PCA) for dimension

reduction. However, PCA linearly reduces the dimension of the data and fails to capture the non-linear relationship of the data. Recently, deep learning (DL) based models demonstrate advantages to handle high-dimensional data and extract linear and non-linear relationships of the data.

DNN models have also been applied for different bioinformatics domains. Denas and Taylor [59] preprocessed their genomic data as a two-dimensional matrix, where rows are the transcription factor activity profiles of genes and columns are positions of different genome elements. They applied a DCNN model to predict DNA-binding site. Kelley et al. [67] introduced a DCNN model to learn the functional activity of DNA sequences for 164 cell-specific DNA accessibility multitask prediction and this model achieved the best result than earlier methods. Zeng and Gifford [68] introduced a DNN to predict the DNA methylation level of a single CpG from the corresponding sequence, which showed improved performance than all previous models. Leung et al. [69] used mouse RNA-Seq data to build a DNN-based model to predict splicing patterns in individual tissues and achieved the best result among the other available methods such as Bayesian methods. However, there are no DNN models built for classifying molecular subtypes of breast cancer by integrating both CNA profiles and gene expression profiles.

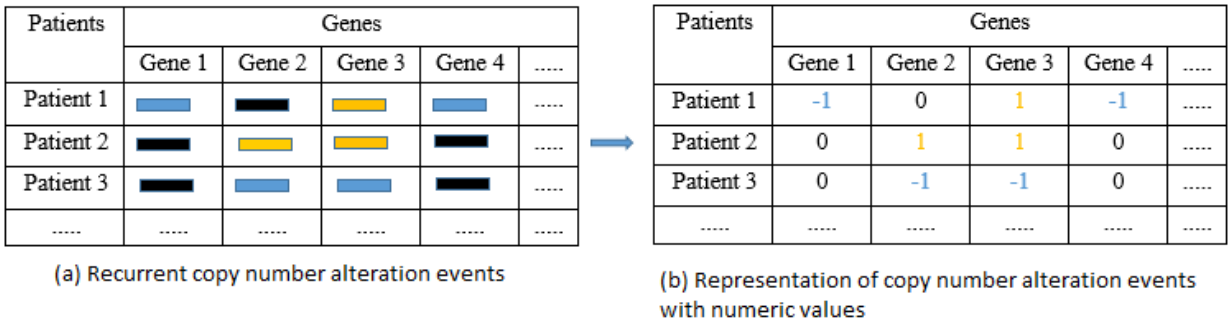
In this experiment, we propose to build our CNA profile- and gene expression profile-based classification models of molecular subtypes of breast cancer using an integrative deep neural network learning approach. The molecular subtypes of breast cancer we aim to predict include the status of estrogen-receptor (ER+ and ER-), which is a binary classification problem, and the status of PAM50 subtypes (luminal A, luminal B, HER-2 enriched and basal-like), which is a multiclass classification problem.

4.2. Materials and methods

4.2.1. Datasets

We use copy number alteration data and gene expression data from METABRIC (Molecular Taxonomy of Breast Cancer International Consortium) project [60]. The group collected around 2,000 clinically annotated primary fresh frozen breast cancer specimens along with a portion of normal specimens from different North American and European tumor banks. The primary tumors could be categorically linked to DNA and RNA specimens. The authors performed quality control assessment and excluded the mismatches between DNA and RNA. After that, paired DNA and RNA profiles were available from tumors that were taken from 991 female patients. They also collected DNA from adjacent normal breast tissue from 485 samples in the discovery set. The second group of 984 cases was collected later which included low cellularity tumors, DCIS (Ductal carcinoma in situ), and three benign cases. This group represents the validation set and was used to test the reproducibility of the integrative cluster and clinical outcome associations.

To determine copy number alteration events in each breast cancer patient, we focus on gene-specific CNA events as shown in **Figure 4**. We use the set of discrete copy number calls −1= copy number loss, 0= diploid, 1= copy number gain. For each CNA region in each patient, we retrieve its gene information based on its chromosome positions using the biomaRt R package [70].



Patients	Genes				
	Gene 1	Gene 2	Gene 3	Gene 4	.....
Patient 1					.....
Patient 2					.....
Patient 3					.....
.....	.....	.....	.....	.....	.....

(a) Recurrent copy number alteration events

Patients	Genes				
	Gene 1	Gene 2	Gene 3	Gene 4	.....
Patient 1	-1	0	1	-1	.....
Patient 2	0	1	1	0	.....
Patient 3	0	-1	-1	0	.....
.....	.....	.....	.....	.....	.....

(b) Representation of copy number alteration events with numeric values

Figure 4 – Representation of copy number alteration events. Patient-level individual copy number alterations are matched to gene regions in the human genome (hg19). (a) Recurrent copy number alteration events. The blue segments are copy number loss, the black segments are copy number diploid and the orange segments are copy number gain. (b) Representation of copy number alteration events with numeric values. "-1" represents copy number loss, "0" represents copy number diploid and "1" represents copy number gain.

Gene expression data were generated from Illumina BeadArrays (i.e. Illumina HT-12 v3 platform). The data were preprocessed (including quantile normalization) using the beadarray R package by Curtis et al. [26]. For our experiment, we focus on the gene expression profiles of the 16,289 genes common in both CNA and gene expression data sets.

For the binary class classification, we take 991 patient samples from the discovery set as our training set and 984 patient samples from the validation set as our test set. In this training set, we have 794 samples for the ER+ class and 197 sample for the ER- class. However, for the multiclass classification, we take 935 patient samples from the discovery set as our training set and 842 patient samples from the validation set as our test set since some of the patients in the whole discovery and validation sets are in the normal group. In this training set, we have 464, 268, 87 and 116 samples for Luminal A, Luminal B, HER-2 enriched and Basal-like classes

respectively. The labels of the molecular subtypes of these patients are extracted from the Supplementary Tables 2 and 3 of [60].

4.2.2. Deep Neural Network Architectures

4.2.2.1. Base network architecture

The network architecture of our base DCNN model to predict the molecular subtypes of breast cancer using individual datasets is shown in **Figure 5**.

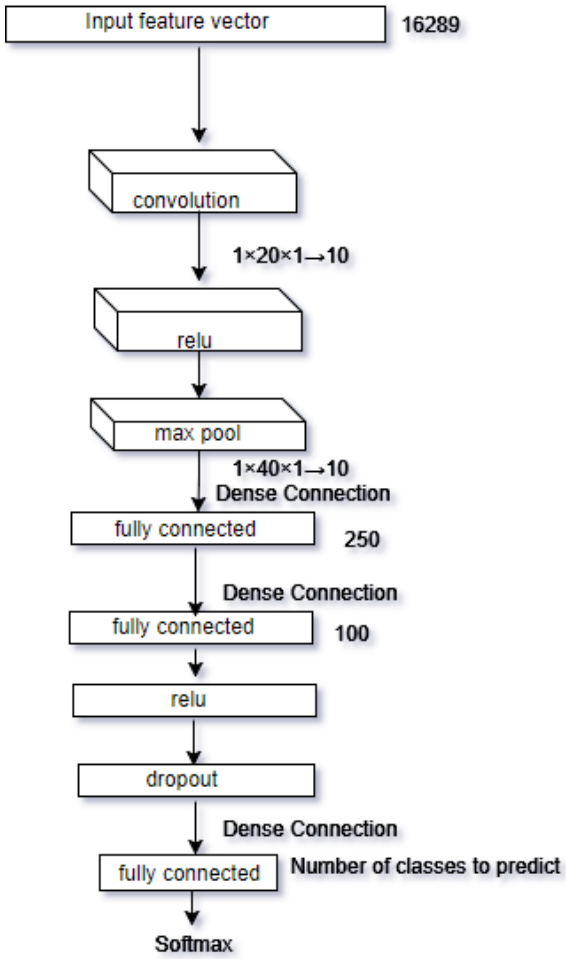


Figure 5 – Individual data source-based DCNN architecture. A backpropagation approach is used to train the multi-layer network. The size of input feature vector, the size of the resulted vectors from fully connected layers and the size of different kernels at different layers are listed. Here, $1\times20\times1 \rightarrow 10$ represents a kernel of size 1×20 and the height of all 10 feature maps is 1. We use a stride of size 1×5 for convolutional layers and 1×10 for the max pooling layer.

This network takes the single data source (such as CNA or gene expression) of a sample as an input feature vector (X) which goes directly to a convolutional layer. A filter F (also known as kernel), which is an array of numbers (also known as weights), slides over all the positions of X. The height of F and X must be the same and here it is 1 as we are dealing with one-dimensional input vector. The region R over which the F is currently moving is known as receptive field. An elementwise multiplication is performed between F and R, which produces a single number to represent R. This process continues until it covers every position of X and the resulted vector is termed as activation or feature map. So, if X is a Q-dimenstional vector, then the size of the activation map would be $1\times(Q-F)$. One can have any number of feature maps by using different Fs. For our base DCNN model we take 10 convolutional feature (CF) maps and the size of our input feature vector is 1×16289 and the size of the convolutional kernel is 1×20. These CFs represent the local patterns of our input feature vector X.

The output of our convolutional layer (CFs) goes to the ReLU (Rectified Linear Units) layer. ReLU is an activation function, which is useful to model the complex non-linear relationship between the input and output of the model. For our experiment, the input can be either gene-specific CNA profiles or gene expression profiles and the output is the prediction score for a patient assigned to one of the molecular subtypes. Unlike other activation functions

(e.g. tanh or sigmoid), ReLU implements a simple thresholding function rather than an expensive exponential function. ReLU function follows the **Equation 1.**

We know that a DCNN model with a large number of neurons can model any complex relationship between its input and output. However, here we have a small number of training samples for our DCNN model, which can be easily overfitted over the training data. Hence, the Relu layer is followed by a max pooling layer to reduce the size of the input feature vector, which is also known as down sampling. A filter goes over its input and takes the maximum value of the receptive field. Although pooling may cause loss of information, such kind of loss is useful because we will have fewer numbers of parameters to be learned which helps the model overcome the curse of overfitting problem. This layer also helps the model become invariant in terms of translation, rotation, and scaling of the input data. Therefore, the pooling layer leads the DCNN model to have better generalization over the test data.

