Latent Feature Construction for Gene Expressions Improves Predictions

Christos Tselas

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science

University of Crete
School of Sciences and Engineering
Computer Science Department
University Campus, Voutes, Heraklion, GR-70013, Greece

Thesis Supervisor: Associate Professor *Ioannis Tsamardinos*

Heraklion, October 2017

This work was partially funded by the ERC Consolidator Grant No 617393 CAUSALPATH and the Toshiba project: "Feasibility study towards the Next Generation of statistical Text to Speech Synthesis System"

University of Crete
Computer Science Department

Latent Feature Construction for Gene Expressions Improves Predictions

Thesis submitted by
Christos Tselas
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Christos Tselas

Committee approvals: _____
Ioannis Tsamardinos
Associate Professor, Thesis Supervisor

_____
Georgios Tziritas
Professor, Committee Member

_____
Ioannis Stylianou
Professor, Committee Member

Departmental approval: _____
Antonios Argyros
Professor, Director of Graduate Studies

Heraklion, October 2017

Abstract

Gene expression analysis aims to improve the understanding of the intrinsic cellular processes and contribute towards the successful implementation of personalized medicine. The advent of high-throughput gene expression technologies such as microarrays and RNA-sequencing (RNAseq) as well as the recent reduction of cost resulted in an explosion of publicly-available datasets. The generated datasets are inevitably high-dimensional with typically small sample size that severely limits the potential for developing reproducible prognostic models. Being able to increase the predictive power without losing the information of the measured genome on a newly-produced dataset is of paramount importance. Despite the fact that various studies attempt to perform dimensionality reduction and dataset integration so as to increase classification performance and robustness, there are still challenging issues primarily due to the limited number of data as well as the technological diversity and heterogeneity across the datasets.

Exploiting the redundancy of genomics data, we constructed low-dimensional, universal, latent feature spaces of the genome utilizing several dimensionality reduction approaches and a diverse set of curated datasets. Standard Principal Component Analysis (PCA), kernel PCA and Neural Network Autoencoders were applied on datasets from four different platforms. While linear techniques showed better reconstruction performance, nonlinear approaches were able to capture more complex gene interactions, and thus enjoyed stronger classification power. When newly-seen gene expression datasets projected to a latent space of 200 dimensions, the classification power was improved. Moreover, we performed a large-scale experiment where the dimensionality reduction methods were trained on an integrated set of 59864 unique samples. The classification power was further improved especially for Autoencoder. Rather surprisingly, the statistical variability of the additional datasets increased the classification performance implying that intricate biological features were better learn. We additionally tested the possibility of cross-platform data augmentation by constructing an intermediate feature space showing that when platforms share common characteristics (such as GLP570 and GLP96) the predictive performance was also improved.

Περίληψη

Η ανάλυση γονιδιακών εκφράσεων στοχεύει στη βελτίωση της κατανόησης των ενδογενών κυτταρικών διεργασιών και συμβάλλει στην επιτυχή εφαρμογή της εξατομικευμένης ιατρικής. Η εμφάνιση των τεχνολογιών γονιδιακών εκφράσεων υψηλών αποδόσεων όπως οι μικροσυστοιχίες (microarrays) και η αλληλουχία RNA (RNAseq) καθώς και η πρόσφατη μείωση του κόστους οδήγησαν στην έκρηξη δημόσιων-διαθέσιμων συνόλων δεδομένων. Τα παραγόμενα σύνολα δεδομένων είναι αναπόφευκτα μεγάλης διαστάσεως με τυπικά μικρό μέγεθος δείγματος που περιορίζει σοβαρά τις δυνατότητες δημιουργίας αναπαραγώγισιμων προγνωστικών μοντέλων. Η δυνατότητα αύξησης της προβλεπτικής ισχύος χωρίς απώλεια πληροφοριών του μετρηθέντος γονιδιώματος σε ένα νεοσύστατο σύνολο δεδομένων είναι ύψιστης σημασίας. Παρά το γεγονός ότι διάφορες μελέτες έχουν προσπαθήσει να επιτύχουν μείωση των διαστάσεων και συγχώνευση συνόλων δεδομένων, ώστε να αυξηθεί η απόδοση και η ευρωστία της ταξινόμησης, εξακολουθούν να υπάρχουν προκλήσεις κυρίως λόγω του περιορισμένου αριθμού δεδομένων καθώς και της τεχνολογικής ποικιλομορφίας και ετερογένειας στα σύνολα δεδομένων.

Αξιοποιώντας την πλεοναστικότητα των γονιδιακών δεδομένων, κατασκευάσαμε καθολικούς κρυμμένους χώρους μικρότερων διαστάσεων του γονιδιώματος, χρησιμοποιώντας διάφορες προσεγγίσεις μείωσης των διαστάσεων και ένα ποικίλο σύνολο συνόλων δεδομένων. Οι τεχνικές Principal Component Analysis (PCA), kernel PCA και Neural Network Autoencoders εφαρμόστηκαν σε σύνολα δεδομένων από τέσσερις διαφορετικές πλατφόρμες. Ενώ οι γραμμικές τεχνικές έδειξαν καλύτερες επιδόσεις ανασυγκρότησης, οι μη γραμμικές προσεγγίσεις ήταν σε θέση να καταγράψουν πιο πολύπλοκες γονιδιακές αλληλεπιδράσεις, απολαμβάνοντας έτσι ισχυρότερη προβλεπτική δύναμη. Όταν νεοφανή σύνολα γονιδιακών εκφράσεων προβάλλονται σε ένα κρυμμένο χώρο 200 διαστάσεων, η προβλεπτική ισχής βελτιώθηκε. Επιπλέον, πραγματοποήσαμε ενα πείραμα μεγάλης κλίμακας, όπου οι μεθοδοι μείωσης των διαστάσεων εκπαιδεύτηκαν σε ένα σύνολο 59864 μοναδικών δειγμάτων. Η ισχύς ταξινόμησης βελτιώθηκε περαιτέρω ειδικά για την τεχνική Autoencoder. Απροσδόκητα, η στατιστική μεταβλητότητα των πρόσθετων συνόλων δεδομένων αύξησε την απόδοση ταξινόμησης, υπονοώντας ότι μαθεύτηκαν καλύτερα περίπλοκα βιολογικά χαρακτηριστικά. Επιπλέον, εξετάσαμε τη δυνατότητα αύξησης των δεδομένων χρησιμοποιόντας δεδομενα από διάφορες πλατφόρμες, κατασκευάζοντας έναν ενδιάμεσο χώρο χαρακτηριστικών που δείχνει ότι οταν οι πλατφόρμες μοιράζονται κοινά χαρακτηριστικά (όπως GPL570 και GPL96) βελτιώνεται η προβλεπτική απόδοση.

Acknowledgements

First of all, I would like to thank my thesis advisor Associate Professor Ioannis Tsamardinos and co-advisor, Dr Yannis Pantazis. During the last year, they were always happy and willing to help me solve the confusions and direct me approach to the final result of this thesis.

I would also like to thank all members of Data Analysis Laboratory ("Mens X Machina") at University of Crete for all their instructions. Especially G. Borboudakis and P. Charonyktakis for their assistance with JAD Bio tool;G. Papoutsoglou and V. Lagani for their help and advice on Gene Set Enrichment Analysis.

In addition, I need to show my gratitude to my dissertation committee, Professor Georgios Tziritas and Professor Ioannis Stylianou.

Finally, I must express my very profound gratitude to my parents Romeo and Mirela for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Gene expression is the process by which genetic instructions synthesize gene products such as proteins and hence controls the various cell mechanisms [1]. The information encoded in gene expression data motivates scientists to computationally analyze them and extract new biological knowledge. High-throughput technologies such as microarrays [2] and RNA sequencing [3] measure gene expression profiles in a fast and automated manner. The objectives of gene expression analysis range from improving the identification of biomarkers which are differentially expressed genes, to increasing the classification accuracy and determine the disease of a person/sample as well as to obtain qualitative clusters of similar populations. Unfortunately, the produced genetic datasets typically have low sample size due to the limited availability of patients (i.e., samples) as well as the formerly expensive measurement process. Furthermore, gene expression datasets are high dimensional which entails not only high computational requirements but also sophisticated algorithms since the large number of variables presents an intrinsic challenge to classification problems. These limitations of individual genetic studies made computational tasks such as disease classification less accurate and statistically not robust.

Despite being high-dimensional, many correlated variables do exist in genetic data, and thus there is redundant information [4] which can be summarized utilizing dimensionality reduction techniques. However, when performed on a single dataset, dimensionality reduction approaches suffer from bias inconsistencies due to the specifics of each study such as laboratory procedures and conditions (batch effects) which in turn imply that results might not be reproducible [5–7].

1.2 Related Work

Taking advantages of all available resources and cover all biological conditions of gene expression by combining multiple datasets overcomes these issues. Integration of similar gene expression

datasets increases the sample size, thus increases the statistical power of the methods resulting in more precise, reproducible and robust findings. Hughey and co-authors [8] concatenated five datasets studying the same disease before building an elastic-net classifier. In other works [9, 10] nearly 9000 samples from the same microarray platform but different studies were fused before PCA was applied. Authors further performed cluster analysis and claimed that just the first few components had clear biological interpretation while the remaining contained irrelevant information. Crucially, *these studies do not consider how those constructed latent feature spaces behave on new unseen datasets.*

Combining information of data obtained from multiple technologies would be crucial for extracting the maximum biological information because they provide different partly and complementary aspects of the whole genome. Data integration over different microarray platforms has also been studied, mainly following two different directions, namely late and early stage integration. The late stage integration is a 'meta-analysis' application, where each platform is examined independently and then the results are combined. This approach is suitable for the purpose of biomarker discovery using statistical or regularization methods [11–13]. In the early stage integration datasets from different platforms are merged solely over the common genes [14–16]. Its main advantage over the late stage integration approach stems from its higher statistical relevance due to the large number of samples in the fused dataset that naturally leads to more powerful inference. However, discarding the non-common features leads to the loss of information since interdependencies among the genes are not taken into account. A possible option that overcomes this issue is to concatenate the datasets using their first principal components as it is proposed by Gregory and co-authors [17], where the authors concatenated matched tumor samples from different platforms to create a large dataset to increase their predictive power. This study is quite specific for the considered disease and does not give general conclusions about its PCA-based method used for gene expression data integration.

1.3 Contribution

In this study, *we build a universal, low-dimensional latent representation able to capture the biological information contained in the whole human genome.* To this aim, we performed an extended analysis and constructed several low dimensional feature spaces that aim to preserve the biological information and enhance machine learning algorithms for newly-seen datasets independently of their attributes such as sample size, sample categories etc. We collected hundreds of datasets with various cell types and diseases such as healthy tissues, cancer subtypes, AML etc. from three microarray platforms and from one Next Generation Sequencing RNA-Seq platform. We initially merged the datasets from each platform creating four large super-populations of gene expression datasets containing thousands of samples each. Then, we applied several dimensionality reduction methods such as PCA [18], Kernel PCA [19] and the state-of-the-art Autoencoder

Neural Networks [20] on these sets. To the best of our knowledge, this is the first time a neural network approach is applied for gene expression integration. Various compression magnitudes were tested and evaluated in terms of reconstruction error and predictive performance on newly-seen datasets. Classification models were trained with the Just Add Data Bio v0.57 (JAD Bio; Gnosis Data Analysis; www.gnosisda.gr), an evolution of the BioSignature Discoverer plug-in. JAD Bio employs a fully-automated machine learning pipeline for producing a classification model given a training dataset, and an estimate of its predictive performance in terms of area under the ROC curve (mean and confidence interval). The total number of classification performance that we obtained for this study was 4339 since we desired to do an extensive investigation of gene's latent structure.

We observed that the constructed latent representations are universal since the extracted low dimensional features on unseen datasets (i.e., datasets not used for training the dimensionality reduction methods) maintained and slightly improved the average predictive power for all four platforms when the dimension of the feature space is 200. Decreasing the dimensionality, or, equivalently, increasing the compression rate resulted in reducing the averaged accuracy of the classifiers. Hence, biological information do exist in higher dimensions contrary to previous studies [9] which reported global feature space with lower dimensionality. We also observed that there is no particular dimensionality reduction method that outperforms on every platform for both reconstruction error and prediction accuracy. PCA was usually the dominant method in terms of reconstruction error while nonlinear methods produced better classification outcomes since they were able to encode complex gene-gene interactions.

In order to provide a biological interpretation of the computed latent feature representations, we performed Gene Set Enrichment Analysis (GSEA) that determines when a pre-defined group of genes (pathway) is differentially expressed. We used the weights of PCA's projection vector as enrichment scores and observe that the first 20 vectors enrich almost all the KEGG pathways [21–23] despite the fact that more PCA projections are required for increased predictive power. This could be justified by the fact that higher components enrich some pathways that have not been enriched by the first PCs, showing that they capture some specialized pathways.

Finally, since we observed an increase of predictive power when more samples were considered for the construction of the latent space, we additionally investigated the merging of gene expression profiles from different platforms. We tested the fusion of the four platforms in several combinations by initially projecting them into their largest 500 Principal Components, which express 96% of the variance in average. Then, we performed additional dimensionality reduction. Cross-platform microarrays integration marginally improved the predictive performance, with the highest improvement occurred when GPL570 and GPL96 are merged indicating that a common cross-platform latent feature space is also feasible. Unfortunately, different combinations like microarrays with NGS had not the same performance with this manner of fusion showing that different mechanisms exist and different treatment is required.

1.4 Outline

The rest of the thesis is structured as follows. Chapter 2 introduces the appropriate background that is required to understand the technical details about this research study. However even without these basic knowledge, it is hoped that the idea and results of this research will be perceptible. Chapter 3 presents the datasets that were used, as well as their importance that prompted us to computational analyze them. Capter 4 gives a brief review of used dimensionality reduction methods and shows how we utilized them. Chapter 5 presents extensively the experiments and the results of this study. Provides details about how we perform the within and cross-platform integration of gene expression data, the construction of latent feature space and the evaluation of this latent space measuring statistical, reconstruction and prediction power. Finally in Chapter 6 we have an overview of the thesis and discuss the interpretation of the results.

Chapter 2

Background

In this Chapter we briefly review some basic machine learning notations and preliminaries that we will refer back to throughout this thesis so that is also readable for someone non-specialist in the field. In Section 2.1 we describe what is classification and how it works, it is a significant part of this thesis since we manage to increase the predictive power of gene expression data. A valid way to evaluate a classification task and the one used in this thesis is Area Under the Curve which is described in Section 2.2. A general idea about the scope of using dimensionality reduction techniques such those which are used in this study (Section 4), is described in Section 2.3, mentioning advantages and an intuitive point of view behind these methods. The basic objective of an Autoencoder (Section 4.3) is to minimize an objective function using optimization techniques. So we give an explanation about optimization theory in Section 2.4. Finally in Section 2.5 we proceed with the description of Hypothesis testing which is the main procedure of Enrichment Analysis (Section 5.3) and in the process of comparing the obtained results.

2.1 Classification

The task of classification occurs in a wide range of human activity. At its broadest, the term could cover any context in which some decision or forecast is made on the basis of currently available information, and a classification procedure is then some formal method for repeatedly making such judgments in new situations. We shall assume that the problem concerns the construction of a procedure that will be applied to a continuing sequence of cases, in which each new case must be assigned to one of a set of pre-defined classes on the basis of observed attributes or features. The construction of a classification procedure from a set of data for which the true classes are known has also been variously termed pattern recognition, discrimination, or supervised learning. X“n example on the data of this study, classification is the process of learning a function from labeled gene expression data by the observed characteristics (probes-genes) and then assigning a diagnosis (e.g. disease or not) on a new patient. The mathematical definition is the following, given a set of data samples with pairs $\{< x_i, y_i >: i = 1, ..., n\}$ where x_i is the

representation of an object usually is a vector and y_i the representation of a known outcome (a specific categorical label) of the object. The objective is to learn a function f (machine learning algorithm), using these data, that would be able to predict an outcome of interest y_i for the object x_i ($f(x_i) = y_i$) and can generalize on new unseen pairs $< x', y' >$ of the same problem. Fig 2.1 visualizes this process. For more details read "Machine learning, neural and statistical classification" [24].

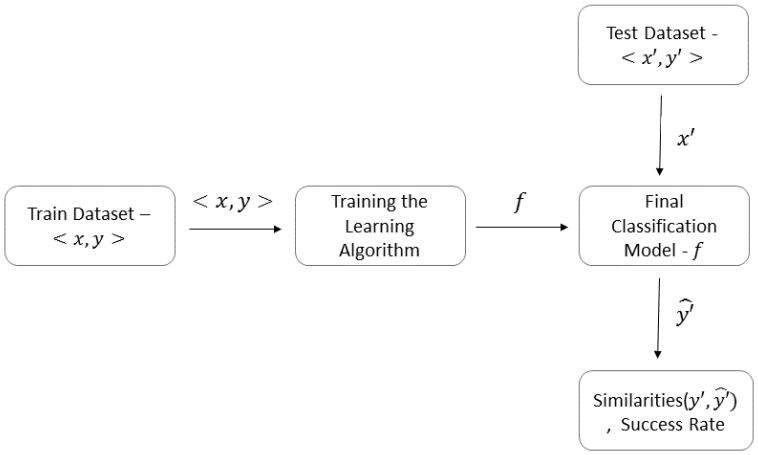

Figure 2.1: Classification process

.

2.2 Evaluation of Classification - Area Under the ROC Curve

When developing a classification system, its going to be indispensable to have an objective metric by which we can know how well it performs. Accuracy is often the starting point for analyzing the quality of a predictive model, as well as an obvious criterion for prediction. Accuracy measures the ratio of correct predictions to the total number of cases evaluated. It may seems obvious that the ratio of correct predictions to cases should be a key metric. However, a predictive model may have high accuracy, but be useless. Accuracy does not account distributions of each class, this can create the accuracy paradox which states that predictive models with a given level of accuracy may have greater predictive power than models with higher accuracy. It may be better to avoid the accuracy metric and use an other metric such is Area Under the Receiver Operating Characteristic Curve (AUC) [25]. AUC is explained in details in the below paragraph.

Lets consider the classification problems that have only two classes, binary classification problems. First we should refer some necessary terminology. For the ease of distinguishment two

classes are denoting as positive and negative. Given a classifier and an instance, there are four possible outcomes. If the instance is positive and it is classified as positive, it is counted as a *true positive*; if it is classified as negative, it is counted as a *false negative*. If the instance is negative and it is classified as negative, it is counted as a *true negative*; if it is classified as positive, it is counted as a *false positive*. True positive rate (TPR) or sensitivity (eq: 2.1), intuitively corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. In other words, the higher TPR, the fewer positive data points we will miss. False positive rate (FPR) (eq: 2.2), intuitively is a metric that corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. In other words, the higher FPR, the more negative data points we will missclassified.

$$TPR = \frac{true\ positives}{total\ positives} \tag{2.1}$$

$$FPR = \frac{false\ positives}{total\ negatives} \tag{2.2}$$

Using TPR and FPR we can create the Receiver Operating Carachteristic (ROC) graph which is useful technique for organizing classifiers and visualizing their performance. ROC graphs are two-dimensional graphs in which TPR is plotted on the Y axis and FPR is plotted on the X axis. A ROC graph depicts relative trade-offs between benefits (true positives) and costs (false positives). If the model is a discrete classifier that outputs only a class label,only a (FPR,TPR) pair is produced , which corresponds to a single point in ROC space. On the other if our classifier is a probabilistic model, ie it produces probabilities that represent the degree to which our samples are member of a class, we can construct the ROC and consequently find AUC. Ranking these probabilities can be used with a threshold to produce a discrete (binary) classifier: if the classifier output is above the threshold, the classifier produces a possitive, else a negative. Each threshold value produces a different point in ROC space. Conceptually, we may imagine varying a threshold from $+\infty$ to $-\infty$ and tracing a curve through ROC space. The AUC is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example, thus a random classifier will produce a ROC point that "slides" back and forth on the diagonal, so will have an $AUC = 0.5$. A classifier with $AUC < 0.5$ is not a realistic classifier and 0.5 is commonly used as a baseline to see whether the model is useful. A reliable and valid AUC estimate can be interpreted as the probability that the classifier will classify correctly a pair of samples with different classes.

Some of the data sets under consideration are multi-class, as we can see at the tables 1,2,3,4 and 5. The calculation of AUC on multi-class problems it is reduced to calculation of multiple AUCs. Each AUC is measured by the ROC of a class $i \in C$ as positive class and all the others as the negative. Then the general AUC is equal to the sum of the AUCs weighted by the reference

class's prevalence in the data $p(c_i)$ (eq: 2.3).

$$AUC_{total} = \sum_{c_i \in C} AUC(c_i) \cdot p(c_i) \tag{2.3}$$

2.3 Dimensionality Reduction

The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces. In machine learning problems that involve learning a "state-of-nature" from a finite number of data samples in a high-dimensional feature space with each feature having a number of possible values, an enormous amount of training data is required to ensure that there are several samples with each combination of values. In addition, analysis with a large number of variables generally requires a large amount of memory and computational power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new sample. Hughes phenomenon describes exactly the curse of dimensionality, which says that for a given sample size, there is a maximum number of features above which the performance of our classifier will degrade rather than improve [26].

Dimensionality reduction is the process of reducing the number of existing variables under consideration, via obtaining a set of principal variables. It can be divided into feature selection which try to find a subset of the original variables and feature extraction which transform the variables to a space of fewer dimensions. Feature selection is above the purpose of this work so it is not covered in depth. An interesting paper about feature selection is "Forward-Backward Selection with Early Dropping" [27].

The problem of feature extraction can be stated as follows. Given a feature space $x_i \in \Re^D$ fid a mapping $z = f(x) : \Re^D \rightarrow \Re^d$ with $d < D$ such that transformed feature vector $z_i \in \Re^d$ preserves the information or structure in $\Re^D$. The selection of the feature extraction mapping $z = f(x)$ is guided by an objective function that we seek to minimize (or maximize), more details in Section 2.4. Depending on the objective function, the goal of the feature extraction mapping is either to represent the samples accurately in a lower space (trying to minimize Reconstruction Error) or to enhance the class-discriminatory information in the lower-dimensional space. After extracting dimensionality reduced features, these are going to be the examining representation of the object mentioned in Section 2.1.

Concluding the advantages of dimensionality reduction are that it reduces the time and storage space required, removal of multi-collinearity which improves the performance of the machine learning model and it becomes easier to visualize the data when reduced to very low dimensions such as 2D or 3D.

2.4 Mathematical Optimization

The process of finding the maximum or minimum of a function with some constraints is referred as optimization. Optimization is the basic learning process for neural networks (Section 4.3), one of the three dimensionality reduction techniques used in this study.
A mathematical optimization problem has the form:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant b_i,\ i = 1, \ldots, m. \end{aligned} \tag{2.4}$$

Here the vector $x = (x_1, \ldots, x_n)$ is the optimization variable of the problem, the function $f_0 : \Re^n \to \Re$ is the objective function, the functions $f_i : \Re^n \to \Re$, $i = 1, \ldots, n$ are the (inequality) constraint functions and the constant $b_1, \ldots, b_m$ are the limits for the constraints. A vector x^* is called optimal or a solution of the problem 2.4 if it has the smallest objective value among all vectors that satisfy the constraints, for any z with $f_1(z) \leqslant b_1, \ldots, f_m(z) \leqslant b_m$, we have $f_0(z) \geqslant f_0(x^*)$

A task in machine learning where we use optimization theory is to find a model, from a family of potential models, that best fits some observed data and prior information. Here the variables are the parameters in the model, and the constraints can represent prior information or required limits on the parameters (such as non negativity). The objective function might be a measure of misfit or prediction error between the observed data and the values predicted by the model, or a statistical measure of the unlikeliness or implausibility of the parameter values. The optimization problem 2.4 is to find the model parameter values that are consistent with the prior information, and give the smallest misfit or prediction error with the observed data. Note that for many problems, more than one optimum (referred to as local optimum) may exist.

Most common techniques of finding a local optimum are gradient-based, which as indicated by the name, make use of gradient information to find the optimum solution of equation 2.4. The general process is described in Algorithm 1, where $t \geqslant 0$ is learning rate. The interest reader is referred to [28] for a rough description of the field.

Algorithm 1 Gradient descent method

Input: a starting point $x \in dom(f)$
1: **repeat**
2: *update:* $x := x - t \cdot \nabla f(x)$
3: **until** stopping criterion is satisfied (e.g. $\|\nabla f(x)\|_2 \leqslant \eta$ where η is small and positive)

2.5 Hypothesis Testing

A statistical hypothesis is an assumption about a population parameter. This assumption may or may not be true. Hypothesis testing refers to the formal procedures used by statisticians

to accept or reject statistical hypotheses. The best way to determine whether a statistical hypothesis (certain condition) is true would be to examine the entire population. Since that is often impractical, researchers examine a sample of data to infer what is true for the entire population. Particularly, a hypothesis test examines two opposing hypothesis about a population: the null hypothesis denoting as H_0 which is the statement being tested and is the contrary of what we want to coclude ie that our observations results purerly from change. And the alternative (H_1) hypothesis is the statement we want to be able to conclude is true. To determine if the Null hypothesis will be rejected or not a test statistic T needs to be defined in such a way as to quantify, within observed data, behaviours that would distinguish the null from the alternative hypothesis. The next step is to find the value t_0 of the chosen test statistic T on the sample data. Then it is decided if H_0 is true or not using a predefined decision rule. A decision rule could be a critical region in the distribution of our test stastic (usually knonwn), if the observed value t_0 is in the critical region reject H_0 otherwise "fail to reject" the null hypothesis. Algorithm 2 details the method.

Algorithm 2 Hypothesis testing process

1: Decide what you what to "prove" and state it as Null and Alternative Hypothesis
2: Find a suitable test statistic T and consider the statistical assumptions being made about the sample in doing the test; for example, assumptions about the statistical independence or about the form of the distributions of the observations.
3: Derive the distribution of T under the null hypothesis from the assumptions
4: Select a significance level (α), a probability threshold below which the null hypothesis will be rejected (e.g. 0.05,0.01).
5: Find the rejection region using α and the distribution of the test statistic T
6: Compute from the observations the observed value t_0 of the test statistic T
7: Decide to either reject the null hypothesis if the observed value t_0 is in the critical region,in favor of the alternative or otherwise not reject it.

A intuitive example from [29], suppose we wanted to determine whether a coin was fair and balanced. A null hypothesis might be that half the flips would result in Heads and half, in Tails. The alternative hypothesis might be that the number of Heads and Tails would be very different. Symbolically, these hypotheses would be expressed as

$$H_0 : P = 0.5$$
$$H_1 : P \neq 0.5$$

Suppose we flipped the coin 50 times, resulting in 40 Heads and 10 Tails. Given this result, we would be inclined to reject the null hypothesis. We would conclude, based on the evidence, that the coin was probably not fair and balanced.

Chapter 3

Data sets

A gene is a specific base sequence of DNA that encodes function. Gene expression is the process by which information from a gene is used in the synthesis of a functional gene products. These products are often proteins, but in non-protein coding genes such as rRNA genes or tRNA genes, the product is a structural or housekeeping RNA. The data that are being used in our analysis are Microarray and Next Generation Sequencing. Typically to obtain microarray data [2, 30] biologists extract mRNA from samples as experimental samples and control samples then make labeled cDNA through reverse transcription, mix samples and hybridize to cDNA microarray after mixing, the cDNA is placed on a microarray slide and left to hybridize and in the end the microarray is placed in the scanner and passes 2 times the slide and reads the intensity emitted by each fluor and generates 2 different 16-bit gray scale images. A false coloring is a applied in these two images, one red and one green, based on a temperature scale then the combination of these two colorized images yields a graphical representation of different gene expression between the two samples. RNA sequencing (RNA-seq) is an alternative technique to measure gene expression [3]. High-trhoghput Sequencing applies to genome sequencing, genome resequencing, transcriptome profiling (RNA-seq). The main difference of microarray data with RNASeq data is that in the later we are dealing with counts instead of just intensities.

We collected gene expression studies from four different platforms. From Gene Expression Omnibus database [31], which is an international repository with gene expression and other genomics data we gathered microarray datasets with sample size greater than 20. More specifically, we obtained datasets from Affymetrix Human Genome U133 Plus 2.0-GPL570 (subsection 3.0.1), Affymetrix Human Genome U133-GPL96 (subsection3.0.2) and "Mus musculus" datasets from Affymetrix Mouse Genome 430 2.0-GPL1261 (subsection 3.0.3). RNA sequencing also called Next Generation Sequencing (NGS) is an alternative technique to measure gene expression, thus we collected also datasets from ReCount database [32] (subsection 3.0.4) . In order to evaluate the results, using text mining and manual curation, we managed to label the samples of 10% of the studies for each platform. Labels correspond to information regarding disease states, cancer subtypes, smoking status etc..

3.0.1 Affymetrix Human Genome U133 Plus 2.0-GPL570

This platform's datasets are homo-sapiens microarray gene expression data. GPL570 measures 54675 features-probes. In these measurements, there are genes that are being referred from more than one probe and probes that not refer to any gene. We collected a total number of 199 different datasets. We labeled 10% (20 sets) of them which are being held out as Test-set. Sets that belong to the Test-set have different types and number of classes as we can see in figure 1 and with extensive details on table 1 in Appendices section. We concatenated the remaining 179 data sets. Then randomly split to 90% of Train-set and 10% of Validation-set, 6795 samples and 756 samples respectively. The Train-set is being used to train our methods and Validation-set for an initial estimate of how the process of method training goes.

In addition, in order to prove that our findings are general and do not altered as the available data increases, we gathered a larger number of studies of GPL570 (899 studies). We labeled 80 out of 899 studies that were used for evaluaton (Test-sets, more detailes in table 5) and the rest after removing dubllicates assembled a dataset of 59864 samples which was used as Train-set.