The output of our pooling layers is then inputted to a fully connected (FC1) layer. This layer has a connection to its previous layer for each of the neurons and the output of this layer is a simple matrix multiplication which is a one-dimensional vector. For our experiment, the size of this vector is 1×250. This FC1 layer is then followed by another fully connected (FC2) layer to get higher level features of our input feature vector X. However, our network is training huge number of parameters using only a few hundreds of training samples. So, we pass this output to another fully connected layer (FC3) via a Relu layer and a Dropout layer. Dropout layer implements a regularization technique to prevent the DCNN model from overfitting. This layer randomly drops different units with its associated connections.

The output of FC3 is a vector of size 1×2 or 1×4, where 2 and 4 represent the number of classes of estrogen-receptor and PAM50 subtypes, respectively. FC3 takes the high-level

features of X from the output of FC2 and regulates each of the features mostly correlates with a specific class. Each of the values of FC3 represents a prediction score for a particular class, which is then converted into a probability score using a softmax classification layer. This layer implements the softmax function using two parameters from the output of FC3 : prediction scores (x) and weights (y), which are used to calculate the probability of the p-th class using the following formula:

$$P(z = p|x) = \frac{e^{x^T y_p}}{\sum_{k=1}^{4} e^{x^T y_k}} \quad (4)$$

Here, $x^T y$ represents an inner product between x and y.

Finally, we use network backpropagation to train our DCNN models.

4.2.2.2. DNN models for data integration

We propose three DNN-based data integration models. The first two models are based on deep convolutional neural networks as shown in **Figure 6** (DCNN_Concat) and **Figure 7** (DCNN_Siamese) and the third one does not involve any convolutional operation rather it is a fully-connected DNN as shown in **Figure 8** (DNN_SE). Below we briefly describe the integration techniques.

4.2.2.2.1. Concatenation

This is an intermediate integration technique. This method takes two feature vectors as input: one from the CNA data and another from the gene expression data. Both of these vectors represent information from the same patient and have the same label.

The outputs of fully connected layers from the left branch (FC_L) and right branch (FC_R) represent the DCNN feature vector of the inputs. To integrate the knowledge of the same patient from these two different sources we use concatenation layer. This takes the outputs of these two fully connected layers and performs a concatenation operation between them. We call this architecture as DCNN_Concat (**Figure 6**).

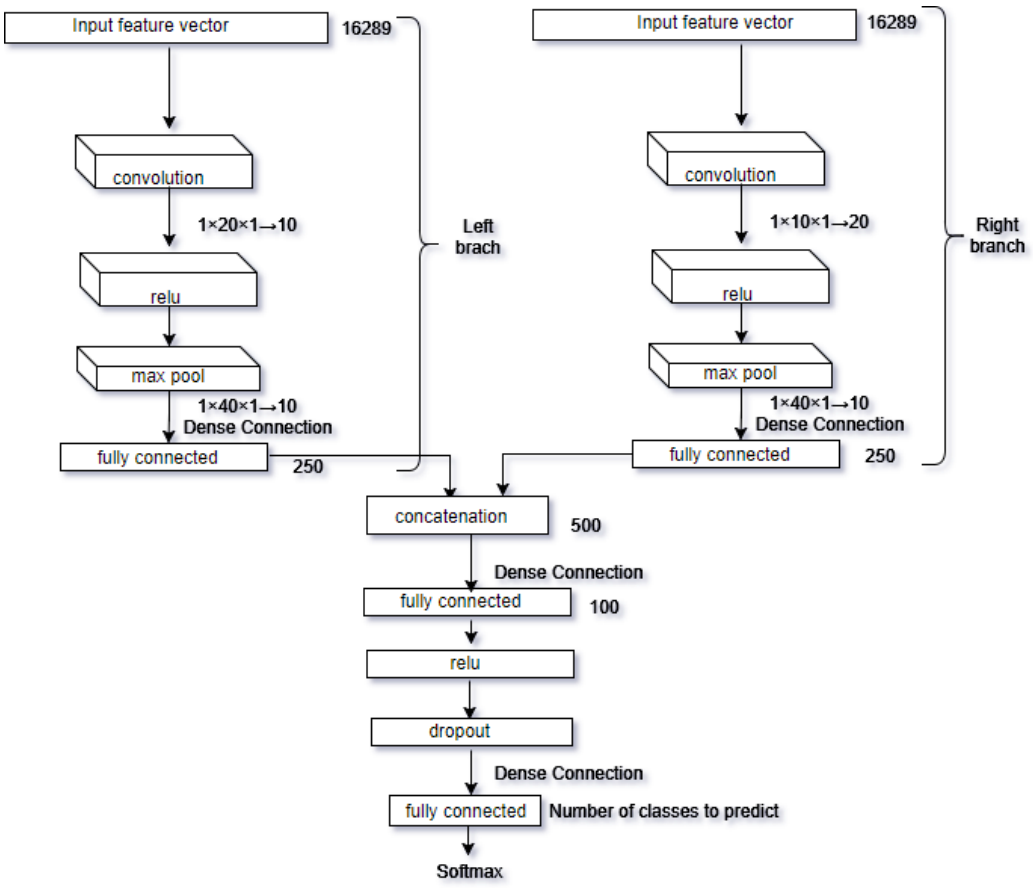


Figure 6 – Concatenation-based data integration for DCNN architecture. The DCNN model is first learned for CNA data (left branch) and gene expression data (right branch), respectively. The high-level features from the two data sources are then concatenated. The DCNN model is further learned based on the concatenated results to make a final prediction of PAM50 subtypes.

Suppose, C represents the CNA data of patient X and G represents the gene expression data of X and the label (tumor subtype) is the same for both C and G. Now, C goes as an input to the left branch and G to the right branch. Then both C and G go through different layers of left and right branches. So, the outputs of FC_L and FC_R layers, which are named as k_L and k_R, respectively, are considered as the higher-level representation of C and G. Both C and G are 250-dimensional vectors so the concatenation layer takes k_L and k_L as inputs and produces a 500-dimensional vector (V):

$$V = k\_L \parallel k\_R \tag{5}$$

Here, $\parallel$ represents the concatenation operation. Then V goes through other different layers of the DCNN to provide the final higher-level reasoning from the integrated data. The final output is the predicted probability for a particular class of molecular subtypes of breast cancer.

4.2.2.2.2. Weight sharing network

Similar to DCNN_Concat (**Figure 6**), the weight-sharing network also contains two different branches to take a patient's raw information in terms of CNA data and gene expression data respectively. However, the architecture of this approach involves sharing information (i.e. weight) between layers of the two branches for the two data sources (**Figure 7**).

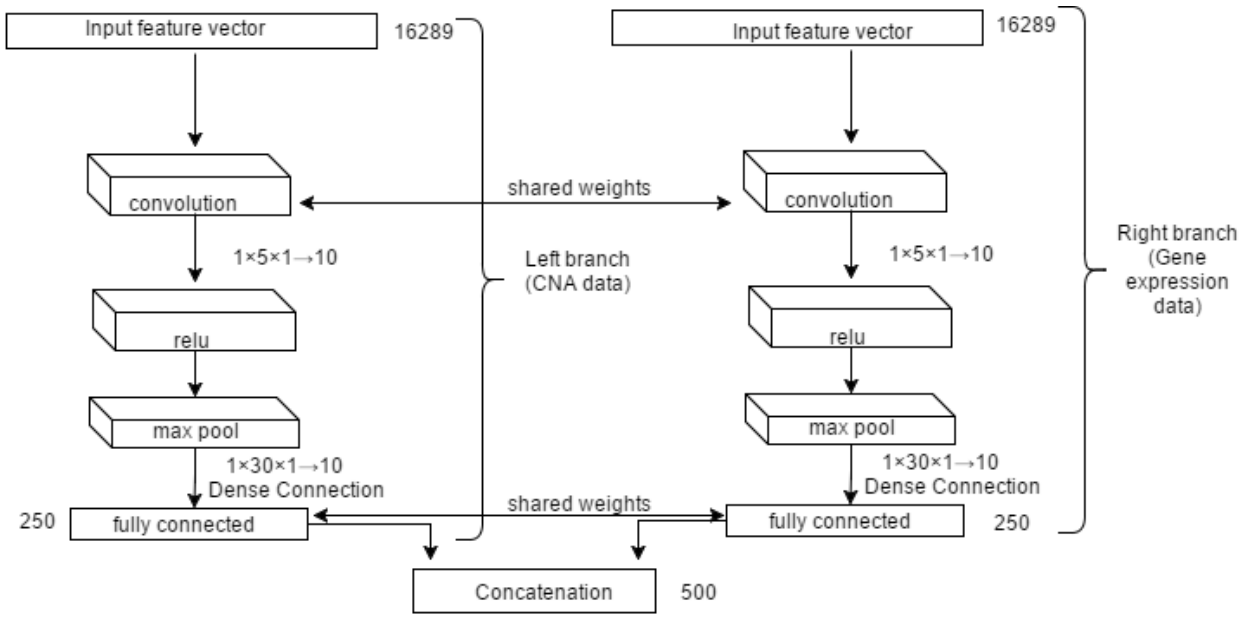


Figure 7 - Weight sharing-based data integration for DCNN architecture. Weight sharing network is similar to concatenation network except that the two branches for learning models from CNA and gene expression data will share the same weights or kernels. To integrate the high-level features from the two data sources, concatenation operation is used in this study, but other operations can be performed.