3.0.2 Affymetrix Human Genome U133-GPL96

GPL96 also measures homo-sapiens microarrays. The total number of features that being measured is 22834. There is a peculiarity with the measurements of GPL96, features-probes represented on the GPL96 are subset of probes that are examined on the GPL570. The number of data sets that are being used from this platform is 86 where 10% of them i.e. 9 are being kept for Test-set. As before Test-set contains different sizes of data from 20 samples to 100 and different number of classes, more details we can see in table 2. The other 77 are being separated to Train (3331 samples) and Validation (371) set.

3.0.3 Affymetrix Mouse Genome 430 2.0-GPL1261

Data sets of this platform are microrrays from an organism named "Mus musculus", a small mammal which has been domesticated as the pet or fancy mouse, and as the laboratory mouse, which is one of the most important model organisms in biology and medicine. We obtained 200 data sets, 20 of them are being kept as Test-set, table 3. The other 180 are being blended and then split to Train-set (7282 samples x 45101 features) and Validation-set(810 samples x 45101 features).

3.0.4 Next Generation Sequencing-NGS

Finaly we took in account 175 data sets measured by RNA-seq technique with number of samples greater than 40 and a feature space sized 23779. Again 10% of them ie. 17 labeled data set (table 4) were kept for Test-set and the rest 157 were merged to a set (Train-set) of 21609 samples and 23779 features.

Chapter 4

Methods and Materials

In this Chapter we describe the dimensionality reduction methods used for the porpuse of this analysis. First in Section 4.1, we explain how linear PCA works and its relation to Singular Value Decomposition. Then Section 4.2 refers to Kernel PCA which is the nonlinear aspect of PCA. Section 4.3 presents a more complex technique, the Deep Learning approach of dimensionality reduction the Autoencoder Neural Network. Finally, in Section 4.4 we briefly describe the automated machine learning tool (JAD bio) which is used to measure the predictive performance of datasets of this study.

4.1 Principal Component Analysis

The most common technique for dimensionality reduction is Principal Component Analysis [18]. PCA uses an orthogonal transformation to convert a set of possibly correlated features into a smaller set of linearly uncorrelated variables. This orthogonal linear transformation transforms our data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on first coordinate called first principal component and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. Let X be our data matrix with m zero mean columns and n rows, columns represent features of data and rows the samples. In mathematical terms, PCA attempts to find a linear mapping M that maximizes the cost function $trace(M^{\intercal}cov(X)M)$, where $cov(X)$ is the sample covariance matrix of the data X. It turns out that this is done by finding the eigenvalues and eigenvectors of the sample covariance matrix. Therefore, the eigevector that corresponds to the i-th eigenvalue λ_i of the covariance matrix is the i-th principal direction. Consequently, the i-th principal component is the projection of X into the i-th principal direction.

Feature dimensions of gene expression datasets are several thousands, so the construction and the manipulation of the covariance matrix is unprofitable since it causes time issues. Additionally, the number of sample size in our data is usually extremely lower than the number

of features making the procedure of finding every eigenvectors unnecessary since the most of them are nearly equal to the zero vector. So, we used another approach named Singular Value Decomposition (SVD) which is a matrix factorization and gives an equivalent solution [33]. SVD factorizes X as $X = U\Sigma V^T$. Σ is an n-by-m rectangular diagonal matrix. The diagonal values of Σ are the positive numbers σ_k called singular values of X, and $\sigma_k = \sqrt{\lambda_k}$ where λ_k is the k-th eigenvalue of covariance matrix of X. U and V called the left and the right singular vectors of X, orthogonal matrices. The right singular vectors V are equal with eigenvectors of the covariance matrix. Therefore, the right singular vectors V are the principal directions-weights (PCWs).

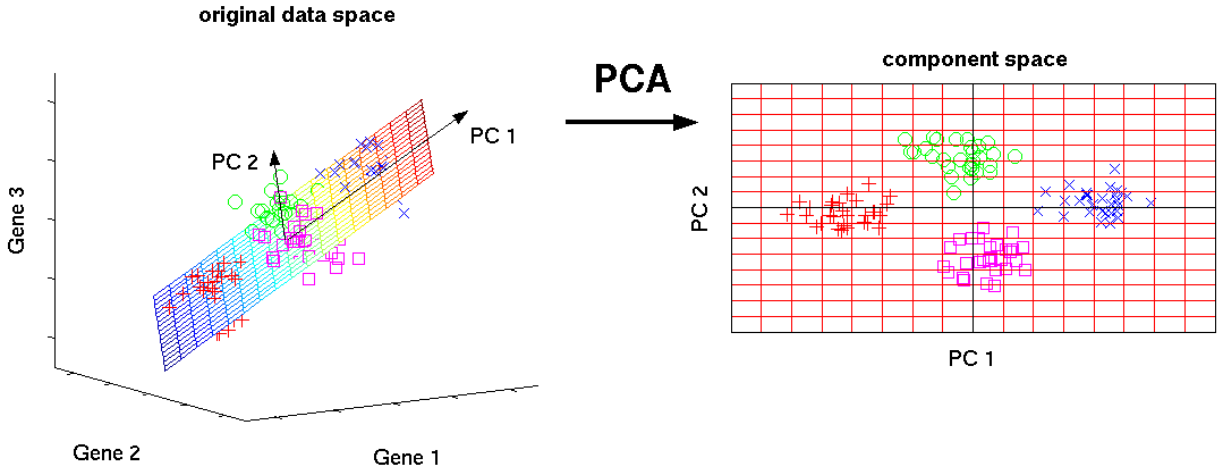

Figure 4.1: The image shows the transformation of a high dimensional data (3 dimension) to low dimensional data (2 dimension) using PCA. Not to forget, each resultant dimension is a linear combination of d features (Credits: www.analyticsvidhya.com,Practical Guide to Principal Component Analysis)

4.2 Kernel-Principal Component Analysis

Kernel PCA [19] generalizes standard PCA to nonlinear dimensionality reduction. The naive way to perform PCA nonlinearly, is to initially use a nonlinear transformation function $\phi(x)$ from the original dimensional feature space to a latent feature space and then perform PCA. This transformation function $\phi(x)$ could be very-high-dimensional making the projection extremely costly and inefficient. The explicit calculation of the new feature space can be avoided using the Kernel trick. The Kernel trick refers to creating a Kernel function (similarity function) which is used over pairs of data points in raw representation $K(x, x') = \phi(x)^\intercal \phi(x')$. The Kernel function is a matrix $N \times N$ which it's eigenvectors V are equivalent to the principal component weights in the latent space created by $\phi(x)$. There exist coefficients $\alpha_1, \ldots, \alpha_N$ such that $V = \sum_{i=1}^{N} \alpha_i \phi(x_i)$,

consequently the principal components in the nonlinear space are equal with

$$y = V\phi(x) = \sum_{i=1}^{N} \alpha_i \phi(x_i)\phi(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x) \tag{4.1}$$

. If the projected dataset $\phi(xi)$ does not have zero mean, we can use the Gram matrix $\widetilde{K}$ to substitute the kernel matrix K. The Gram matrix is given by

$$\widetilde{K} = K - 1_N K - K 1_N + 1_N K 1_N \tag{4.2}$$

where $1_N$is the $N \times N$ matrix with all elements equal to $1/N$ [34]. The power of kernel methods is that we do not have to compute $\phi(x_i)$ explicitly. The most widespread kernels are Polynomial and Gaussian. We apply Polynomial Kernels of degree 2 and 3 and Gaussian Kernel with gamma parameter equal with $\frac{1}{\#features}$.

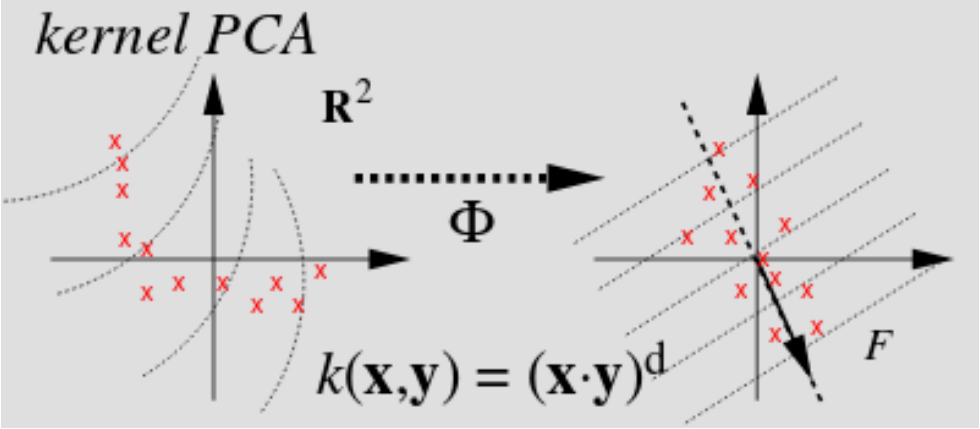

Figure 4.2: Kernel PCA is implicitly performing a linear PCA in some high dimensional feature space, that is nonlinearly related to input space. (Scholkopf et al., International Conference on Artificial Neural Networks (1997))

4.3 Auto-Encoder

Hinton and Salaktutdinov [20] proposed Deep Learning Autoencoders in order to convert high-dimensional images to low-dimensional codes with compact information. Neural Networks require large datasets and computational power due to their complexity. Only recently, Neural Networks have been used in biological research due to the limited number of samples. We alleviate this issue by dataset integration which resulted in tens of thousands of gene expression samples.

Autoencoder is Neural Network that has two parts: the encoder,f, that maps input x into a nonlinear represantation $h = f(x)$ and the decoder,g, that maps h back to the original space. The way Autoencoder is trained is by finding the appropriate weights w which are the coefficients of

the functions. Appropriate weights w are the ones that minimize an error function that compares the output of the Network with its input. A trivial Autoencoder whith one hidden layer trained using mean square error and a linear function f is equivalent to computing the first principal components of the data. When the hidden layer is nonlinear, the Autoencoder behaves differently from PCA. Neural Networks are able to capture multi-modal aspects of the input distribution. To capture even more complex information of data, more hidden layers are used allowing them to compactly represent highly nonlinear and highly-varying functions.

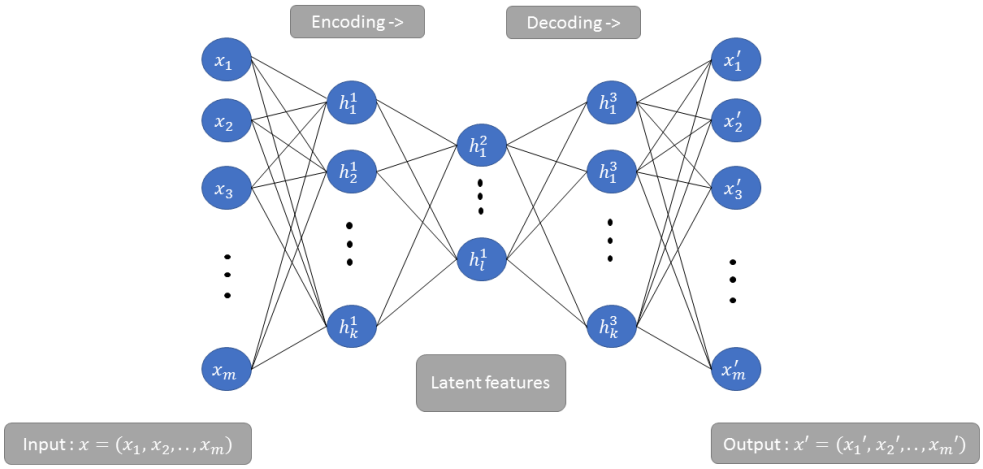

Figure 4.3: The structure of an Auto-Encoder. As input is every sample of data and the goal is to learn a representation (encoding) and then reconstruct the input (decoding) as better as possible. Find weights for each connection that minimize the reconstruction error $\min_w D(x, f_w(x))$

.

We trained the Networks using a greedy layer-wise unsupervised learning algorithm with Restricted Boltzmann Machines [35]. Restricted Boltzmann Machines or RBMs [36] are a one layer Autoencoder where all units of visible layer are connected to all hidden units, trying to learn nonlinear features from input data with the ability to reconstruct this input with effectiveness. So what greedy layer-wise unsupervised algorithm does is to pre-train each layer separately for some iteration-epochs starting from the first layer of the Autoencoder, i.e. the first RBM has as input and hidden layer the input and the first hidden layer of Autoencoder respectively, the second RBM's input is the first hidden layer and as hidden layer uses the second layer of Autoencoder and so forth. When pre-training completed a fine-tuning all over the network is being performed.

Learning of Neural Networks is nontrivial. There are several parameters that should be tuned, the most significant is the selection of the network's structure. To decide the structure,

we tried a series of experiments with different number of hidden layers and number of units. In our experiments we use a 3 hidden layer Autoencoder. The exact structures that are used are declared in table 4.1. As transformation functions we used encoder function to be the sigmoid function (equation 4.3) and the decoder function also the sigmoid with transposed weights. Also learning rate, the number of iterations and the number of batches are some other hyper-parameters that should be pre-defined. We performed pre-training for 600 epochs(iterations) where we used a minibatch size 100 and a learning rate 0.01. After pre-training, we performed a tunning on the whole structure for 3000 epochs with a vanilla gradient descent. Used learning rates were 0.1 on first 1500 epochs, 0.03 for epochs 1501-2300, 0.01 for 2301-2800 and 0.003 for the the last epochs. Finally, the cross-entropy (equation 4.4) function was used as the objective function to be minimized because of its appropriateness on networks with sigmoid function as transfer function [37]

Input layer	Hidden layer 1	Hidden layer 2	Hidden layer 3
500	700	500	200
500	700	200	50
500	700	200	20
500	700	200	10
500	700	200	5
500	700	200	2

Table 4.1: Autoencoder's structures

$$\sigma(x) = \frac{1}{1 + e^{-wX-b}} \tag{4.3}$$

X is the input, W the weights of the edges and a bias vector b

$$C = -\frac{1}{n}\sum_{n=1}^{N} y_n \log(y'_n) + (1 - y_n)\log(1 - y'_n) \tag{4.4}$$

N is the total number of samples of training data, the sum is over all training inputs, y_n is the corresponding desired output in auto-encoder's case is equal x_n, and y'_n is the output of the network.

4.4 JAD Bio

For our computational experiments, we used the Just Add Data tool (JAD Bio; Gnosis Data Analysis; www.gnosisda.com). Just Add Data is an automated tool that produces a supervised machine learning model and an estimate of its predictive performance. For classification problems (i.e., when the outcome is an integer value), JAD employs state-of-the-art machine learning

algorithms, such as random forest (RF) [38], support vector (SVM) [39] using both polynomial and Gaussian kernels and linear although the list is continuously being enriched. A high-level overview of the pipeline used by JAD is shown in figure 4.4. All those algorithms require the user to set a number of parameters (called hyperparameters in this context) that determine their behavior, and whose optimal values are problem-dependent. Results can greatly vary depending on correctly tuning the values of the hyper-parameters. The hyper-parameters are depicted as sliders in figure 4.4. Their optimal values cannot be found analytically; their values must be found by trial-and-error. JAD uses the statistical properties of the input data (such as the number of training examples and number of features) to determine a set of hyper-parameter combinations (called configuration hereafter) to try. In order to find the best algorithm and hyper-parameter configuration and to learn a final model, JAD uses the K-fold cross-validation protocol, described next. The K-fold cross-validation protocol splits the data into K non-overlapping approximately equal-sized sets (called folds). Each of them is held-out for testing purposes and the rest are used for training. It proceeds by keeping each fold out once, training models using all configurations on the remaining K-1 folds, and estimating their performance on the held-out fold. The held-out test sets are used to simulate the application of the models on new data. In the end, K performance estimates are computed and average of them is equal with the predictive performance of the dataset.

JAD has also been recently successfully applied to the prediction of proteins to periplasmic or cytoplasmic given their mature amino acid sequence [40]. The application shows the ability of the automated pipeline to learn patterns from data that generalize to new data. JAD employs a fully-automated machine learning pipeline for producing a model from a dataset and an estimate of its predictive performance on new. The latter is especially important, as the main goal is to create a model that is able to perform well on new data, rather than the data used for producing it.

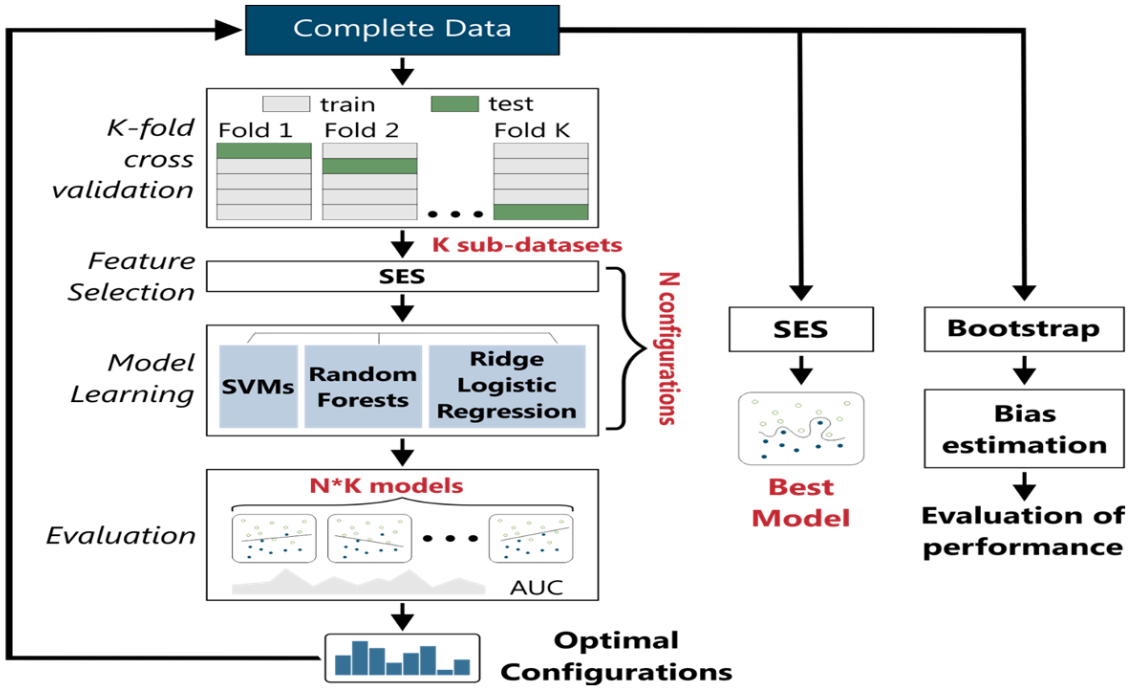

Figure 4.4: Schematic representation of the analysis pipeline employed by JAD. Based on the type of data and its size, the tool determines a set of combinations of tuning hyper-parameter values to try, called configurations. Hyper-parameters are depicted as tuning sliders. The data are partitioned to K folds and for each fold and configuration a predictive model is trained. These are evaluated on the held-out folds and the average performance of each configuration is estimated. Based on the best configuration found a final model is produced on all data. In our study we did not utilize the feature selection part.

Chapter 5

Experiments and Evaluation

5.1 Within platform integration

For each platform, we merged randomly 90% of the collected studies and generated an integrated dataset with thousands of samples which served as the training set to the dimensionality reduction algorithms. Figs. 5.1(a.1) & 5.1(b) depict the integration and the latent feature construction processes, respectively. PCA as well as kernel PCA were applied on the original dimension, however, training a deep Autoencoder was impractical on the raw data, due to the fact that the number of Autoencoder's parameters were too large making the training infeasible given the relatively limited number of training samples. In order to overcome this issue, we performed an initial dimensionality reduction using PCA and kept the 500 largest Principal Components (PCs) of each set. Note that 500 PCs explained in average 96% of the relative variability (i.e., one minus the ratio between reconstruction error and squared Euclidean norm). Hence, without loss of any significant information, these PCs were used as input to the Autoencoder for further dimensionality reduction (see also Fig. 5.1(b)). The evaluation of the dimensionality reduction performance is measured with the reconstruction error on newly-seen studies (the 10% of datasets that were kept out) defined as the mean squared error between the original and the reconstructed datasets. We additionally employed JAD Bio which is an automated Machine Learning tool to measure the prediction performance as shown in Fig. 5.1(c). As an evaluation metric for classification performance we reported the Area Under the ROC Curve (AUC). AUC is a reliable metric since it is invariant of the sample size of each category. Finally, the AUC on the raw datasets was also computed and used as a reference point. For a statistical robust comparison between the AUCs of the reference and the latent features AUCs, we performed a t-test which is a statistical hypothesis which determines if two sets of data are significantly different from each other.

Fig. 5.2 presents the reconstruction error (upper row of panels) as well as the AUC (lower row of panels) on newly-seen datasets. We chose to compute the performance metrics at 200, 50, 20, 10, 5 and 2 latent feature space dimensions which constitute a wide range of values. As expected, the reconstruction error is increased as the dimension of the latent space is decreased.

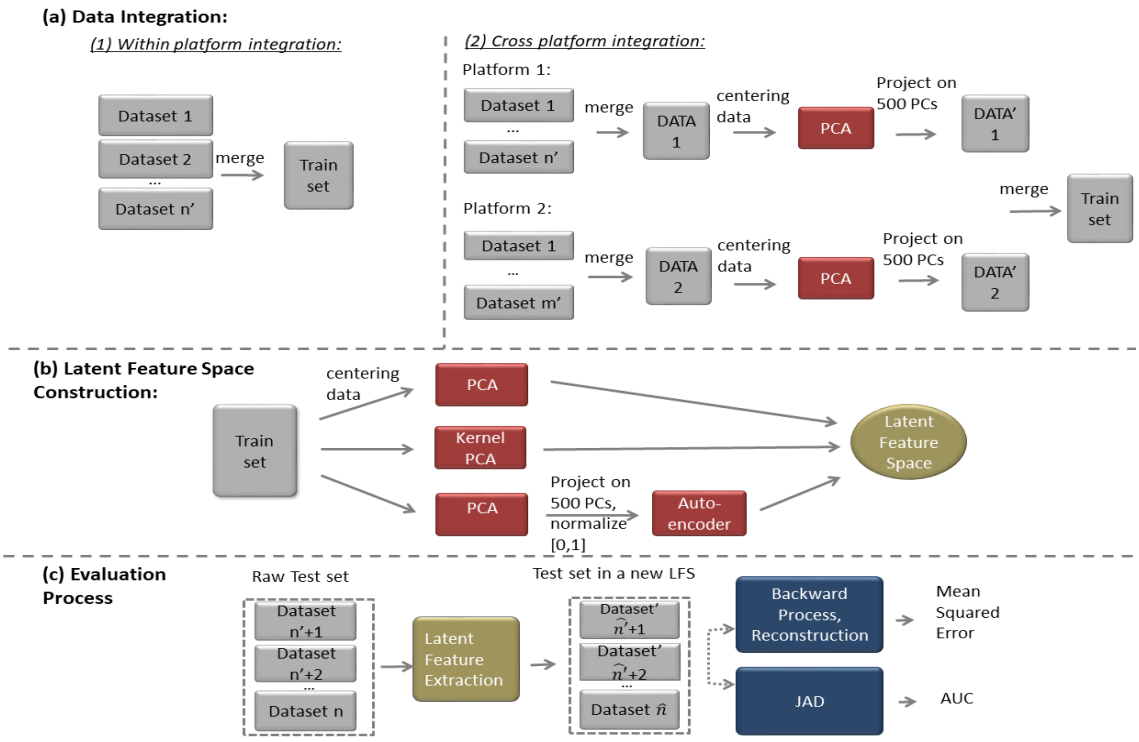

Figure 5.1: Outline of our integration analysis approach: (a) Data Integration. We merged 90% (i.e., $n' = ceil(0.9n)$) of the collected studies and denoted them as the Train set. Two cases were considered; (a.1) the "within platform integration" case where we straightforwardly concatenated the datasets and (a.2) the "cross platform integration" case where we initially performed dimensionality reduction using PCA keeping the first 500 PCs (which explain in average 96% of the relative variance) and then we concatenated the projected samples of each platform. (b) Latent Feature Space Construction. After fusion, we applied several dimensionality reduction methods with various values for the latent space dimension. For the Autoencoder's approach in "within platform integration" case, we first projected to the first 500 PCs because the number of parameters became very large making the neural network training impractical. (c) Evaluation Process. We projected the remaining 10% of the studies (i.e., the Test set) onto the constructed latent feature space. Then, we evaluated the quality of the dimensionality reduction algorithms in terms of both the reconstruction error in the original space and the classification performance using JAD Bio.

We also observe that nonlinear dimensionality reduction methods perform slightly worse than PCA (blue line) in terms of reconstruction error in almost all cases. More specifically, using 200 dimensions, in average for each platform PCA explains 87% of the relative variance compared to 81% of Autoencoder (green line) which is the second best method. Moreover, it is evident that Autoencoder produced similar reconstruction error with PCA when it was trained with NGS dataset Fig. 5.2(d) which is the platform with the largest number of training samples. Regarding the significant high reconstruction error using the Gaussian kernel PCA in NGS is due to inappropriate for this case hyperparameter gamma (gamma parameter equal with $\frac{1}{\#features}$).

In contrast, the predictive performance as measured by averaged AUC was higher for the Autoencoder method than PCA particularly using human microarray datasets. On the other

Kernel PCA methods had a consistent modest performance on all platforms. For GPL570 (Fig. 5.2(e)), we were able to slightly improve the performance by 1% when compared with the reference AUC (pink line) for PCA (blue line), Autoencoder (green line) when the latent feature dimension is set to 200. For GPL96 (Fig. 5.2(f)), Autoencoder and PCA with gaussian and 2-polynomial kernels improved the classification accuracy compared to the reference, again, for 200 dimension. Irrespectively of the method, we obtained equal or slightly higher results than raw data when the latent feature space dimension is 200. Interestingly, Autoencoder, PCA and 2-kernel PCA for GLP96 got better or equal classification performance than the reference even for 20 dimensions showing that the gene expression data might be represented with only 20 features. The other two platforms GPL1261 (Fig. 5.2(g)) and NGS (Fig. 5.2(h)) showed similar behavior to GPL570. Looking also the t-tests results (details in tables 7,8,9 and 10 at Appendices), when we reduce the dimensions to 200, independent the platform there is no statistically significant difference from the reference's results, since almost all p-values are larger than 0.05. Indicating that we obtain similar results with the results from the initial high dimensional space. In addition, GPL96 and GPL1261 show not statistically important difference even in a latent space sized 20. For example using pca and 20 dimensional space, we obtain p-values 0.99 and 0.18 for GPL96 and GPL1261 respectively.

The overall conclusion is that indeed gene expression data are redundant with two to three orders of magnitude lower intrinsic dimensionality. A fact that demonstrates that gene expression can even be represented with few latent features. Nevertheless, in order to preserve the predictive performance, gene expression data can be reduced to a latent feature space with approximately 200 dimensions. This value is larger than the reported in previous studies indicating that there is crucial biological information in higher dimensions that boosts the machine learning algorithms to achieve better predictive results.

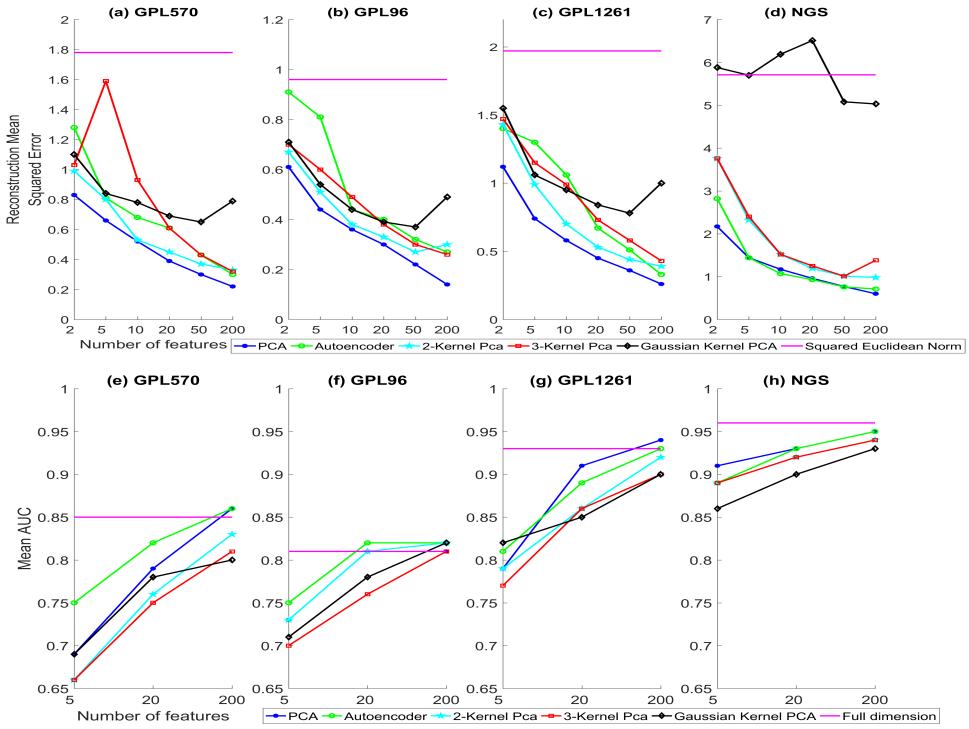

Figure 5.2: Performance assessment of dimensionality reduction techniques on new datasets for each platform in "within platform integration", regarding the reconstruction error (first row) and classification AUC (second row). It is evident that the fewer the number of features the larger the reconstruction error. The reference point (pink line) is the squared Euclidean norm of the sets and practically it is the variance since sets are centered. PCA (blue line) is the dominant method in terms of the reconstruction performance. Regarding classification performance, it is verified that a strong reduction, below 200 dimension, results in a loss of accuracy. Nevertheless, in a latent feature space sized 200 the prediction performance is equal or better compared to the raw data performance for all platforms.