This type of network is termed as Siamese network. So, we call this architecture as DCNN_Siamese. This network takes two feature vectors for the two data sources as input: CNA data (left branch) and gene expression data (right branch) of the same patient with the same class label. Both convolutional layers of the two branches use the same sized kernel with the exact same weights. Fully connected layers of both branches also do the same. However, the Relu and max-pooling layers do not have any weight parameters to learn and they perform only mathematical operations so they are not involved in weight sharing. This means the model needs to learn fewer parameters which help the model not to be overfitted over training data. Like the architecture of DCNN_Concat (**Figure 6**) we merge the outputs from k_L and k_R by a

concatenation layer and then we pass the resultant vector to other different layers of the DCNN to provide the final higher-level reasoning from the integrated data to get the final prediction of a particular class label of molecular subtypes of breast cancer.

4.2.2.2.3. DNN integration model with weights initialized by stacked autoencoder

We first train a deep network in an unsupervised fashion and this creates a set of feature detector layers without using the labels of the samples. To do this we use a stacked autoencoder (SE) approach.

We concatenate CNA data and gene expression data for each of the samples, which results in a 32578-dimensional vector as an input to the SE network (**Figure 8(a)**).

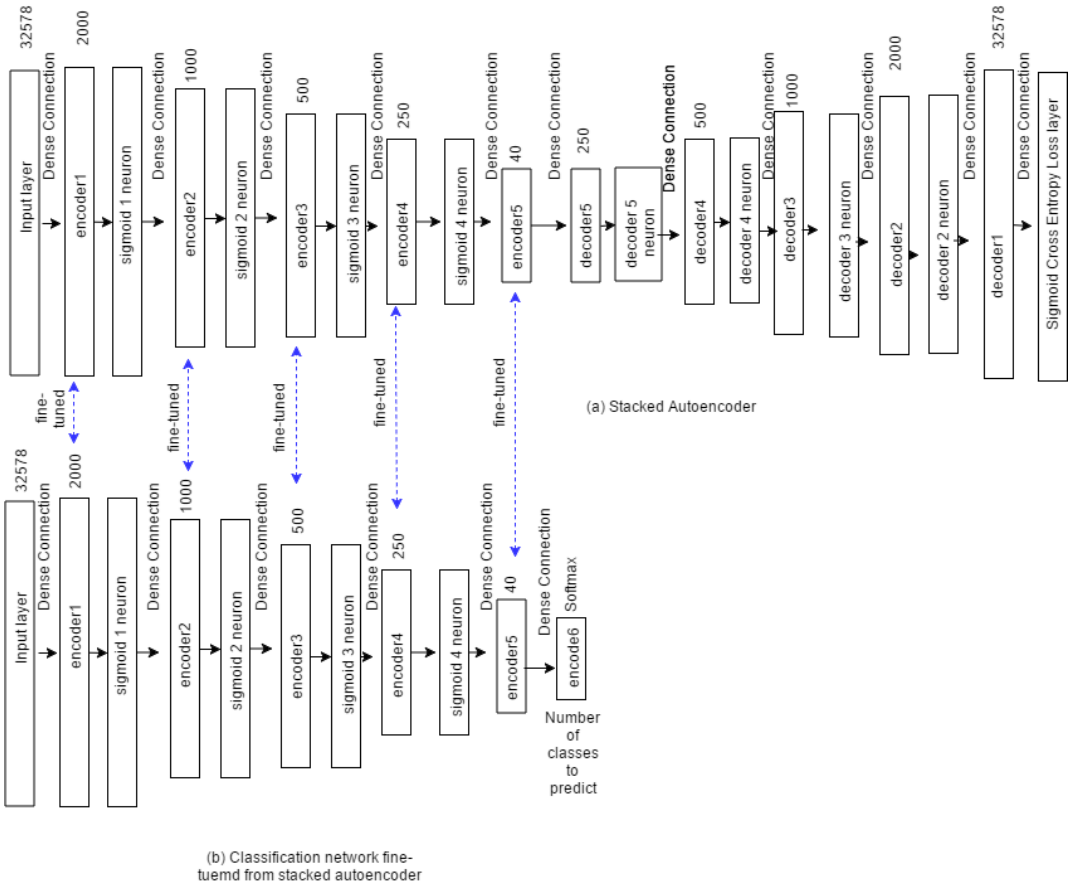


Figure 8 - Stacked autoencoder-based data integration for DCNN architecture. (a) Build stacked autoencoder from integrated data; (b) Build classification model fine-tuned from the pre-trained stacked autoencoder in (a).

In this architecture, we have two parts: encoders and decoders. Each of the encoder layers has a corresponding decoder layer. The purpose of learning this network is to reconstruct the raw inputs in the corresponding decoder layers. Each of the encoder and decoder layers is followed by a sigmoid neuron except the last decoder layer. We use sigmoid neuron layer so that small changes in one of the encoder or decoder layers do not make large changes to their outputs since such small changes can sometimes flip the output such as 0 to 1. The output of sigmoid function can be defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

Here, x represents an input to a neuron. Sigmoid function squashes the real numbers to range between 0 and 1. Therefore, the network becomes non-linear.

We use sigmoid cross entropy loss function (**Equation 7**) to train our SE network in a backpropagation style. This loss function takes the output of a fully connected layer as its input and it uses a sigmoid function to provide a gradient estimation.

$$Loss(Y, X) = -\sum_{i=1}^{n} X_i \log(Y_i) \tag{7}$$

Here, n is the total number of training inputs, X is the label, which is the input itself and Y is the prediction of the network.

After training the SE, we train another deep network (**Figure 8(b)**) which contains the same layers as the encoder layers of this SE and in addition we put another layer on the top to get

the final prediction of a particular class label of molecular subtypes of breast cancer. Here, the weights of all layers except the lastly added layer are fine-tuned from the encoder layers of the SE. In this way, the weights of the network are initialized with much more practical values which may lead to better training and classification results. We call this DNN network architecture as DNN_SE.

4.3. Data integration for SVM and RF classifications

For each specified number of top genes, we first select the genes based on a $\chi 2$ test for CNA data and ANOVA test for gene expression data, then we concatenate the selected CNA data and gene expression data, finally, we perform the classification analysis using SVM and RF for the selected gene sets.

4.4. Software and parameters

We build our DNN models using CAFFE [27], which is a C++ based deep learning library. We use all the 16,289 genes common to both data sources as input vectors to each of our DNN models (**Figures 5-8**). We train all our DNN models using learning rate 0.001. We run the models with several different sets of parameters and report the best results in this paper.

4.5. Model performance evaluation and baseline models

We use two methods to measure the performance of our DNN classifiers. The first one is overall accuracy, which is the proportion of patients with correctly predicted molecular subtypes.

The second one is Receiver Operating Characteristics (ROC) curve, which depicts the pattern of sensitivity (1-FNR) and specificity (1-FPR) of a classifier at several different discrimination thresholds, such as the probability assigning a given sample to a given molecular subtype of breast cancer. Here, FNR means false negative rate and FPR means false positive rate. The quantitative index used to evaluate a classifier based on ROC is the area under the ROC curve (AUC). We use an R function called multiclass.roc [64] to generate multiple ROC curves for computing the multiclass AUC.

The performance of our DNN models is compared to that of other two state-of-the-art supervised classification models: SVM and RF. We build these models using R packages e1071 [62] for SVM and randomForest [63] for RF. Since we have more than 16,000 genes or features and only around 1,000 samples, and it is well-known that SVM and RF will suffer from overfitting problem if such high-dimensional data is applied, we select top significant genes to build the baseline models. We calculate the significance of each of the genes using different supervised approaches. For CNA data, we use χ^2 test since it is category data while for gene expression data, we perform parametric ANOVA test. The selected top significant genes are used to build the baseline models.

We also compare our results with another DNN model whose weights are initialized by a pre-trained DBN (DNN_DBN). We use an R package called deepnet [71] to build the DNN_DBN model. At first, we take the input vectors from CNA data and gene expression data respectively for each of the samples. Then we perform a concatenation operation between them which produces a 32578-dimensional vector. We give this vector as an input to both DNN_DBN and DBN. Finally, we use the learned weights of DBN to initialize the DNN_DBN.

4.6. Results and discussion

Table 5 presents the accuracies and AUCs of our DCNN models for multiclass (PAM50 subtypes) classification. It can be seen that the models can predict different classes of PAM50 subtypes more accurately using gene expression data (classifier: Gene_DCNN) than CNA data (classifier: CNA_DCNN). The prediction performance of the model "Gene_DCNN" in terms of accuracy is 77.3% and AUC is 0.832, which is significantly better than the model "CNA_DCNN" of accuracy 50.5% and AUC 0.677. Among the DNN-based integration models, we get the best result when we integrate the two data sources using concatenation layers without sharing the weights (classifier: DCNN_Concat). Overall, the integrated DNN models (DCNN_Concat, DCNN_Siamese, and DNN_SE) show better performance than the DCNN models (CNA_DCNN and Gene_DCNN) trained on individual data sources.

Table 5 - The overall accuracies (%) and AUCs of our DNN models for multiclass classification. CNA_DCNN and Gene_DCNN are based on the architecture of **Figure 5** for CNA data and gene expression data, respectively. DCNN_Concat, DCNN_Siamese, and DNN_SE are DNN models based on the network architectures described in **Figure 6, 7** and **8**, respectively.

Classifier (all genes)	Datasets	Performance Measurement	
		Accuracy (%)	AUC
CNA_DCNN	CNA	50.5	0.677
Gene_DCNN	Gene expression	77.3	0.832
DCNN_Concat	CNA and gene expression	79.2	0.850
DCNN_Siamese	CNA and gene expression	76.7	0.838
DNN_SE	CNA and gene expression	77.3	0.838

DCNN_Concat shows better performance over DCNN_Siamese because the layers in the two branches for the two data sources in DCNN_Concat model (**Figure 6**) learn different weights but those in DCNN_Siamese model (**Figure 7**) learn the same weights. This gives us the insight that CNA data and gene expression data need to be treated differently. Besides, DCNN_Concat captures the correlation among the genes using convolutional layers but DNN_SE model (**Figure 8**) does not consider this correlation. This may cause the lower performance using DNN_SE over DCNN_Concat.

Performance of our baseline models (SVM and RF only) are shown in **Table 6** (accuracies) and **Table 7** (AUCs). There are no significant changes in the results of using a different number of top selected genes for both SVM and RF models. Similar to our proposed DNN models, SVM and RF also provide better prediction results using gene expression data than CNA data. This may be due to the fact that CNA is very sparse. Generally speaking, in terms of both accuracy and AUC RF model gives better results than SVM for CNA data while SVM provides better results than RF for gene expression data. The integration of the gene expression data and CNA data using SVM and RF has not improved the prediction performance (**Tables 6** and **7**).

Table 6 – Accuracy (%) of the baseline models (SVM, RF) and our best performing deep learning model (DCNN_Concat as shown in Table 5) for multiclass classification. The results are shown for SVM and RF models using individual CNA data (CNA_SVM, CNA_RF) and gene expression data (Gene_SVM, Gene_RF) as well as the concatenation of both data sources (SVM_Concat and RF_Concat). The results for DCNN_Concat using the selected top genes are also shown. The best results for classifiers with a different number of top genes selected by χ^2 for CNA data and ANOVA for gene expression data are shown in bold color.

Classifier (top genes)	Test	Top selected genes								
		100	150	200	250	300	350	400	450	500
CNA_SVM	χ^2	41.7	43.2	41.9	42.6	42.6	43.2	**43.3**	42.8	43.2
CNA_RF	χ^2	46.4	48.7	47.9	48.5	49.8	49.0	**49.9**	48.2	49.2
Gene _SVM	ANOVA	72.4	75.8	75.6	75.6	**76.0**	75.4	75.9	75.5	75.9
Gene _RF	ANOVA	70.4	71.4	**71.5**	**71.5**	70.7	71.0	**71.5**	71.0	71.3
SVM_Concat		72.0	72.7	72.4	72.3	**73.4**	72.8	73.1	72.9	73.1
RF_Concat		70.1	71.7	70.4	71.1	71.1	69.6	71.7	70.8	**72.1**
DCNN_Concat		72.9	72.9	71.6	72.6	71.5	72.7	74.4	74.8	**76.6**

Table 7 - AUC of the baseline models (SVM, RF) and our best performing deep learning model (DCNN_Concat as shown in Table 5) for multiclass classification. The results are shown for SVM and RF models using individual CNA data (CNA_SVM, CNA_RF) and gene expression data (Gene_SVM, Gene_RF) as well as the concatenation of both data sources (SVM_Concat and RF_Concat). The results for DCNN_Concat using the selected top genes are also shown. The best results for classifiers with a different number of top genes selected by χ^2 for CNA data and ANOVA for gene expression data are shown in bold color.

Classifier (top genes)	Test	Top selected genes								
		100	150	200	250	300	350	400	450	500
CNA_SVM	χ^2	0.589	0.590	0.632	0.630	0.629	**0.636**	0.629	0.630	0.633
CNA_RF	χ^2	0.643	0.651	0.642	0.641	0.650	0.649	0.655	0.658	**0.662**
Gene _SVM	ANOVA	0.804	0.818	0.812	0.799	0.808	0.807	0.814	0.814	**0.819**
Gene _RF	ANOVA	0.808	**0.812**	0.806	0.804	0.810	0.806	0.802	0.798	0.804

SVM_Concat	0.810	**0.815**	0.810	0.818	0.810	0.810	**0.815**	0.814	0.814
RF_Concat	0.802	**0.810**	0.808	0.803	0.801	0.806	0.803	0.808	0.807
DCNN_Concat	0.810	0.817	0.815	0.817	0.821	0.811	0.829	0.834	**0.852**

Comparison of **Table 5** with **Tables 6** and **7** shows that when we use only individual data sources to build their DCNN models (CNA_DCNN for CNA data and Gene_DCNN for gene expression data), we get higher accuracy and AUC results than corresponding SVM and RF models (i.e. CNA_SVM and CNA_RF for CNA data and Gene_SVM and Gene_RF for gene expression data). It is also seen that our integrated models (DCNN_Concat, DCNN_Siamese, and DNN_SE) outperform the models (CNA_DCNN, Gene_DCNN, SVM and RF) built on individual datasets in terms of both accuracy and AUC.

Table 8 shows the best results from **Tables 5, 6** and **7** and the results of our baseline DNN model (DNN_DBN). It can be easily seen that the integration model DCNN_Concat outperforms overall baseline models. All our proposed DCNN models CNA_DCNN, Gene_DCNN, DCNN_Concat, DCNN_Siamese and DNN_SE (**Table 5**) also provide better prediction result than the baseline model DNN_DBN, which has accuracy 49.89% and AUC 0.625 **(Table 8)**. This may be due to the fact that the proposed DCNN models consider the correlation among the genes and the proposed DNN_SE model is less susceptible to the undesirable noisiness in the data.

Table 8 –**Performance comparison of multiclass classification between baseline models and our proposed model**. DCNN_Concat is the classifier with the best performance in our proposed

DCNN models (**Tables 5, 6 and 7**) using combined data sets. Gene_SVM, Gene_RF, RF_Concat are our baseline models with best performance shown in **Tables 6** and **7**. DNN_DBN is our baseline deep neural network model. The model with best results is bolded.

Classifier	Accuracy (%)	Classifier	AUC
DCNN_Concat (all genes)	**79.2**	DCNN_Concat (all genes)	0.850
DCNN_Concat (top 500 genes)	76.6	**DCNN_Concat (top 500 genes)**	**0.852**
Gene_SVM (top 300 genes)	76.0	Gene_SVM (top 500 genes)	0.819
RF_Concat (top 500 genes)	71.5	Gene_RF (top 150 genes)	0.812
DNN_DBN (all genes)	49.89	DNN_DBN (all genes)	0.625

The similar procedure for multiclass classification was applied to binary class classification (the classes of estrogen-receptor) and the results of the accuracies and AUCs of our DCNN, SVM and RF models are shown in **Tables 9, 10** and **11**. Generally speaking, the integration of the CNA data and gene expression data using the DCNN and SVM models have greatly improved the prediction performance, but this has not been observed for the RF models (**Table 12**). The proposed DCNN models have better performance than the SVM models and our baseline DNN model (B_DNN_DBN), but slightly worse performance than the RF models.

Table 9 - The overall accuracies (%) and AUCs of our DNN models for binary classification. B_CNA_DCNN and B_Gene_DCNN are based on the architecture of **Figure 5** for CNA data and gene expression data, respectively. B_DCNN_Concat is the DNN model based on the network architecture described in **Figure 6**.

Classifier (all genes)	Datasets	Performance Measurement	
		Accuracy (%)	AUC
B_CNA_DCNN	CNA	62.8	0.504

B_Gene_DCNN	Gene expression	62.9	0.502
B_DCNN_Concat	CNA and gene expression	96.3	0.993

Table 10 – Accuracy (%) of the baseline models (SVM, RF) and our deep learning model (B_DCNN_Concat as shown in Table 9) for binary classification. The results are shown for SVM and RF models using individual CNA data (B_SVM_CNA, B_RF_CNA) and gene expression data (B_SVM_GENE, B_RF_GENE) as well as the concatenation of both data sources (B_SVM_Concat and B_RF_Concat). The results for B_DCNN_Concat using the selected top genes are also shown. The best results for classifiers with a different number of top genes selected by χ^2 for CNA data and ANOVA for gene expression data are shown in bold color.

Classifier (top genes)	Concatenation of top selected genes from CNA and gene expression data								
	100	150	200	250	300	350	400	450	500
B_SVM_CNA	**76.8**	76.7	76.4	76.3	76.0	75.9	75.4	75.7	75.7
B_RF_CNA	82.4	82.3	82.7	82.5	**82.8**	81.7	81.5	81.8	82.7
B_SVM_GENE	72.8	72.8	72.8	72.8	72.8	72.8	72.8	72.8	72.8
B_RF_GENE	96.9	96.9	97.2	97.0	96.9	97.0	96.8	**97.1**	**97.1**
B_SVM_Concat	**95.7**	95.5	95.2	95.2	95.3	95.2	95.1	95.4	95.4
B_RF_Concat	**97.5**	97.2	97.0	97.1	97.5	96.4	96.8	96.5	97.1
B_DCNN_Concat	95.9	**96.3**	95.5	95.6	95.4	95.6	96.0	95.5	96.1

Table 11 - AUC of the baseline models (SVM, RF) and our deep learning model (B_DCNN_Concat as shown in Table 9) for binary classification. The results are shown for SVM and RF models using individual CNA data (B_SVM_CNA, B_RF_CNA) and gene expression data (B_SVM_GENE, B_RF_GENE) as well as the concatenation of both data sources (B_SVM_Concat and B_RF_Concat). The results for B_DCNN_Concat using the selected top genes are also shown. The best results for classifiers with a different number of top genes selected by χ^2 for CNA data and ANOVA for gene expression data are shown in bold color.

Classifier (top genes)	Concatenation of top selected genes from CNA and gene expression data								
	100	150	200	250	300	350	400	450	500
B_SVM_CNA	**0.601**	0.589	0.591	0.585	0.576	0.572	0.563	0.568	0.568
B_RF_CNA	0.758	0.774	0.802	0.801	0.807	0.811	0.811	0.817	**0.835**
B_SVM_GENE	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
B_RF_GENE	0.993	0.994	**0.995**	0.993	0.994	0.993	0.994	0.994	0.994
B_SVM_Concat	**0.940**	0.936	0.931	0.930	0.932	0.929	0.927	0.932	0.932
B_RF_Concat	0.994	**0.995**	0.994	**0.995**	**0.995**	0.994	**0.995**	0.993	0.992
B_DCNN_Concat	0.991	**0.992**	0.991	0.991	0.990	0.991	0.991	0.990	0.991

Table 12 – **Performance comparison of binary classification between baseline models and our proposed model**. B_DCNN_Concat, B_SVM_Concat, and B_RF_Concat are the classifiers with the best performance in our proposed DCNN models, SVM and RF models (**Tables 9, 10 and 11**) using combined data sets. B_RF_GENE is the RF model with the best performance shown in **Table 11** using individual gene expression data. B_DNN_DBN is our baseline deep neural network model. The model with best results is bolded.