5.2 Large scale within platform integration

As gene expression data become more available, we would like to verify that the results presented above are robust as we increase the number of training datasets and generalize on any newly-produced dataset provided by the biologists. In addition, we would like to investigate if higher the sample size the higher the performance. For these purpose, we gathered 899 studies from GPL570 assembling a large dataset of 59864 unique samples. We merged 90% of the datasets and then performed dimensionality reduction as in Section 5.1 while the rest 80 labeled datasets are used for performance evaluation. We chose as latent feature size the values $500, 200, 50, 20$. Fig. 5.3 presents both the mean reconstruction error (left column) and the mean prediction performance (right column). In terms of reconstruction error, we observe that PCA (blue line) outperforms kernel PCA (cyan & red lines). Interestingly, PCA is less accurate than Autoencoder (magenta line) which has lower reconstruction error revealing again that Autoencoder can be a highly competitive method when enough samples are available. For instance, when the dimension of the latent feature space is 20, 74% of the relative variance is kept using Autoencoder which is 10% higher than PCA's relative variance. Autoencoder is also the leading method in the classification task and achieves higher AUC when compared to the PCA-based methods. As in Section 5.1, the larger the latent feature space the better the classification accuracy as it is evident from Fig. 5.3. The 2-polynomial kernel PCA and linear PCA managed an increase of mean AUC by 3% with 500 latent dimension, which was also achieved by Autoencoder in only 200 dimension. In all cases where we reach better mean AUC from the reference, we observe very low p-values (e.g. p-value of $Autoencoder200 = 0.004$) as shown in table 11. Demonstrating that the improvement is statistical significant. Also in 20 dimension using Autoencoder the p-value is larger than 0.05, which means that the results on the reduced latent feature space (mean AUC 0.84) are not statistically different with the reference (mean AUC 0.85). We also report in Fig. 5.4 the classification accuracy for each individual test dataset. Using Autoencoder with a representation of 200 dimension, 12 datasets have improved AUC by at least 10% while only 1 dataset's performance deteriorated by the same percentage compared to the raw dataset's performance.

The larger sample size gave the opportunity for higher dimensionality reduction. Evidently, we were able to get comparable results with the reference AUC when Autoencoder was applied for the construction of a 50 and even 20 dimensional latent feature representation. Overall, the increase of training sample size instead of negatively affecting the robustness of the dimensionality reduction methods, it actually yields improved classification accuracy to newly-seen datasets and allows further reduction of the dimension of latent spaces implying that the nonlinear interactions between genes and/or experimental conditions can be captured by applying more sophisticated reduction methods given the availability of a large number of samples. These facts encourage us to gather more datasets and create even larger integrated sets.

In order to further substantiate the statement that larger integrated dataset results in higher

classification performance, we trained Autoencoders using variable number of sample sizes. We tested 5, 10 and 30 thousands of training samples and showed (Fig. 5.5) that the predictive performance is increased with the size of the training set. Showing the importance of learning from a large data gene expression dataset which includes a variety of sets from various diseases and different laboratories in order to remove bias and retain only important biological information.

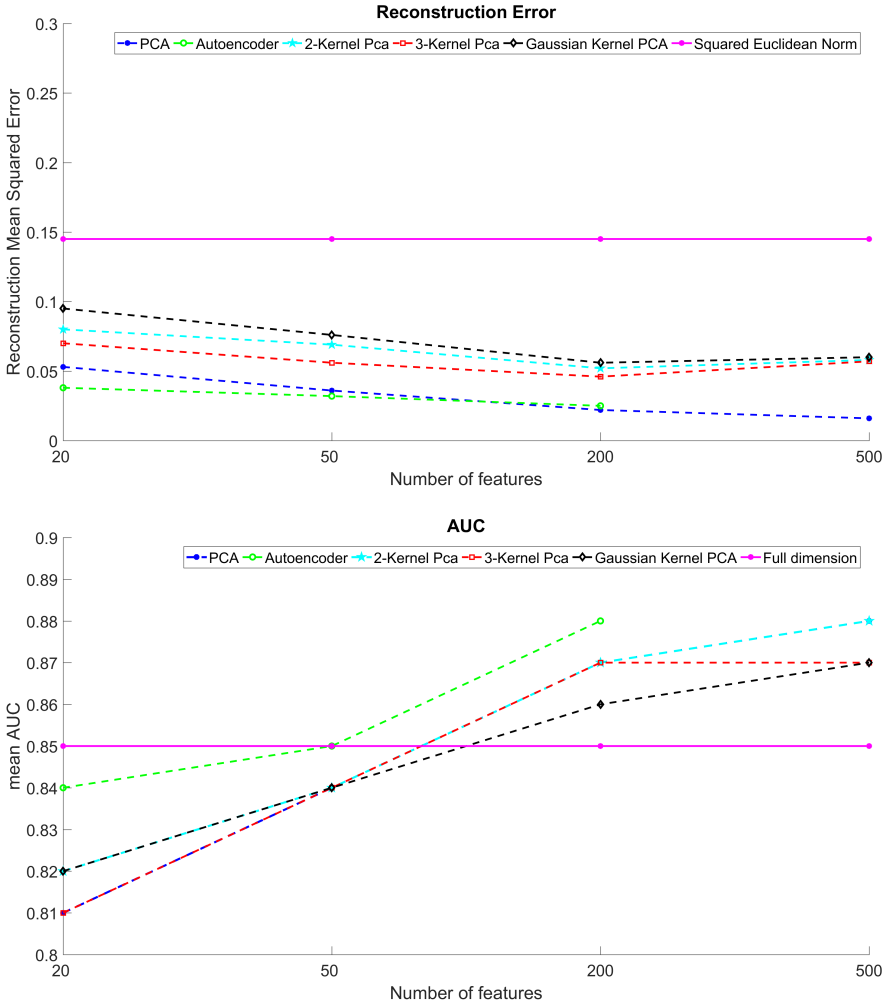

Figure 5.3: Mean reconstruction error (first row) and classification accuracy in terms of AUC (second row) for 80 newly-seen datasets. Standard PCA (blue line) and Autoencoder (green line) have comparable reconstruction error. However, Autoencoder is superior in classification performance compared to all the other methods with high mean AUC even in lower dimensions.

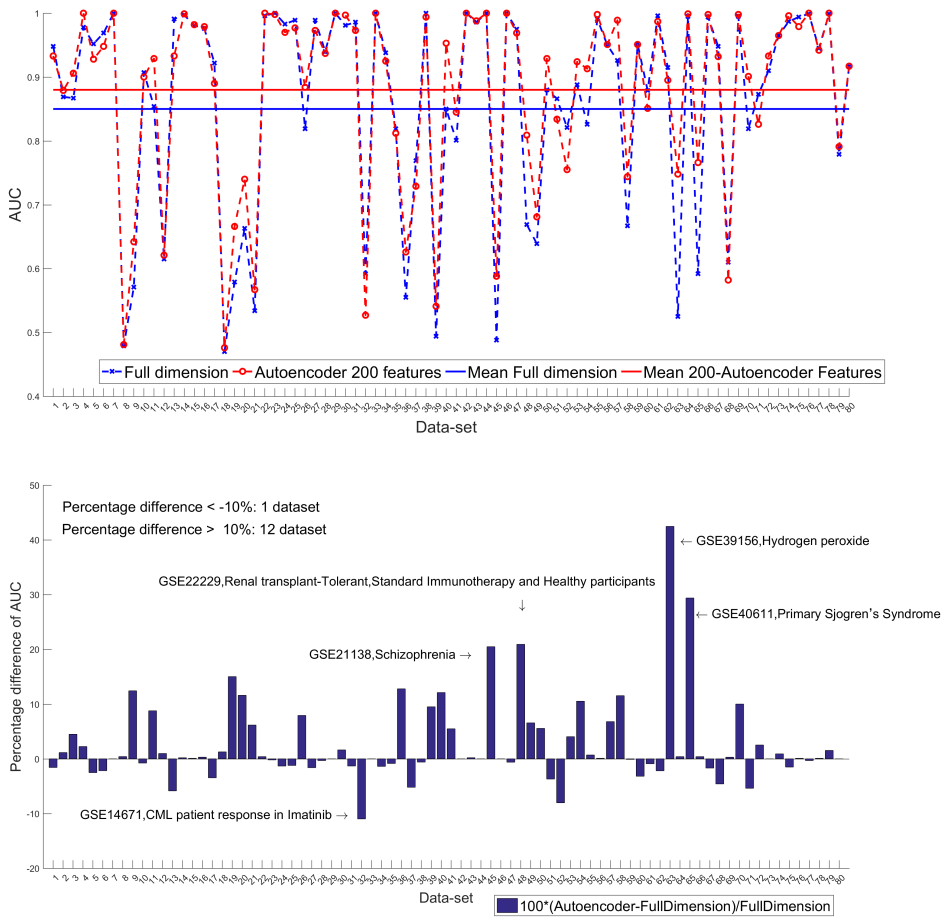

Figure 5.4: Predictive performance comparison of Autoencoder 200 feautes with Full dimension data. First row depicts the AUC of each of the new 80 datasets for 200 features obtained by Autoencoder (red line) and the full dimensional datasets (blue line). Second row shows the percentage difference between these two feature spaces in each dataset. Demonstrating that 12 datasets have improved by at least 10% their perfomance in the latent feature space created by Autoencoder. On the other hand, only the dataset 32 (named GSE14671) has reduced its performance by 10%. Which has low predictive power on full dimension space as well.

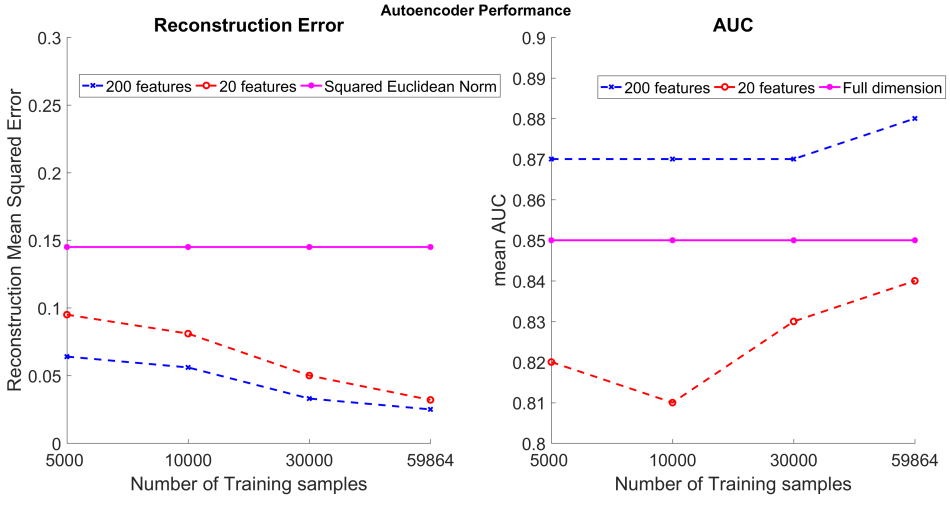

Figure 5.5: The behavior of performance relative to the number of training samples. The increase of training sample size yields improved reconstruction and classification performance. Moreover, it allows higher dimensionality reduction. Given that we have almost equivalent reconstruction error in a latent space sized 20 compared to 200 features when using 60000 samples. Also, the predictive power in 20 dimensions reaches the full dimensional space performance as we increase the size of the training sample.

5.3 Gene Set Encrichment Analysis

We presented objective measures such as reconstruction error and discriminative biological measures such as classification accuracy for assessing the dimensionality reduction methods. However, biologists are also interested on particular biomarkers (i.e., features) that are able to discriminate the states of the measured cells. For a more detailed biological perspective of the features, we performed Gene Set Enrichment Analysis (GSEA) [41] using principal component directions-weights (PCWs) as enrichment scores. PCWs, which are $d \times 1$ vectors (d number of features), correspond to the columns of transformation matrix in PCA analysis while GSEA is the process that determines when a pre-defined group of genes is differentially expressed. These pre-defined groups of genes are called pathways and have been prescribed by biologists over the years. Each group contains genes that are involved in the same biological processes and function or have similar patterns. A popular pathway database is KEGG [21–23] from which we downloaded 186 different gene sets. For each gene set of KEGG we separate the PCW into weights of genes that belong to the set-pathway (A1) and the rest (A2). Intuitively if A1 is statistical different from A2 we say that this gene set is being enriched. We used Wilcoxon Rank Test [42] as statistical test and $\alpha = 0.05$ as p-value threshold. To increase the statistical power with the risk of incorrectly rejecting a true null hypothesis (a "false positive"), a control of False Discovery Rate (FDR) [43] is being used. For robustness purposes, we do not take into account probes that do not indicate on any gene as well as probes that point to the same gene. Finally, we exclude gene-sets that belong to KEGG but have less than 10 genes measured by the analysed platforms. The remaining gene-sets are 143 and 161 for GPL570 and GPL96 platforms, respectively. The same process is repeated for each of the 200 PCWs with the highest eigenvalues for both platforms.

Fig. 5.6 graphically demonstrates the results of GSEA for each gene-set on every PCW. The x-axis represents gene-sets while the y-axis corresponds to PCWs. A dot indicates that the corresponding gene-set has been enriched by the corresponding PCW. Different colors distinguish gene-sets on six broad biological categories reported in the legend of Fig. 5.6. Almost all of the examined pathways were enriched, 142 from 143 for GPL570 and 156 from 161 for GPL96. As expected the strongest 20 PCs enrich most of the pathways for both platforms, since the strongest 20 PCs enriched 140 out of 142 pathways for GPL570 and 142 out of 156 for GPL96 demonstrating that the strongest PCWs have crucial biological information. However, as we can see in Fig. 5.7 more PCWs are required to enrich the rest pathways indicating that weaker components are necessary for a complete gene expression analysis. Finally, an interesting observation is that there are 7 gene-sets in Fig. 5.6 that were enriched for the majority of PCWs and for both GPL570 and GPL96 platforms. These gene-sets are: 'Oxidative Phosphorylation', 'Ribosome', 'Complement and Coagulation Cascades', Alzheimers Disease, 'Parkinsons Disease', 'Huntingtons Disease' and 'Systemic Lupus Erythematosus'. On the other hand, there are sets like 'Regulation of Autophagy' that were not enriched from any of the first 200 PCWs of GPL570 and the sets 'Glycosaminoglycan Biosynthesis Heparan Sulfate', 'Hedgehog Signaling Pathway',

'Melanoma' and 'Small Cell Lung Cancer' from GPL96. The importance of these findings comes from that PCWs consider interactions of genes and seems that they capture the underlying biological mechanisms of gene expression.