Classifier	Accuracy (%)	Classifier	AUC
B_DCNN_Concat (all genes)	96.3	B_DCNN_Concat (all genes)	0.993
B_DCNN_Concat (top 150 genes)	96.3	B_DCNN_Concat (top 150 genes)	0.992
B_SVM_Concat (top 100 genes)	95.7	B_ SVM_Concat (top 100 genes)	0.940
B_RF_Concat (top 100 genes)	**97.5**	**B_ RF_Concat (top 150 genes) and B_RF_GENE (top 200 genes)**	**0.995**
B_DNN_DBN (all genes)	86.4	B_DNN_DBN (all genes)	0.522

To investigate the effects of the selection of different hyper parameter values on the prediction performance of our best DCNN model (DCNN_Concat), we consider different values for two hyperparameters: learning rate and dropout ratio. We report the parameter values with the best prediction performance. It must also be pointed out that the deep learning based models are much more expensive in terms of both computational speed and memory than SVM and RF-based models. The gene expression data and copy number variation data used for the analysis can be accessed from European Genome-phenome Archive [29].

4.7. Conclusion

Developing efficient methods to stratify cancer subtypes is necessary to provide best-personalized therapies for patients. It is expected that integration of knowledge from multiple data sources measured on the same individuals should improve the prediction performance of PAM50 intrinsic subtypes of breast cancer. In this experiment, we propose multiple deep learning-based models (DCNN_Concat, DCNN_Siamese, and DNN_SE) for

multiclass classification and B_DCNN_Concat for binary classification to integrate copy number alteration and gene expression level data measured in the same breast cancer patients to achieve this goal. Our experimental results show that integration of knowledge from these datasets into a learning method can improve the prediction of the molecular subtypes of breast cancer. The model DCNN_Concat achieves the best prediction performance among the three integration models (DCNN_Concat, DCNN_Siamese, DNN_SE) and the models (CNA_DCNN and Gene_DCNN) built using individual data sources for multiclass classification.

We also compared the prediction results of our proposed models with one integrative DNN-based model (DNN_DBN) and two other traditional machine learning models: SVM and RF. All our proposed knowledge integration models and the models built on individual datasets achieve improved prediction performance than the baseline models except the RF models show higher predictive performance for binary classification.

The proposed integrative DNN-based learning frameworks are not restricted to integrate only copy number alteration and gene expression data. They can be extended to incorporate many more data sources, such as methylation data, clinical data, etc. We will investigate this issue in more detail in the future.

Chapter 5

Triglyceride concentrations prediction using epigenome-wide DNA methylation profiles

5.1. Background

DNA methylation (DNAm) is a major epigenetic modification involving the addition of a methyl (CH3) group to the 5 position of cytosine residues in CpG (5'-cytosine-phosphate-guanine-3') dinucleotide sequences by DNA methyltransferases to form 5-methylcytosine (5-mC). In humans, DNAm is very common and 5-mC is found in approximately 1.5% of genomic DNA. The mutation of specific CpG sites is always associated with tissue-specific genes transcriptional repression, phenotype transmission and contributes to the development of different diseases by altering DNA accessibility and chromatin structure. The quantification of 5-mC content or global methylation in diseased or environmentally impacted cells could provide useful information for the understanding of disease progression and mechanisms. DNAm variation has been proposed as an epigenetic biomarker for predicting the stage of disease, to determine a patient's response to therapy, and to evaluate the prognosis [72].

Experimental and epidemiological evidence have reported that associate DNAm variations with blood lipid levels, such as high-density lipoprotein cholesterol, low-density lipoprotein cholesterol, triglycerides, and total cholesterol, by regulating the related gene of interindividual lipid levels. DNAm variations of CpG sites within *CPT1A* and *SREBF1*[2] gene promoters were linked with high triglycerides [73].

CpG sites with the high interindividual variability of DNAm can indicate the possibility of different diseases, which means these CpG sites hold the patterns that are capable of

discriminating between different phenotypes. As a heritable epigenetic mark, DNAm can explain the progress of many disease courses. Epigenome-wide DNAm has been used to predict different phenotypic traits. For example, Xu et al [74] developed a novel support vector regression model for forensic age prediction by DNAm. Wilhelm [75] proposed a machine-learning model named Model-Selection–Supervised Principle Component Analysis (MS-SPCA) to predict different stages of cervical cancer using DNAm data. To avoid a potential overfitting problem in building these models, only a small handful of CpG sites are used in the models.

Newer machine-learning methods, such as a deep neural network (DNN), can build a model using a large number of input features. These models show very promising results for several classification problems [25] in the field of computer vision. Unlike support vector machine (SVM), DNN does not require any handcrafted features and can automatically extract features from the raw input data. However, an SVM model will be likely overfitted when it is applied to methylation data with 450,000 CpG sites and only hundreds of samples because the underlying distribution is under-sampled. In this experiment, we propose DNN regression models for the prediction of triglyceride concentrations from multiple peripheral blood draws that are measured at different visits based on the individual's epigenome-wide methylation profiles that are generated before and after medication interventions.

5.2. Methods and Materials

5.2.1. Datasets

The data sets provided by Genetic Analysis Workshop 20 (GAW20) include epigenome-wide DNAm profiles and triglyceride concentrations (mg/dL) measured at the baseline level (pretreatment) of visit 2 and changes in response to treatment with fenofibrate (posttreatment) at

visit 4. The differential DNAm profiles were generated using the Illumina Infinium HumanMethylation450 BeadChip array. The beta value measuring the methylation level is expressed as a value between 0 and 1 in 993 participants of the Genetics of Lipid Lowering Drugs and Diet Network (GOLDN) study. It should be noted that there are only 499 participants with the posttreatment DNAm data. The GOLDN study recruited families with at least 2 siblings. For pretreatment data, we randomly selected 900 samples as the training set and another 93 samples as the test set; for posttreatment data, we randomly selected 400 samples as the training set and another 99 samples as the test set. We built the deep-learning models to predict triglyceride concentrations at visits 2 (pretreatment) and 4 (posttreatment) using the pretreatment DNAm data and at visit 4 using the posttreatment DNAm data. When we developed the model to predict posttreatment of triglyceride concentrations at visit 4 using pretreatment of DNAm data measured at visit 2, we only had 714 participants, from which 620 samples were randomly selected as the training set and the other 94 samples as the test set. The procedure to split the training and test sets was repeated three times to get more robust results. It should be noted that we did not use the "Answers" provided by GAW20 organizers during the analysis.

5.2.2. Regression-based prediction models

5.2.2.1 Deep-learning regression model

We proposed a DNN model (**Figure 9**) to predict individuals' triglyceride concentrations based on their epigenome-wide DNAm profiles provided by GAW20. DNN is an artificial neural network–based method, which is made up of a series of hidden layers between the input and output layers. DNN builds a hierarchy of features by producing high-level features from the low-

level features. The bottommost layer (i.e., input layer) of a DNN takes the raw input data and each next hidden layer learns an abstract form of the data from the previous layer.

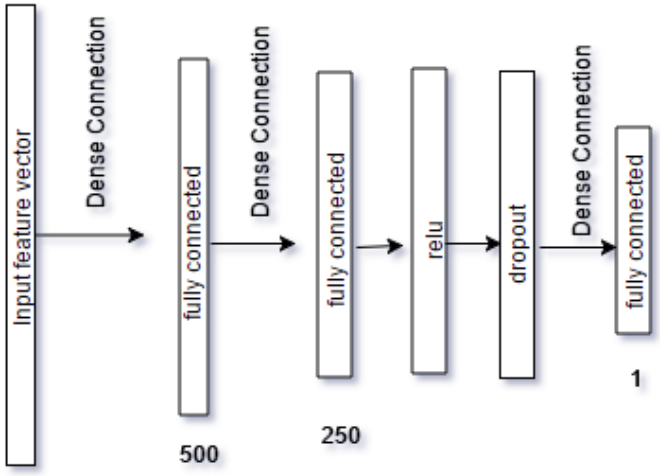


Figure 9 - Proposed architecture of DNN. The numbers shown in the figure represent the size of the output of each layer.

The input of our proposed DNN network is a vector of the epigenome-wide DNAm profile of a given sample. Because the feature vector is quite high dimensional (>450,000), we passed this input vector to two fully connected layers with different output sizes to reduce its dimension. These outputs can be thought as a matrix multiplication for getting a high-level abstraction of the information in the input vector.

Because of the complex nonlinear relationship between triglyceride concentrations and genome-wide DNAm, we used a ReLU (rectified linear unit) layer followed by the second fully connected layer. The ReLU layer performs a ReLU thresholding function over the output of the second fully connected layer. The output of the ReLU layer is the nonlinear representation of the input to the network (see **Figure 9**). ReLU function follows the **Equation 1.**

To provide generalization ability over the test data to the network we used a regularization technique called Dropout [76]. Dropout layers randomly drop out hidden neurons

from the network. This technique allows the network to overcome the curse of overfitting because the network has to learn fewer parameters. Consequently, the output from the ReLU layer in our network was subjected to the dropout regularization technique by applying a dropout layer.

To get the final predictions of triglyceride concentrations we passed the output of the dropout layer to the last layer of the network, which is also a fully connected layer. We considered the score of this layer as the prediction of the network. Instead of using a greedy layer-wise (layer-by-layer) approach to training our network, we used a Euclidean loss layer to train our network in a backpropagation style. In this case, each layer of our DNN took an input and performed a transformation of the input to produce an output. This output was then used as an input to the next layer and so on until the loss layer was reached. This loss layer computed an error over its input data with respect to the ground truth value. Finally, a remedial gradient with respect to the error value was passed down to the DNN network to update its parameter values.