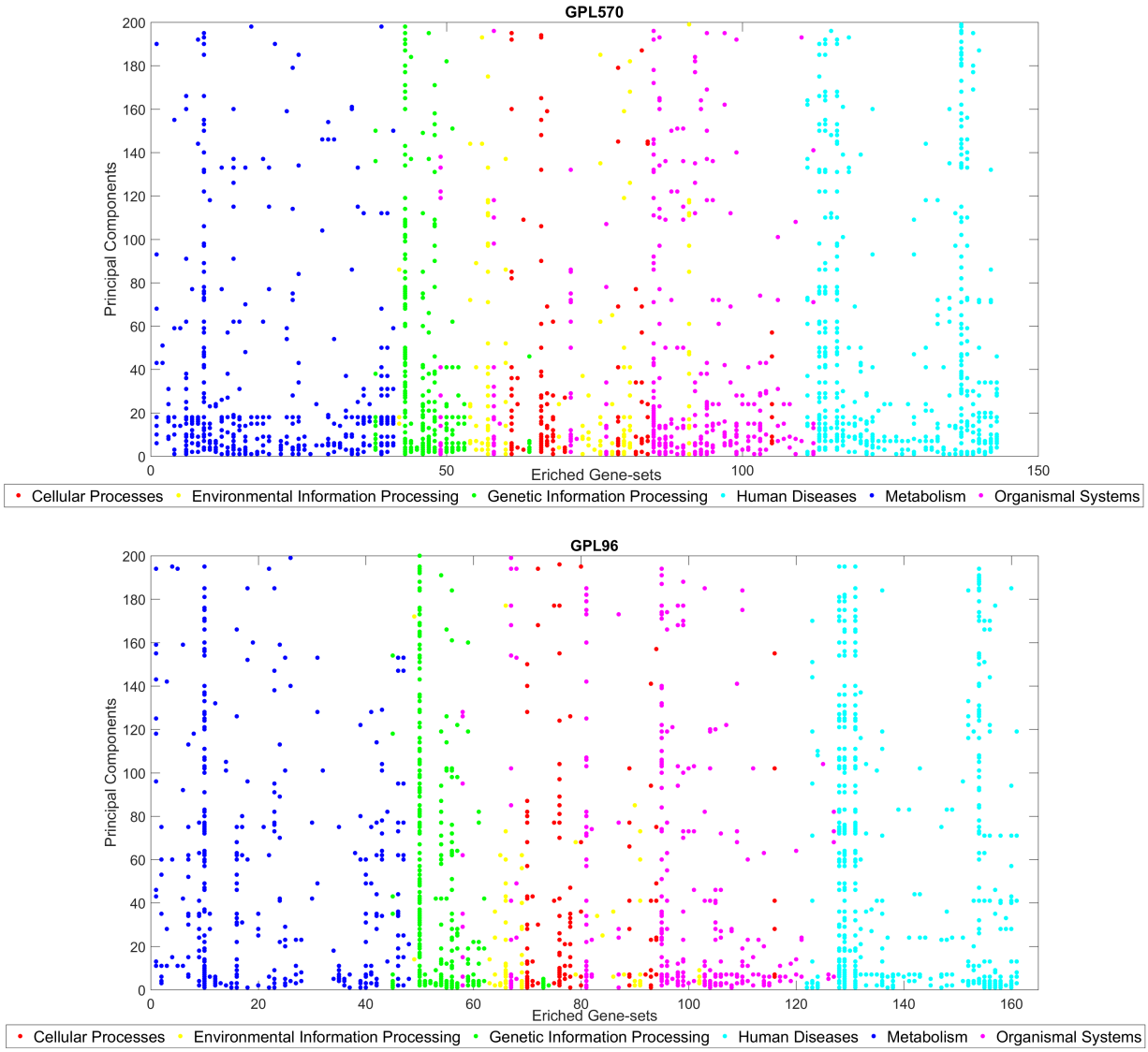

Figure 5.6: Gene Set Enrichment Analysis. Each dot at position (i,j) indicates that the i-th gene-set is being enriched by the j-th Principal Component Absence of a dot at (i,j) means the contrary. The upper plot correspond to GPL570 platform while the lower plot to GPL96. Different colors distinguish the biological categories that each KEGG gene-set belongs to.

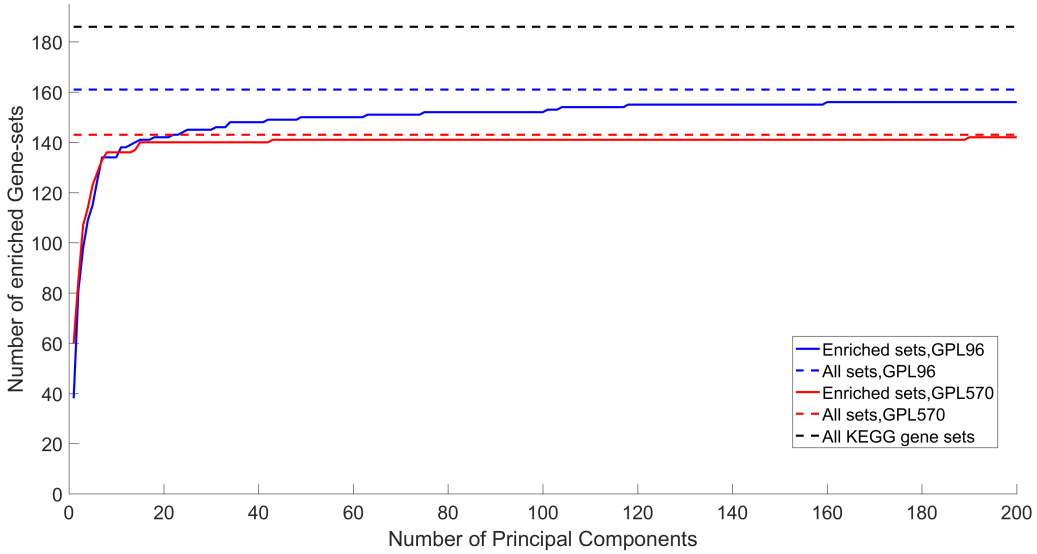

Figure 5.7: Summary of the accumulated number of gene-sets that have being enriched using PCs. The black dashed line shows number of all 186 KEGG gene-sets. Blue dashed line is equal to 143 which is the number of gene-sets considered using GPL570. Red dashed line is 161 which coresponds to number of genesets of GPL96 .They are less than the complete number of gene-sets because we do not consider gene-sets that have fewer than 10 genes represented from the corresponding platform. The two solid lines show how many unique gene-sets are being enriched using a number of principal component, max for GPL570 is 142 and for GPL96 is 156.

5.4 Cross-platform integration

Motivated by the large scale experiment, we investigated whether the increase of samples by the integration of different platforms could be beneficial to the dimensionality reduction techniques. The fusion of datasets from different platforms is not trivial due to the increased heterogeneity of the measurements. Indeed, not only their initial feature spaces are different but also each platform has its statistical characteristics which results in strong batch effects. Nevertheless, we perform cross platform integration by initially projecting each platform's datasets to a 500 dimension space using PCA as shown in Fig. 5.1(1.b), then we concatenate the projected samples from the different platforms and perform further dimensionality reduction. We do not preprocess the projected data for batch effect removal letting the dimensionality reduction methods to learn the batch information of the platforms. After merging the datasets from two or more platforms, we follow the same procedure as in the "within platform integration" case.

Two different human microarray platforms GPL570 and GPL96 are integrated in one dataset. Since the probe set of GPL96 is a subset of the probes of GPL570, it is highly probable that a joint analysis would create a common latent feature space. We have also visualized the results obtained from within platform integration (dark blue bars) for reference comparison. Furthermore, we compared the reference with the new results using t-test, with the detailed results being in the tables 12,13 and 14. Utilizing PCA and Autoencoder, the reconstruction error remained stable (Fig. 5.8(a) & 5.8(b)) and the classification accuracy was slightly increased when the two microarray platforms were merged as it is evident from Fig. 5.8(e) & 5.8(f) (light blue bars compared to the dark ones). Kernel PCA with 2 polynomial kernel had an increased reconstruction error however the predictive performance did not highly affected. On the other, gaussian Kernel PCA, independent the fusion combination, had very poor results on both reconstruction and classification performance due to the inappropriate gamma training parameter. Overall, the integration of the two human microarrays leaded us to learn a broad map of human gene expression which enabled increased predictive accuracy of newly-seen datasets from both platforms especially when the dimensionality reduction was performed utilizing autoencoder neural networks. In addition, we examined if there is a common latent feature space for microarray dataset independent the analyzed species. Therefore, we integrated the human microarray platforms with a mice muscul microarray platform denoted by GPL1261. Moreover, we explore the possibility of integrating NGS data which also measure human gene expression profiles with the human microarrays improve the performance in terms of predictive accuracy. Both reconstruction error and AUC did not reveal any specific trend when either GPL1261 or NGS data are merged with the human microarrays (green and yellow bars in Fig. 5.8).

Interestingly, we observe no deterioration of the performance in almost all cases which is also evidenced by the statistical tests that have been carried out. Implying that the constructed latent spaces are still valid and contain biological information that can be utilized for prognostic and predictive purposes. However, it seems that more sophisticated integration is required in order

to gain statistically better results.

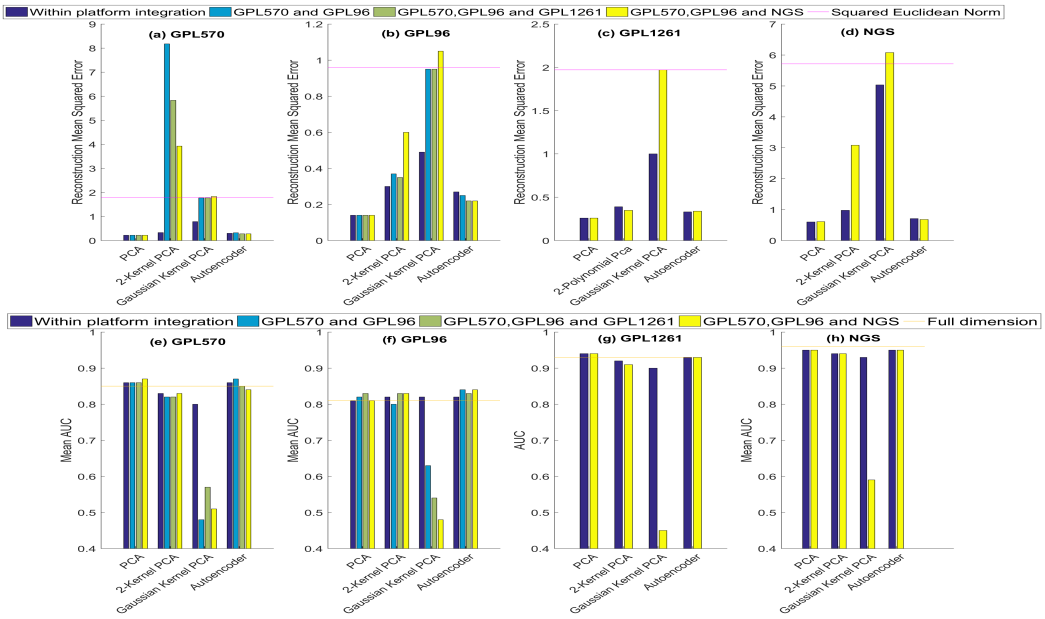

Figure 5.8: Performance metrics for integrating different platforms from the same as well as different species and differentechnologies. The reconstruction error (upper row of panels) remianed majorly unaffected by the merging of heterogeneous datasets. In contrast, the classification power (lower row of panels) of microarrays is mostly increasing after the fusion of the platforms' sets (blue and light blue bars at (e) and (f)), especially with Autoencoder which needs large amounts of data in order to be correctly trained. Furthermore, the fusion of three different microarray platforms (green bars at (e), (f) and (g)) shows that the prediction power does not deteriorate.

Chapter 6

Summary

6.1 Discussion

In this study, we analyzed a large number of available gene expression datasets, apply and compare dimensionality reduction methods and fuse datasets from the same as well as different platforms. Despite not being the first who applied dimensionality reduction techniques on gene expression data, our novelty stems from the fact that we are utilizing hundreds of datasets that belong to the same platform or belong to different platforms and extensively search for the appropriate dimension of the latent representations. The constructed latent feature spaces produced robust performance results on newly-seen datasets thus we could benefit from memory space saving and exceptional reduction of calculation time while maintaining or even improving the classification performance. As we showed earlier, the mean prediction accuracy of a new dataset is markedly improved in the constructed latent feature space compared to the accuracy on the raw features. Moreover, we significantly reduced the computational time of a comprehensive predictive analysis required by machine learning tools such as JAD Bio. For instance, a predictive analysis using raw gene expression data takes hours, in contrast to the same analysis in the constructed latent feature space which requires only few seconds. An additional advantage is that we gain a high compression of the data without deteriorating their performance. For example, the Test-sets that we used in Section 5.2 (Large scale within platform integration experiment) from 2.68 GB can be converted to 200 dimensions in 10.3 MB making feasible a quick real-time analysis using even a mobile phone with much higher accuracy as evidenced by the results.

Additionally, we integrated gene expression data of different platforms effectively since we obtained similar or slightly improved reconstruction and classification performance. Our approach considers every dependency relationships among the genes across platforms without discarding information. However, a more sophisticated approach is required for statistically improved results. It is expected that this will guide to further research as to train thoroughly a Neural Network with all available studies from several platforms. In general, being capable of integrating studies from different platforms opens new scientific direction to bioinformatics and brings closer the

dream towards personalized medicine.

6.2 Conclusion

Gene expression datasets have low sample number and are high dimensional making the integration as well as the dimensionality reduction two mandatory steps for robust and reliable statistical and computational analysis. In this study, we integrated hundreds of studies from four different platforms and extensively investigated the construction of latent feature spaces using various dimensionality reduction methods (PCA, Kernel PCA, Neural Network Autoencoder). We demonstrated that a large dimensionality reduction is possible without affecting the underlying biological information. In addition, we showed that dimensionality reduction techniques that can handle nonlinear interactions of genes achieved better classification outcomes. Furthermore, we scaled up to approximately 900 datasets with 59864 unique samples where we observed that our results are very robust. Actually, we managed to increase both the reconstruction accuracy and the classification performance on unseen datasets showing that the integration of all available gene expression datasets can lead to the construction of low-dimensional latent representations with high predictive performance. We also performed a two-step cross-platform integration where we showed that the fusion of related microarray platforms (such as GPL570 and GPL96) results in maintaining predictive performance. Overall, the Neural Network Autoencoder method demonstrated the best performance in terms of classification accuracy, especially when the number of available samples is large paving the road for further research on training and use of neural networks on genomics data applications.

Appendices

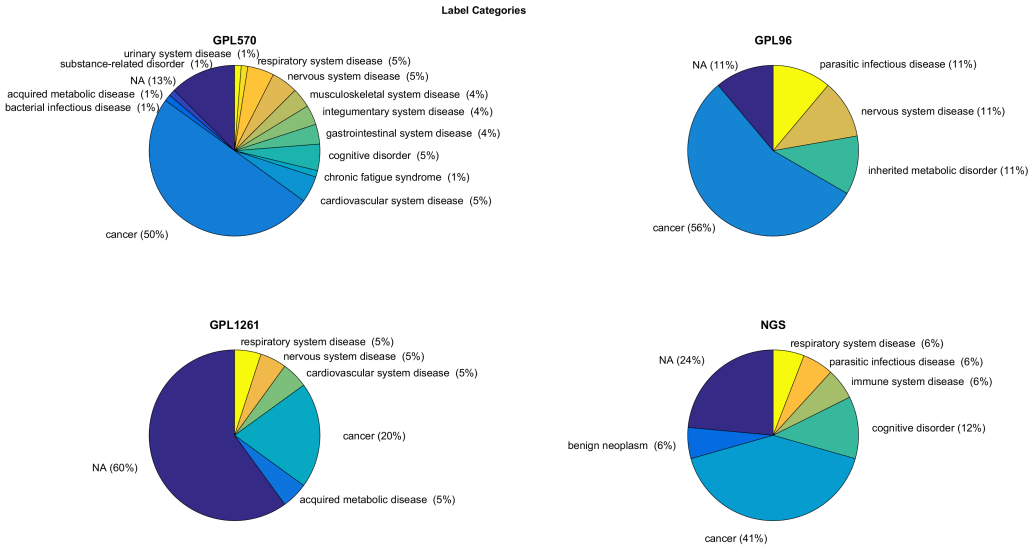

Figure 1: Categories of Test sets' labels for each platform

Data-set	#Samples	#Classes	Type of Classes
1 GDS1269	45	3	non-smoker control/ asthmatic disease control/ smoker
2 GDS1962	180	4	non-tumor/ astrocytomas/ glioblastomas/ oligodendrogliomas
3 GDS2821	25	2	control/ Parkinson's disease
4 GDS3341	41	2	control/ nasopharyngeal carcinoma
5 GDS3627	58	2	squamous cell carcinoma/ adenocarcinoma
6 GDS3795	200	2	myelodysplastic syndrome/ healthy
7 GDS4130	104	2	control/ thapsigargin
8 GDS4181	80	2	AML-multilineage dysplasia sole (AML-MLD-sole)/ AML-not otherwise specified (AML-NOS
9 GDS4206	197	3	early relapse/late relapse/ no relapse
10 GDS4602	180	2	psoriasis/ healthy
11 GDS4837	88	3	control/bipolar, medicated/bipolar, first-episode unmedicated
12 GDS3884	50	3	type 2 diabetes/normoglycemia (FH-)/ normoglycemia (FH+)
13 GDS4198	70	3	gastric cancer subtype: invasive/ gastric cancer subtype: metabolic/ gastric cancer subtype: proliferative
14 GDS4274	130	2	septic shock/ healthy control
15 GDS3539	82	2	control/ psoriasis
16 GDS3837	120	2	lung cancer/ control
17 GDS3952	162	6	benign breast abnormalities/ ectopic (gastrointestinal and brain) cancers/ malignant breast cancer/ healthy/Pre-Surgery (malignant)/Post-Surgery (malignant)
18 GDS4182	96	2	AML-myelodysplasia related changes (AML-MRC)/AML-multilineage dysplasia sole + AML-not otherwise specified (AML-MLD-sole + AML-NOS)
19 GDS4265	143	3	COPD Stage 2/COPD Stage 3/ COPD Stage 4
20 GDS4456	93	5	stage pTa/ stage pT1/ stage pT2/ stage pT3/ stage pT4

Table 1: **GPL570** Test-set used in "Within platform integration" experiment

Data-set	#Samples	#Classes	Type of Classes
1 GDS1062	27	2	no metastasis/ metastasis
2 GDS1815	100	4	WHO grade III/ WHO grade IV/ WHO grade IV without necrosis/ WHO grade IV with necrosis
3 GDS2880	20	3	normal stage I cRCC stage II cRCC
4 GDS3057	64	2	leukemia/ normal
5 GDS3713	79	2	stress (control/ cigarette smoke)
6 GDS810	31	4	control/ incipient AD/ moderate AD/ severe AD
7 GDS2643	56	4	normal/ Waldenstrom's macroglobulinemia/ chronic lymphocytic leukemia/multiple myeloma
8 GDS3097	48	2	non-inflammatory breast cancer/ inflammatory breast cancer
9 GDS2362	71	3	uninfected/ presymptomatic, experimentally acquired/ symptomatic, naturally acquired

Table 2: **GPL96** Test-set used in *"Within platform integration"* experiment

Data-set	#Samples	#Classes	Type of Classes
1 GSE58307	20	2	kras expression: No/ Yes
2 GSE58629	23	3	Mouse cerebellar tumour/ Non-neoplastic mouse cerebellar cell/ Non-neoplastic mouse cerebellar tissue
3 GSE58654	25	2	exposure: Air/Hyperoxia
4 GSE60413	89	4	tissue: cerebellum/ liver/ midbrain/ striatum
5 GSE6116	71	3	Bioanalyzer Results: Good/ Sample Preservation: RNA later/ Strain or Line: B6C3F1
6 GSE61659	58	2	genotype/variation: loxP TP53/RB/PTEN (no Cre)/ strain background: FVB
7 GSE61937	37	6	Cortex from tamoxifen treated mouse/ Cortex from vehicle treated mouse/ Hippocampus from tamoxifen treated mouse/ Hippocampus from vehicle treated mouse/ Hypothalamus from tamoxifen treated mouse/ Hypothalamus from vehicle treated mouse
8 GSE63027	39	4	disease status: healthy/ hepatocellular carcinoma/ non-aollcoholic steatohepatitis/ steatosis
9 GSE65997	50	2	gender: female/ male
10 GSE67985	60	2	tissue: Distal bonetissue/Proximal bone
11 GSE68515	20	4	group: aged/impaired/ unimpaired/young
12 GSE7404	144	2	mouse leukocytes/ mouse splenocytes
13 GSE76628	78	8	tissue: flank/ treated with: DC101 20 mpk at day 20/ DC101 20 mpk at day 5/ DC101 20 mpk at day 60/ G6 10 mpk at day 20/ G6 10 mpk at day 5/ G6 10 mpk at day 60/none
14 GSE7793	48	2	agent: saline/ vancomycin
15 GSE8150	20	2	Old neocortex/ Young neocortex
16 GSE84245	23	2	genotype: Ezh1; Ezh2 Camk Cre/ genotype: WT
17 GSE8790	22	2	Air exposed /CS exposed
18 GSE8949	20	7	Aorta from: P465L PPAR gamma/ control mice/mice at a dose of 10 mg/kg/day for 14 days/ mice at a dose of 10 mg/kg/day for 2 days/ mice at a dose of 3 mg/kg/day for 14 days/ mice at a dose of 3 mg/kg/day for 2 days/ wildtype mice

19 GSE9444	131	12	Strain: AKR/J, Tissue: Liver/ Strain: AKR/J, Tissue: whole brain/ Strain: C57BL/6J, Tissue: Liver/ Strain: C57BL/6J, Tissue: whole brain/ Strain: DBA/2J, Tissue: Liver/ Strain: DBA/2J, Tissue: whole brain/ mRNAs pull-down, control/ mRNAs pull-down,sleep deprivation/ total RNA pull-down supernatant, control/ total RNA pull-down supernatant, sleep deprivation/ total RNA, control/ total RNA, sleep deprivation
20 GSE9763	20	4	Control embryonic progenitors/ Control postnatal progenitors/ Transduced embryonic progenitors/ Transformed postnatal progenitors

Table 3: **GPL1261** Test-set used in *"Within platform integration"* experiment

Data-set	#Samples	#Classes	Type of Classes
1 SRP026126	422	4	tissue: Brain, reference rna: Agilent Universal Human Reference RNA, referen.../ tissue: Brain, reference rna: FirstChoiceB Human Brain Reference Total RNA.../ tissue: Pooled tumor, reference rna: Agilent Universal Human Reference RNA../ tissue: Pooled tumor, reference rna: Agilent Universal Human Reference RNA...
2 SRP030617	113	5	cdna synthesis method: Clontech SMARTer/ NuGEN Ovation/ NuGEN Ovation/ Sigma WTA TransPlex/ Superscript RT
3 SRP032775	232	4	time point, infection agent: post-infection Plasmodium falciparum/ pre-infection, n/a/ Post-infection, Plasmodium falciparum (Pf)/ Pre-infection, n/a
4 SRP033266	144	2	tissue: Bone marrow/ Heparinised blood
5 SRP033725	62	2	disease state: BD/ Control
6 SRP035988	179	2	tissue type: lesional psoriatic skin/ normal skin
7 SRP037775	63	6	cell line, drug treatment: BT474, drug treatment: no drug/ BT474, trastuzumab/ BTR50, no drug/ BTR50, trastuzumab/ HCC1954, no drug/ HCC1954, trastuzumab
8 SRP041471	313	4	cell line,time, treatment : HeLa, 0 min, 3 h DRB 0 min 4sU/ HeLa, 4 min, 3 h DRB 4 min 4sU/ HeLa, 8 min, 3 h DRB 8 min 4sU/ HeLa, control, untreated
9 SRP041538	189	2	disease state: COPD/ Normal
10 SRP042620	168	6	psoriasis/ healthy
11 SRP044668	94	3	tissue type: glioma - contrast-enhancing sample glioma - non-enhancing FLAIR+ sample non-neoplastic brain
12 SRP048759	434	3	tissue: Bone marrow leukemia/ Heparinised blood/ Leukapheresis
13 SRP050223	402	2	tissue: T cell acute lymphoblastic leukemia/ normal thymus
14 SRP050992	460	2	passages: 30-35, treatment: not sorted/ passages: 35-40, treatment: FACS sorted
15 SRP051848	188	4	condition,time-point: Case (PTSD risk), Pre-deployment/ Case (PTSD),Post-deployment/ Control Post-deployment/ Control, Pre-deployment
16 SRP052740	169	3	mapki sensitivity,treatment: resistant, BRAFi/ resistant, BRAFi+MEKi/ sensitive, none

17 SRP056295	525	4	tissue: Bone marrow/ EDTA Blood/Heparinised blood/ Leukapheresis

Table 4: **NGS** Test-set used in *"Within platform integration"* experiment

Data-set	#Samples	#Classes	Type of Classes
1 GSE2125	45	3	non-smoker control/ asthmatic disease control/ smoker
2 GSE2125	180	4	non-tumor/ astrocytomas/ glioblastomas/ oligodendrogliomas
3 GSE7621	25	2	control/ Parkinson's disease
4 GSE12452	41	2	control/ nasopharyngeal carcinoma
5 GSE10245	58	2	squamous cell carcinoma/ adenocarcinoma
6 GSE19429	200	2	myelodysplastic syndrome/ healthy
7 GSE19519	120	2	control/ thapsigargin
8 GSE21261	80	2	AML-multilineage dysplasia sole (AML-MLD-sole)/ AML-not otherwise specified (AML-NOS
9 GSE13576	197	3	early relapse/late relapse/ no relapse
10 GSE13355	180	2	psoriasis/ healthy
11 GSE46449	88	3	control/bipolar, medicated/bipolar, first-episode unmedicated
12 GSE25462	50	3	type 2 diabetes/normoglycemia (FH-)/ normoglycemia (FH+)
13 GSE35809	70	3	gastric cancer subtype: invasive/ gastric cancer subtype: metabolic/ gastric cancer subtype: proliferative
14 GSE26440	130	2	septic shock/ healthy control
15 GSE14905	82	2	control/ psoriasis
16 GSE19804	120	2	lung cancer/ control
17 GSE27567	162	6	benign breast abnormalities/ ectopic (gastrointestinal and brain) cancers/ malignant breast cancer/ healthy/Pre-Surgery (malignant)/Post-Surgery (malignant)
18 GSE21261	80	2	AML-myelodysplasia related changes (AML-MRC)/AML-multilineage dysplasia sole + AML-not otherwise specified (AML-MLD-sole + AML-NOS)
19 GSE22148	143	3	COPD Stage 2/COPD Stage 3/ COPD Stage 4
20 GSE31684	93	5	stage pTa/ stage pT1/ stage pT2/ stage pT3/ stage pT4
21 GSE10041	72	3	no relaxation response practice/ 8 weeks of relaxation response practice/ long-term daily relaxation response practice
22 GSE10063	60	2	smoker/non-smoker
23 GSE10810	58	2	control/ tumor
24 GSE10927	65	3	Human adrenocortical carcinomas (33)/ adenomas (22)/ and normal adrenal cortex (10)
25 GSE11135	204	2	five-day course of protocol training/ independent proficiency testing

26 GSE11869	75	5	5 Doses [Vehicle Control/Very Low (1 pM)/ Low (100 pM)/ High (1 nM)/ Very High (1 uM)]
27 GSE13139	54	2	LOX-1 overexpression/control
28 GSE13367	56	2	mucosal colonic biopsies /isolated colonocytes
29 GSE13548	42	4	treated cells [glucose free/ medium normal growth condition/ 2-Deoxy-D-glucose (2DG)/ Tunicamycin (TM)]
30 GSE13732	113	2	CIS patients / controls
31 GSE13911	69	2	gastric tumors/control
32 GSE14671	59	2	responce in chronic phase cml patiens treated with imatinib/ not response
33 GSE14924	41	2	AML/ healthy
34 GSE15605	74	3	normal skin/ primary melanoma/ melanoma metastasis
35 GSE15913	40	2	thalidomide treated/ untreated
36 GSE16059	88	3	controls/ chronic fatigue syndrome/ idiopathic chronic fatigue
37 GSE16214	240	3	controls/chronic fatigue syndrome / idiopathic chronic fatigue
38 GSE16515	52	2	pancreatic tumor/ control
39 GSE17612	51	2	schizophrenic/ control
40 GSE18206	48	2	skin irritants SLS/ non
41 GSE18781	55	3	Axial Spondyloarthropathy/ control/ Sarcoidosis
42 GSE18842	91	2	lung cancer/ control
43 GSE19188	156	2	tumor/ normal lung tissue samples
44 GSE20489	54	2	acute ethanol exposure/ control
45 GSE21138	59	2	schizophrenic/ control
46 GSE21545	223	2	carotid plaques/ peripheral blood mononuclear cells
47 GSE21610	68	3	non-failing hearts (NF)/ VAD-HTx/ VAD-IP
48 GSE22229	58	3	Tolerant (TOL) participants/ Standard Immunotherapy (SI) participants/ Healthy Controls (HC)
49 GSE22459	65	3	histologically normal ($n = 25, i/cg/ci = 0$)/ IF/TA ($n = 24, i/cg = 0, ci > 0$)/ IFTA+i ($n = 16, cg = 0, i/ci > 0$)
50 GSE24147	42	2	recent-onset (RO) T1D sera/ control
51 GSE26051	46	2	diseased tendons/ healthy
52 GSE27383	72	2	schizophrenic/ control
53 GSE27536	54	2	COPD patient/ healthy
54 GSE27858	56	2	before/ after treatment with SPC2996
55 GSE28750	41	3	Sepsis/ Post-Surgical/ Control