5.2.2.1 SVM model

SVM is a supervised learning algorithm that was initially developed to solve classification problems but later was extended to solve regression problems [77]. SVM regression maintains all the key features that characterize the maximal margin theory and avoids difficulties of using linear functions in the high-dimensional feature space by transforming the optimization problem into dual convex quadratic programs. The loss function in SVM regression, which is used to penalize errors, usually leads to the sparse representation of the decision rule. This gives significant algorithmic and representational advantages over other regression methods.

5.2.3. Feature selection for DNN and SVM

For each sample, we have 463,995 CpG sites. As we know that CpG sites with high interindividual variability hold the most discriminative information [78], we defined the interindividual variability (I_v) as the difference between 90th percentile and 10th percentile of the DNAm of a given CpG.

We built DNN models based on the selected CpG sites with I_v greater than or equal to different cutoffs of DNAm values (minimum [no filtering], first quartile, second quartile, mean, and third quartile). For each of these cutoff points we had 463,995, 348,223, 232,131, 165,817 and 116,057 CpG sites in the pretreatment data set and 463,995, 348,252, 231,901, 157,073 and 116,054 CpG sites in the posttreatment data set. The distributions of the interindividual variability I_v of DNAm in all CpG sites are shown in **Figure 10**. **Figure 10** clearly shows that the DNAm of a majority of the CpG sites has very small variation across samples. SVM will likely overfit the regression models if we use all 463,995 CpG sites to train the models. Consequently, we selected hundreds of the top CpG sites with larger interindividual variability of DNAm to build the SVM regression models.

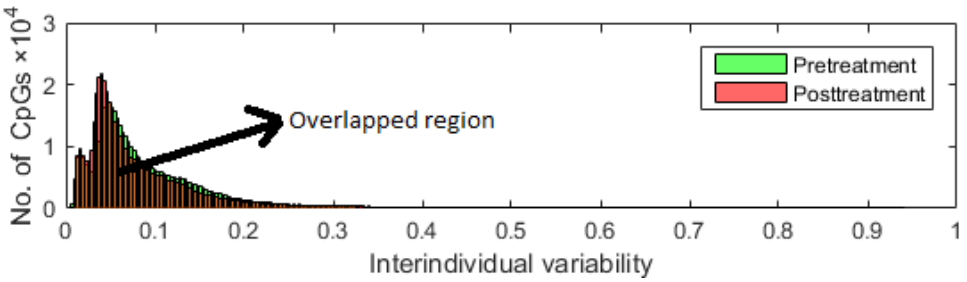


Figure 10 - Distribution of interindividual variability of DNAm for pretreatment and posttreatment.

5.3. Building the DNN

We used CAFFE, which is a C++-based deep-learning library, to implement the DNN models (see **Figure 9**) for different cutoffs of interindividual variability of DNAm (see **Figure 10**). We trained all our DNN models using a learning rate (defined in the context of optimization, and minimizing the loss function of a neural network) of 0.000001, batch size (the number of training samples used in a single iteration/forward pass) of 10 and dropout ratio of 0.5.

5.4. Performance evaluation

We used root mean square error (RMSE) and Pearson correlation (Cor) methods to compare the performance of our DNN and standard SVM regression models. Gal and Ghahramani [76] also used RMSE to measure the performance of their deep-learning-based regression models. RMSE can be calculated as follows:

$$\text{RMSE}=\sqrt{mean((y-\hat{y})^2)} \quad (8)$$

Here, y represents the observed triglyceride concentrations at different blood draws and $\hat{y}$ represents the predicted triglyceride concentrations at different blood draws.

Cor was calculated between y and $\hat{y}$. We performed three random splits between training and test data. The results of RMSE and Cor were averaged and their SDs were estimated. We used R package e1071 [62] to build the SVM regression models (default parameters were used). Models with smaller RMSE or higher Cor are preferable and have better prediction performance.

5.5. Results and discussion

The *p* values of the Shapiro test of the log (base 2) of observed triglyceride concentrations in test sets from **Case A**(pretreatment DNAm data to predict the triglyceride levels measured at visit 2),**Case B**(pretreatment DNAm data to predict the triglyceride levels measured at visit 4), and **Case C**(posttreatment DNAm data to predict the triglyceride levels measured at visit 4), were 0.17, 0.25, and 0.25, respectively, suggesting that the observed triglyceride levels followed log-normal distribution. We performed the same procedure on their averaged predicted values from the three splits of training and test sets using the SVM models with largest Cor values (bold in **Table 13**) and the DNN model with largest Cor values (bold in **Table 14**) and the *p* values for **Case A**, **Case B**, and **Case C** were 0.09, 0.05, and 0.78, respectively, for SVM models, and 0.08, 0.14, and 0.59, respectively, for DNN models, which suggest that the predicted triglyceride levels using either DNN models or SVM models also follow log-normal distribution. The scatter plots of the observed and predicted triglyceride levels for **Case A**, **Case B**, and **Case C** are shown in **Figure 11**.

Table 13: Performance of SVM models

Data[1]	Evaluation Metric[2]	Cutoffs[3]				
		100	200	300	400	500
1	RMSE	**90.3**(27.5)[4]	90.9(28.8)	90.9(29.2)	90.8(28.8)	95.8(23.8)
	Cor	**0.13**(0.06)	0.11(0.12)	0.11(0.14)	0.11(0.14)	0.10(0.13)
2	RMSE	**48.7**(13.7)	49.4(12.9)	49.0(12.9)	**48.7**(12.8)	50.1(14.3)
	Cor	**0.19**(0.08)	0.12(0.10)	0.15(0.06)	0.17(0.05)	0.04(0.20)
3	RMSE	48.0(7.2)	47.6(7.0)	47.5(6.9)	**46.9**(7.0)	47.0(6.9)
	Cor	0.04(0.08)	0.07(0.09)	0.07(0.10)	**0.13**(0.10)	0.12(0.12)

[1]Data 1. Pretreatment DNAm data to predict the triglyceride levels measured at visit 2;

Data 2. Pretreatment DNAm data to predict the triglyceride levels measured at visit 4;

Data 3. Posttreatment DNAm data to predict the triglyceride levels measured at visit 4.

[2]RMSE: root-mean-squared-error; Cor: Pearson correlation between observed and predicted values.

[3]The top number of CpGs selected based on inter-individual variability.

[4]The averaged RMSE or Cor value and their standard deviation (SD) from the three splits of training and test sets. The bold value indicates the model has the best performance across a different number of selected CpGs at the given DNAm data set and performance metric.

Table 14: Performance of DNN models

Data	Evaluation Metric	Cutoffs[1]						
		Min	1st quartile	Mean	Median	3rd quartile	10kCpGs	1kCpGs
1	RMSE	**88.5**(26.3)	88.8(25.6)	89.3(25.7)	89.0(27.3)	88.8(26.1)	89.2(25.9)	89.8(26.4)
	Cor	0.19(0.05)	**0.27**(0.08)	0.19(0.09)	0.14(0.11)	0.11(0.10)	0.24(0.02)	0.14(0.11)
2	RMSE	48.5(14.4)	48.4(14.7)	**47.4**(13.7)	48.5(14.3)	47.5(13.8)	48.6(12.9)	48.8(13.0)
	Cor	0.23(0.13)	0.10(0.29)	**0.29**(0.07)	0.14(0.19)	0.29(0.07)	0.20(0.11)	0.10(0.14)
3	RMSE	48.5(4.7)	48.7(4.8)	48.5(4.5)	**48.1**(3.5)	48.6(4.6)	48.2(5.0)	48.5(5.3)
	Cor	0.17(0.07)	0.18(0.08)	**0.22**(0.13)	0.20(0.12)	0.19(0.08)	0.17(0.06)	0.16(0.04)

[1]The selected CpGs with inter-individual variability greater than or equal to different cutoffs of DNAm values (minimum (no filtering), 1st quartile, 2nd quartile, mean and 3rd quartile) as well as the top 10,000 CpGs (10kCpGs) and top 1,000 CpGs (1kCpGs).

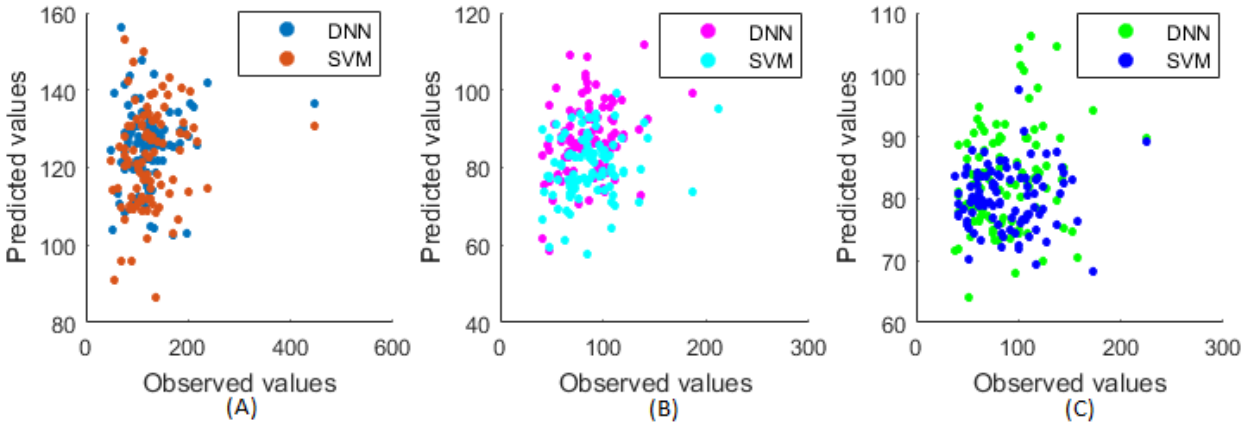


Figure 11 - Scatter plots of the observed triglyceride levels (mg/dl) and their predicted triglyceride levels (mg/dl).