56 GSE29265	49	3	control/ Papillary thyroid carcinoma/ Anaplastic thyroid carcinoma
57 GSE29722	20	2	cancer/ healthy
58 GSE31189	92	2	Cancer Urothelia/ non- Cancer Urothelia
59 GSE32448	80	2	HomoProstateN/ HomoProstateT
60 GSE32688	32	2	Pancreatic Cancer/ non-malignant pancreas
61 GSE36895	76	3	clear-cell renal cell carcinoma (ccRCC) primary tumors/ tumors growing in immunodeficient mice (tumorgrafts)/ and normal kidney cortices
62 GSE38666	45	3	Normal/ Cancer Stroma/ Cancer Epithelia
63 GSE39156	64	2	hydrogen peroxide/ control
64 GSE40595	77	4	Normal/ Ovarian cancer sttroma/ Human ovarian surface epthelium/ Tumor epthelial component
65 GSE40611	49	2	control/ primary Sjogren syndrome
66 GSE40791	194	2	non-neoplastic (N) lung samples/ lung adenocarcinoma (AD) frozen tissues
67 GSE4183	53	4	frozen colonic biopsies of patients with CRC/ adenoma/ IBD /healthy normal controls
68 GSE42057	136	2	chronic obstructive pulmonary disease/ control
69 GSE42568	121	2	Breast cancer/ control
70 GSE43592	20	2	multiple sclerosis (MS)/ control
71 GSE46474	40	2	rejection kidney transplant patients/ non-rejection
72 GSE47908	60	4	left-sided colitis/ pancolitis/ UC-associated dysplasia/ controls
73 GSE50006	279	2	hronic lymphocytic leukemia (CLL) tumors/ healthy donors
74 GSE50772	81	2	SLE patients/ controls
75 GSE51024	96	2	Malignant Pleural Mesothelioma Tumor/ Normal Lung tissue
76 GSE58294	92	2	CardioembolicSTROKE/ control
77 GSE59312	79	2	HCV/ control
78 GSE59312	129	2	SLE patient/ healthy
79 GSE63514	128	5	normal/ CIN1 lesions/ CIN2 lesions/ CIN3 lesions/ cancers specimens
80 GSE64300	42	3	PBMC-tolerant/ PBMC-non-tolerant/ control

Table 5: **GPL570** Test-set used in *"Large scale within platform integration" experiment*

Dimensions	PCA	2-PCA	3-PCA	Gaussian PCA	Autoencoder
20	0.00002	0.00012	0.00007	0.00019	0.08180
50	0.04090	0.05550	0.01360	0.05250	0.47240
200	0.02760	0.11910	0.26690	0.31010	0.00410
500	0.00150	0.00190	0.00930	0.00630	-

Table 6: P-values obtained by performing a t-test in order to compare the AUC results between the reference (full dimension) and constructed latent feature spaces in **"Large scale within platform integration"** experiment. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Dimensions	PCA	2-PCA	3-PCA	Gaussian PCA	Autoencoder
5	0.00002	0.00001	0.00001	0.00006	0.00340
20	0.01640	0.00490	0.00360	0.00500	0.10380
200	0.27140	0.28250	0.07750	0.04410	0.21450

Table 7: P-values obtained by performing a t-test in order to compare the AUC results between the reference (full dimension) and constructed latent feature spaces in **"Within platform integration"** experiment for **GPL570** sets. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Dimensions	PCA	2-PCA	3-PCA	Gaussian PCA	Autoencoder
5	0.09040	0.20670	0.02680	0.02380	0.16560
20	0.99640	0.85800	0.13240	0.28050	0.70460
200	0.55700	0.24100	0.87180	0.39890	0.19650

Table 8: P-values obtained by performing a t-test in order to compare the AUC results between the reference (full dimension) and constructed latent feature spaces in **"Within platform integration"** experiment for **GPL96** sets. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Dimensions	PCA	2-PCA	3-PCA	Gaussian PCA	Autoencoder
5	0.00043	0.00240	0.00036	0.00150	0.00190
20	0.17760	0.08300	0.08880	0.00210	0.03080
200	0.43830	0.35490	0.04800	0.09640	0.59890

Table 9: P-values obtained by performing a t-test in order to compare the AUC results between the reference (full dimension) and constructed latent feature spaces in **"Within platform integration"** experiment for **GPL1261** sets. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Platform	PCA	2-PCA	Gaussian PCA	Autoencoder
GPL570	0.09360	0.07050	0.00000	0.09370
GPL96	0.55670	0.83580	0.00730	0.67590

Table 12: P-values obtained by performing a t-test in order to compare the AUC results between the reference the constructed latent feature spaces in "Within platform integration" experiment and and **Cross-Platform Integration** using **GPL570 and GPL96** sets. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Platform	PCA	2-PCA	Gaussian PCA	Autoencoder
GPL570	0.47940	0.45920	0.00000	0.08260
GPL96	0.25610	0.09620	0.00110	0.78130
GPL1261	0.45890	0.84240	0.00000	0.64470

Table 13: P-values obtained by performing a t-test in order to compare the AUC results between the reference the constructed latent feature spaces in "Within platform integration" experiment and and **Cross-Platform Integration** using **GPL570, GPL96 and GPL1261** sets. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Dimensions	PCA	2-PCA	3-PCA	Gaussian PCA	Autoencoder
5	0.02470	0.00200	0.00250	0.00250	0.00730
20	0.03620	0.00910	0.00740	0.00240	0.03540
200	0.05920	0.02880	0.03240	0.01970	0.09160

Table 10: P-values obtained by performing a t-test in order to compare the AUC results between the reference (full dimension) and constructed latent feature spaces in **"Within platform integration"** experiment for **NGS** sets. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Dimensions	PCA	2-PCA	3-PCA	Gaussian PCA	Autoencoder
20	0.00002	0.00012	0.00007	0.00019	0.08180
50	0.04090	0.05550	0.01360	0.05250	0.47240
200	0.02760	0.11910	0.26690	0.31010	0.00410
500	0.00150	0.00190	0.00930	0.00630	-

Table 11: P-values obtained by performing a t-test in order to compare the AUC results between the reference (full dimension) and constructed latent feature spaces in **"Large scale within platform integration"** experiment. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

Platform	PCA	2-PCA	Gaussian PCA	Autoencoder
GPL570	0.12070	0.68980	0.00000	0.23260
GPL96	0.92540	0.58380	0.01410	0.45360
NGS	0.30260	0.83790	0.00000	0.53660

Table 14: P-values obtained by performing a t-test in order to compare the AUC results between the reference the constructed latent feature spaces in "Within platform integration" experiment and and **Cross-Platform Integration** using **GPL570, GPL96 and NGS** sets. If p-value < 0.05 there is a statistically important difference between AUC of the reference (full dimensional) and the AUC in the constructed latent feature space; gray color cell.

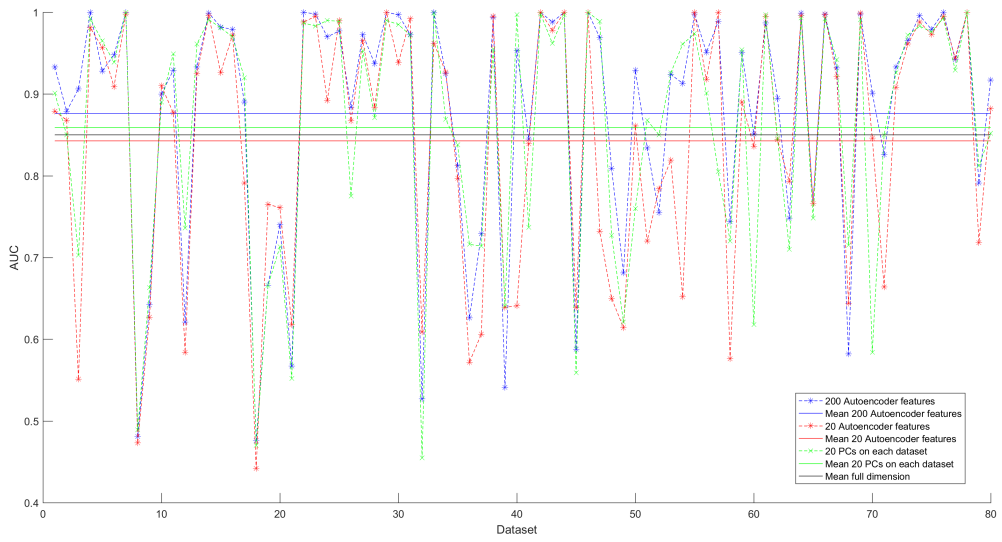

Figure 2: Area Under the Curve comparison of pre-trained features using integration and features obtained by simple PCA on each dataset seperately. We observe that 20 first PCs of each datasets perform slightly better than 20 autoencoder's features. However getting PCs from the whole datasets violates Golden Rule, which says learn from S then test on new samples S'. Since the validation set that is used in Cross-validation had been "seen" before the estimation procedure by PCA.

Bibliography

[1] E. S. Lander, "Array of hope," *Nature genetics*, vol. 21, pp. 3–4, 1999.

[2] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary dna microarray," *Science*, vol. 270, no. 5235, p. 467, 1995.

[3] Z. Wang, M. Gerstein, and M. Snyder, "Rna-seq: a revolutionary tool for transcriptomics," *Nature reviews genetics*, vol. 10, no. 1, pp. 57–63, 2009.

[4] M. A. Nowak, M. C. Boerlijst, J. Cooke, and J. M. Smith, "Evolution of genetic redundancy," *Nature*, vol. 388, no. 6638, pp. 167–171, 1997.

[5] J. Shi and Z. Luo, "Nonlinear dimensionality reduction of gene expression data for visualization and clustering analysis of cancer tissue samples," *Computers in Biology and Medicine*, vol. 40, no. 8, pp. 723–732, 2010.

[6] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[7] T. Madeswaran and G. K. Nawaz, "A comparative analysis of classification of micro array gene expression data using dimensionality reduction techniques," *IJCER*, vol. 1, no. 4, pp. 192–201, 2012.

[8] J. J. Hughey and A. J. Butte, "Robust meta-analysis of gene expression using the elastic net," *Nucleic acids research*, p. gkv229, 2015.

[9] M. Lukk, M. Kapushesky, J. Nikkilä, H. Parkinson, A. Goncalves, W. Huber, E. Ukkonen, and A. Brazma, "A global map of human gene expression," *Nature biotechnology*, vol. 28, no. 4, pp. 322–324, 2010.

[10] M. Lenz, F.-J. Müller, M. Zenke, and A. Schuppert, "Principal components analysis and the reported low intrinsic dimensionality of gene expression microarray data," *Scientific reports*, vol. 6, 2016.

[11] S. Wu, Y. Xu, Z. Feng, X. Yang, X. Wang, and X. Gao, "Multiple-platform data integration method with application to combined analysis of microarray and proteomic data," *BMC bioinformatics*, vol. 13, no. 1, p. 320, 2012.

[12] C. J. Walsh, P. Hu, J. Batt, and C. C. D. Santos, "Microarray meta-analysis and cross-platform normalization: integrative genomics for robust biomarker discovery," *Microarrays*, vol. 4, no. 3, pp. 389–406, 2015.

[13] J. Taminau, C. Lazar, S. Meganck, and A. Nowé, "Comparison of merging and meta-analysis as alternative approaches for integrative gene expression analysis," *ISRN bioinformatics*, vol. 2014, 2014.

[14] A. Buness, M. Ruschhaupt, R. Kuner, and A. Tresch, "Classification across gene expression microarray studies," *BMC bioinformatics*, vol. 10, no. 1, p. 453, 2009.

[15] P. Warnat, R. Eils, and B. Brors, "Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes," *BMC bioinformatics*, vol. 6, no. 1, p. 265, 2005.

[16] S. A. Mitchell, K. M. Brown, M. M. Henry, M. Mintz, D. Catchpoole, B. LaFleur, and D. A. Stephan, "Inter-platform comparability of microarrays in acute lymphoblastic leukemia," *BMC genomics*, vol. 5, no. 1, p. 71, 2004.

[17] K. B. Gregory, A. A. Momin, K. R. Coombes, and V. Baladandayuthapani, "Latent feature decompositions for integrative analysis of multi-platform genomic data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 11, no. 6, pp. 984–994, 2014.

[18] J. E. Jackson, *A user's guide to principal components*. John Wiley & Sons, 2005, vol. 587.

[19] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 583–588.

[20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[21] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, "Kegg: new perspectives on genomes, pathways, diseases and drugs," *Nucleic Acids Research*, vol. 45, no. D1, pp. D353–D361, 2017.

[22] M. Kanehisa, Y. Sato, M. Kawashima, M. Furumichi, and M. Tanabe, "Kegg as a reference resource for gene and protein annotation," *Nucleic acids research*, p. gkv1070, 2015.

[23] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.

[24] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," 1994.

[25] T. Fawcett, "Roc graphs: Notes and practical considerations for researchers," *Machine learning*, vol. 31, no. 1, pp. 1–38, 2004.

[26] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE transactions on information theory*, vol. 14, no. 1, pp. 55–63, 1968.

[27] G. Borboudakis and I. Tsamardinos, "Forward-backward selection with early dropping," *arXiv preprint arXiv:1705.10770*, 2017.

[28] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[29] S. Trek, "Hypothesis test: difference between proportions," *StatTrek. com*, 2016.

[30] V. Trevino, F. Falciani, and H. A. Barrera-Saldaña, "Dna microarrays: a powerful genomic tool for biomedical and clinical research," *MOLECULAR MEDICINE-CAMBRIDGE MA THEN NEW YORK-*, vol. 13, no. 9/10, p. 527, 2007.

[31] E. Clough and T. Barrett, "The gene expression omnibus database," *Statistical Genomics: Methods and Protocols*, pp. 93–110, 2016.

[32] L. Collado-Torres, A. Nellore, K. Kammers, S. E. Ellis, M. A. Taub, K. D. Hansen, A. E. Jaffe, B. Langmead, and J. Leek, "Recount: A large-scale resource of analysis-ready rna-seq expression data," *bioRxiv*, p. 068478, 2016.

[33] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[34] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[35] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[36] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[37] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[38] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[39] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[40] G. Orfanoudaki, M. Markaki, K. Chatzi, I. Tsamardinos, and A. Economou, "Maturep: prediction of secreted proteins with exclusive information from their mature regions," *Scientific Reports*, vol. 7, 2017.

[41] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander *et al.*, "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles," *Proceedings of the National Academy of Sciences*, vol. 102, no. 43, pp. 15 545–15 550, 2005.

[42] F. Wilcoxon and R. A. Wilcox, *Some rapid approximate statistical procedures*. Lederle Laboratories, 1964.

[43] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the royal statistical society. Series B (Methodological)*, pp. 289–300, 1995.