The prediction results (RMSE and Cor) of the SVM and our DNN models using a different number of CpG sites with the larger interindividual variability of DNAm are shown in **Tables 13** and **14**, respectively. In general, the number of CpG sites used in each model (DNN or SVM) has little effect on the prediction performance as measured by RMSE of the triglyceride concentrations measured at a specific visit. However, the number of CpG sites used in each model (DNN or SVM) does impact the prediction performance measured by Cor. For example, the SVM models with a larger number of CpG sites (eg,500) have poorer performance than those with a smaller number of CpG sites (eg, 100; see **Table 13**), but the DNN models with a larger number of CpG sites (eg, 165,817) have much better performance than those with a smaller number of CpG sites (eg,1000; see **Table 14**). Comparison of the performances of our DNN models (see **Table 14**) with those of SVM models (see **Table 13**) shows that our proposed models have a lower RMSE and a higher correlation between predicted and observed triglyceride concentrations, which suggests that our DNN models have better prediction performance than do the SVM models.

Overall, using DNN and SVM models to predict triglyceride concentrations with DNAm profiles has worse performance at visit 2 than at visit 4. Remarkably, our DNN results (the averaged RMSE and Cor) show that the performances of using pre- and posttreatment DNAm to predict triglyceride levels at visit 4 are similar. For example, the best performance of using pretreatment DNAm to predict triglyceride levels at visit 4 is 47.4 for RMSE and 0.29 for Cor while the best performance of using posttreatment DNAm to predict triglyceride levels at visit 4 is 48.1 for RMSE and 0.22 for Cor. Furthermore, this finding also shows that pretreatment DNAm has the slightly better capability to predict triglyceride levels than posttreatment DNAm at visit 4. These results have two potential implications: (a) the variation of DNAm may not be altered greatly as a result of treatment, and (b) early DNAm variation could predict the internal response of the individuals to lipid-lowering drugs. Consequently, DNAm may have a long-term effect on genome sequence under exposure to early environmental experiences that were associated with stable changes in the gene expression that emerged in the initial stage of disease and were sustained into later stages. Much research [79] supports the long-term epigenetic effect on genomes, making the DNAm profile usable as the epigenetic marker to predict development and prognoses of diseases.

5.6. Conclusions

This study proposed a DNN architecture for predicting triglyceride concentrations, a complex phenotypic trait, using epigenome-wide DNAm profiles measured at different patient visits for a blood draw. The new model framework has advantages over some traditional learning algorithms (such as SVM), which are prone to overfitting when the input data are quite high dimensional. We showed that DNAm profiles measured at pretreatment and posttreatment have a

better capability to predict triglyceride concentrations measured from blood drawn at visit 4 than do DNAm profiles measured at pretreatment to predict triglyceride concentrations measured from blood drawn at visit 2. We also found that DNAm profiles measured at pretreatment can predict triglyceride concentrations measured from blood drawn at visit 4 more accurately than DNAm profiles measured at posttreatment, which suggests a long-term epigenetic effect on phenotypic traits. The limitations of the study are that the proposed model neither considered the familial relationships of the participants in the study nor explored the usefulness of the available genetic data to predict the triglyceride levels. We will investigate whether the DNN model is sensitive to the familial structure and integrate both genetic and methylation data to predict triglyceride levels in the future.

Chapter 6

Conclusion and future work

Classification of molecular subtypes of breast cancer using omics profiles is a challenging problem since the data sets are quite high-dimensional and highly correlated. The curse of high-dimensionality also affects the performances of predicting a phenotype using DNAm data. Traditional machine learning methods, such as SVM and RF, have limitations in handling high-dimensional and highly correlated data sets. Recently, DNN learning has been demonstrated advantages over these methods as it does not require any hand-crafted features, but rather automatically extracts features from the raw data and efficiently analyzes high-dimensional and correlated data. In this thesis, we have developed several DNN frameworks for classifying molecular subtypes of breast cancer using only one data source or two heterogeneous data sources as input to the frameworks. In addition, we also developed a DNN-based regression framework which takes epigenome-wide DNAm data as input to predict triglyceride concentrations in our blood.

From the project called METABRIC [60] we collected two omics profiles: CNA and gene expression profiles measured on the same set of breast cancer patients. We have used only CNA profiles to build a DNN model to predict a patient's ER status as well as the status of PAM50 subtypes. In both cases, our proposed models provide improved prediction performance than the baseline models: SVM and RF. We also introduced multiple integrative DNN frameworks which take both CNA and gene expression profiles of a patient as input to predict

the patient's ER status and PAM50 subtypes. Our proposed integrative frameworks outperform three baseline models (i.e. SVM, RF, and DNN_DBN) when predicting the PAM50 subtypes. Experimental results show that integrative frameworks are superior to those that use only one data source in predicting breast cancer subtypes. Our proposed integrative frameworks are not limited to CNA and gene expression profiles. In future, we will use other omics data (e.g. DNAm) in our integrative DNN frameworks.

The epigenetic modification has an effect on gene expression under the environmental alteration, but it does not change corresponding genome sequence. DNAm is one of the important epigenetic mechanisms. Predictions of phenotypic traits, e.g. blood pressure and triglyceride concentrations in human blood, can be done using the variations in DNAm data. In the thesis, we proposed DNN-based regression frameworks which take epigenome-wide DNAm data of a patient as input to predict triglyceride concentration (before and after medication). We used pretreatment (i.e. before medication) and posttreatment (i.e. after medication) DNAm data to predict pretreatment and posttreatment triglyceride concentrations in a patient's blood. In both cases, our proposed DNN-based regression model provides improved prediction performance than the baseline SVM models. Our framework also uses pretreatment DNAm data to predict posttreatment triglyceride concentrations. In this case, our framework gives the best prediction performances than the above two cases. Therefore, pretreatment DNAm data is more capable to predict posttreatment triglyceride concentrations than posttreatment DNAm data. This outcome implies that the treatment did not properly altered DNAm variations as well as advises long-term epigenetic consequence on phenotypic traits. We did not consider the familial relationships of the participants during the model building. In future, we will incorporate this information and other genetic available data with DNAm data into our framework.

References

1. Knasmüller S, Nersesyan A, Mišík M, Gerner C, Mikulits W, Ehrlich V, Hoelzl C, Szakmary A, Wagner KH. Use of conventional and-omics based methods for health claims of dietary antioxidants: a critical overview. British Journal of Nutrition. 2008 May;99(E-S1):ES3-52.
2. Schneider MV, Orchard S. Omics technologies, data and bioinformatics principles. Bioinformatics for Omics Data: Methods and Protocols. 2011:3-0.
3. Lippert C, Sabatini R, Maher MC, Kang EY, Lee S, Arikan O, Harley A, Bernal A, Garst P, Lavrenko V, Yocum K. Identification of individuals by trait prediction using whole-genome sequencing data. Proceedings of the National Academy of Sciences. 2017 Sep 19;114(38):10166-71.
4. Chen YC, Douville C, Wang C, Niknafs N, Yeo G, Beleva-Guthrie V, Carter H, Stenson PD, Cooper DN, Li B, Mooney S. A probabilistic model to predict clinical phenotypic traits from genome sequencing. PLoS computational biology. 2014 Sep 4;10(9):e1003825.
5. Liu F, Wen B, Kayser M. Colorful DNA polymorphisms in humans. InSeminars in cell & developmental biology 2013 Jul 31 (Vol. 24, No. 6, pp. 562-575). Academic Press.
6. Hart KL, Kimura SL, Mushailov V, Budimlija ZM, Prinz M, Wurmbach E. Improved eye-and skin-color prediction based on 8 SNPs. Croatian medical journal. 2013 Jun 15;54(3):248-56.
7. Claes P, Liberton DK, Daniels K, Rosana KM, Quillen EE, Pearson LN, McEvoy B, Bauchet M, Zaidi AA, Yao W, Tang H. Modeling 3D facial shape from DNA. PLoS genetics. 2014 Mar 20;10(3):e1004224.
8. Breast Cancer Information and Awareness. http://www.breastcancer.org. Accessed on 20 January 2017.
9. Perou CM, Sørlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA, Fluge Ø. Molecular portraits of human breast tumours. Nature. 2000 Aug 17;406(6797):747-52.
10. Parker JS, Mullins M, Cheang MC, Leung S, Voduc D, Vickery T, Davies S, Fauron C, He X, Hu Z, Quackenbush JF. Supervised risk predictor of breast cancer based on intrinsic subtypes. Journal of clinical oncology. 2009 Feb 9;27(8):1160-7.
11. Milioli HH, Vimieiro R, Tishchenko I, Riveros C, Berretta R, Moscato P. Iteratively refining breast cancer intrinsic subtypes in the METABRIC dataset. BioData mining. 2016 Jan 13;9(1):2
12. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. InAdvances in neural information processing systems 2012 (pp. 1097-1105).

13. Khosla A, Zhou T, Malisiewicz T, Efros AA, Torralba A. Undoing the damage of dataset bias. InEuropean Conference on Computer Vision 2012 Oct 7 (pp. 158-171). Springer, Berlin, Heidelberg.
14. Saenko K, Kulis B, Fritz M, Darrell T. Adapting visual category models to new domains. Computer Vision–ECCV 2010. 2010:213-26.
15. Aytar Y, Zisserman A. Tabula rasa: Model transfer for object category detection. InComputer Vision (ICCV), 2011 IEEE International Conference on 2011 Nov 6 (pp. 2252-2259). IEEE.
16. Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T. Decaf: A deep convolutional activation feature for generic visual recognition. InInternational conference on machine learning 2014 Jan 27 (pp. 647-655).
17. Sharif Razavian A, Azizpour H, Sullivan J, Carlsson S. CNN features off-the-shelf: an astounding baseline for recognition. InProceedings of the IEEE conference on computer vision and pattern recognition workshops 2014 (pp. 806-813).
18. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229. 2013 Dec 21.
19. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. InProceedings of the IEEE conference on computer vision and pattern recognition 2014 (pp. 580-587).
20. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580. 2012 Jul 3.
21. Wan L, Zeiler M, Zhang S, Cun YL, Fergus R. Regularization of neural networks using dropconnect. InProceedings of the 30th international conference on machine learning (ICML-13) 2013 (pp. 1058-1066).
22. Hinton GE. Training products of experts by minimizing contrastive divergence. Training. 2006 Mar 30;14(8).
23. Larochelle H, Erhan D, Courville A, Bergstra J, Bengio Y. An empirical evaluation of deep architectures on problems with many factors of variation. InProceedings of the 24th international conference on Machine learning 2007 Jun 20 (pp. 473-480). ACM.
24. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research. 2010;11(Dec):3371-408.
25. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC. Imagenet large scale visual recognition challenge. International Journal of Computer Vision. 2015 Dec 1;115(3):211-52.

26. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence. 2017 Jun 1;39(6):1137-49.
27. Abdel-Hamid O, Mohamed AR, Jiang H, Deng L, Penn G, Yu D. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing. 2014 Oct;22(10):1533-45.
28. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. Journal of Machine Learning Research. 2011;12(Aug):2493-537.
29. Maienschein-Cline M, Zhou J, White KP, Sciammas R, Dinner AR. Discovering transcription factor regulatory targets using gene expression and binding data. Bioinformatics. 2011 Nov 13;28(2):206-13.
30. Danaee P, Ghaeini R, Hendrix DA. A DEEP LEARNING APPROACH FOR CANCER DETECTION AND RELEVANT GENE IDENTIFICATION. InPacific Symposium on Biocomputing. Pacific Symposium on Biocomputing 2016 (Vol. 22, p. 219). NIH Public Access.
31. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research. 2010;11(Dec):3371-408.
32. Yuan Y, Shi Y, Li C, Kim J, Cai W, Han Z, Feng DD. DeepGene: an advanced cancer type classifier based on deep learning and somatic point mutations. BMC bioinformatics. 2016 Dec 23;17(17):476.
33. Liang M, Li Z, Chen T, Zeng J. Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB). 2015 Jul 1;12(4):928-37.
34. American Cancer Society.Cancer facts & figures 2016. Atlanta, American Cancer Society 2016.http://www.cancer.org/acs/groups/content/@research/documents/document/acspc047079.pdf.
35. Stern RS. Prevalence of a history of skin cancer in 2007: results of an incidence-based model. Archives of dermatology. 2010 Mar 1;146(3):279-82.
36. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017 Feb 2;542(7639):115-8.
37. Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J, Madabhushi A. Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images. IEEE transactions on medical imaging. 2016 Jan;35(1):119-30.
38. Chong YT, Koh JL, Friesen H, Duffy SK, Cox MJ, Moses A, Moffat J, Boone C, Andrews BJ. Yeast proteome dynamics from single cell imaging and automated analysis. Cell. 2015 Jun 4;161(6):1413-24.

39. Kraus OZ, Grys BT, Ba J, Chong Y, Frey BJ, Boone C, Andrews BJ. Automated analysis of high-content microscopy data with deep learning. Molecular systems biology. 2017 Apr 1;13(4):924.
40. Wang S, Sun S, Li Z, Zhang R, Xu J. Accurate de novo prediction of protein contact map by ultra-deep learning model. PLoS computational biology. 2017 Jan 5;13(1):e1005324.
41. Seemayer S, Gruber M, Söding J. CCMpred—fast and precise prediction of protein residue-residue contacts from correlated mutations. Bioinformatics. 2014 Jul 26;30(21):3128-30.
42. Jones DT, Singh T, Kosciolek T, Tetchner S. MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. Bioinformatics. 2014 Nov 26;31(7):999-1006.
43. Sun T, Zhou B, Lai L, Pei J. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. BMC bioinformatics. 2017 May 25;18(1):277.
44. Eser U, Churchman LS. FIDDLE: An integrative deep learning framework for functional genomic data inference. bioRxiv. 2016 Jan 1:081380.
45. Malabat C, Feuerbach F, Ma L, Saveanu C, Jacquier A. Quality control of transcription start site selection by nonsense-mediated-mRNA decay. Elife. 2015 Apr 23;4:e06722.
46. Chen Y, Li Y, Narayan R, Subramanian A, Xie X. Gene expression inference with deep learning. Bioinformatics. 2016 Feb 11;32(12):1832-9.
47. Godinez WJ, Hossain I, Lazic SE, Davies JW, Zhang X. A multi-scale convolutional neural network for phenotyping high-content cellular images. Bioinformatics. 2017 Feb 15:btx069.
48. Qin Q, Feng J. Imputation for transcription factor binding predictions based on deep learning. PLoS computational biology. 2017 Feb 24;13(2):e1005403.
49. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature biotechnology. 2015 Aug 1;33(8):831-8.
50. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning-based sequence model. Nature methods. 2015 Oct 1;12(10):931-4.
51. Oakden-Rayner L, Carneiro G, Bessen T, Nascimento JC, Bradley AP, Palmer LJ. Precision Radiology: Predicting longevity using feature engineering and deep learning methods in a radiomics framework. Scientific Reports. 2017 May 10;7(1):1648.
52. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In European conference on computer vision 2014 Sep 6 (pp. 818-833). Springer, Cham.
53. Calo E, Wysocka J. Modification of enhancer chromatin: what, how, and why?. Molecular cell. 2013 Mar 7;49(5):825-37.
54. Yang B, Liu F, Ren C, Ouyang Z, Xie Z, Bo X, Shu W. BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone. Bioinformatics. 2017 Feb 17:btx105.

55. Lin C, Jain S, Kim H, Bar-Joseph Z. Using neural networks for reducing the dimensions of single-cell RNA-Seq data. Nucleic Acids Research. 2017 Sep 29;45(17):e156-.
56. Beroukhim R, Getz G, Nghiemphu L, Barretina J, Hsueh T, Linhart D, Vivanco I, Lee JC, Huang JH, Alexander S, Du J. Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. Proceedings of the National Academy of Sciences. 2007 Dec 11;104(50):20007-12.
57. Han L, Yuan Y, Zheng S, Yang Y, Li J, Edgerton ME, Diao L, Xu Y, Verhaak RG, Liang H. The Pan-Cancer analysis of pseudogene expression reveals biologically and clinically relevant tumour subtypes. Nature communications. 2014;5:3963.
58. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. Journal of machine learning research. 2014 Jan 1;15(1):1929-58.
59. Denas O, Taylor J. Deep modeling of gene expression regulation in an erythropoiesis model. InRepresentation Learning, ICML Workshop 2013.
60. Curtis C, Shah SP, Chin SF, Turashvili G, Rueda OM, Dunning MJ, Speed D, Lynch AG, Samarajiwa S, Yuan Y, Gräf S. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. Nature. 2012 Jun 21;486(7403):346-52.
61. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. InProceedings of the 22nd ACM international conference on Multimedia 2014 Nov 3 (pp. 675-678). ACM.
62. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F. e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien, 2015. R package version.:1-6.
63. Cutler DR, Edwards TC, Beard KH, Cutler A, Hess KT, Gibson J, Lawler JJ. Random forests for classification in ecology. Ecology. 2007 Nov 1;88(11):2783-92.
64. Hand DJ, Till RJ. A simple generalization of the area under the ROC curve for multiple class classification problems. Machine learning. 2001 Nov 1;45(2):171-86.
65. Futreal PA, Coin L, Marshall M, Down T, Hubbard T, Wooster R, Rahman N, Stratton MR. A census of human cancer genes. Nature Reviews Cancer. 2004 Mar 1;4(3):177-83.
66. Yeung KY, Ruzzo WL. Principal component analysis for clustering gene expression data. Bioinformatics. 2001 Sep 1;17(9):763-74.
67. Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome research. 2016 Jul 1;26(7):990-9.

68. Zeng H, Gifford DK. Discovering DNA motifs and genomic variants associated with DNA methylation. bioRxiv. 2016 Jan 1:073809.
69. Leung MK, Xiong HY, Lee LJ, Frey BJ. Deep learning of the tissue-regulated splicing code. Bioinformatics. 2014 Jun 11;30(12):i121-9.
70. Durinck S, Spellman PT, Birney E, Huber W. Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Nature protocols. 2009 Jul 1;4(8):1184-91.
71. Rong X. deepnet: deep learning toolkit in R. R package version 0.2. 2016.
72. Mikeska T, Craig JM. DNA methylation biomarkers: cancer and beyond. Genes. 2014 Sep 16;5(3):821-64.
73. Dekkers KF, van Iterson M, Slieker RC, Moed MH, Bonder MJ, Van Galen M, Mei H, Zhernakova DV, van den Berg LH, Deelen J, van Dongen J. Blood lipids influence DNA methylation in circulating cells. Genome biology. 2016 Jun 27;17(1):138.
74. Xu C, Qu H, Wang G, Xie B, Shi Y, Yang Y, Zhao Z, Hu L, Fang X, Yan J, Feng L. A novel strategy for forensic age prediction by DNA methylation and support vector regression model. Scientific reports. 2015;5.
75. Wilhelm T. Phenotype prediction based on genome-wide DNA methylation data. BMC bioinformatics. 2014 Jun 17;15(1):193.
76. Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning 2016 Jun 11 (pp. 1050-1059).
77. Gunn SR. Support vector machines for classification and regression. ISIS technical report. 1998 May 10;14:85-6.
78. Feinberg AP, Irizarry RA, Fradin D, Aryee MJ, Murakami P, Aspelund T, Eiriksdottir G, Harris TB, Launer L, Gudnason V, Fallin MD. Personalized epigenomic signatures that are stable over time and covary with body mass index. Science translational medicine. 2010 Sep 15;2(49):49ra67-.
79. Kanherkar RR, Bhatia-Dey N, Csoka AB. Epigenetics across the human lifespan. Frontiers in cell and developmental biology. 2014;2.