UNIVERSITY OF BRISTOL

DEPARTMENT OF ENGINEERING MATHEMATICS

Transcriptome profiling to differentiate primary cancers of unknown origin

Matthew Ramcharan (Engineering Mathematics)

Project thesis submitted in support of the degree of Master of Engineering

February 5, 2021

Supervisors: Dr. Colin Campbell, Engineering Mathematics

Abstract

Transcriptome profiling is a method used to measure RNA abundance in a tissue sample. Through RNA sequencing, the expression of individual genes can be measured, giving insight into the function of the sample, including its location in the body. This is key in the area of cancer bioinformatics, where carcinomas of unknown primary (CUPs) in organs are difficult to trace back to their primary tumour. In this report, several classifiers have been proposed, trained and tested. Training was performed on publicly available primary cancer data sets, including The Cancer Genome Atlas. These classifiers include Support Vector Machines, Multiple Kernel Learning, AdaBoost, and Deep Learning. It is proven that, using data derived from transcriptome profiling, it is possible to create highly successful classifiers to distinguish different cancerous tissues. The highest accuracy for distinguishing between 4 different cancer strains was 83% with a Deep Learning model. However, in the area of binary classification, all models were highly successful, with 100% accuracy in the classification between Pancreatic and Breast cancer for the SVM, MKL and AdaBoost models, and the Deep learner still achieving 98% accuracy.

Contents

Chapter 1

Introduction

1.1 Motivation

It is estimated that approximately 4% of all patients with cancer present with carcinomas of unknown primary (CUPs), representing a higher incidence than known malignancies such as non-Hodgkin lymphoma or ovarian cancer.

Identification of the site of origin of carcinoma of unknown primary using immunohistochemistry is a frequent requirement of anatomic pathologists. Diagnostic accuracy is crucial, particularly in the current era of targeted therapies and smaller sample sizes.

The identification of a primary site in such a setting has taken on dramatically increased clinical relevance, given the differences in prognosis and treatment; in particular, targeted therapies of carcinomas of various primary sites [1].

In previous studies into tumours of known origin, Gene Expression Profiling (GEP) correctly identified the site of origin in 85% of cases and compares favourably with immunohistochemical (IHC) staining, the current method used to definitively identify the tissue of origin [2].

1.2 Current Work in the field

By integrating morphology with well-performed and well-interpreted immunohistochemistry (IHC), a pathologist can, in most cases, provide definitive diagnostic information regarding the most likely primary site or sites of the carcinoma presenting as metastases [1].

Commercially, there are a few methods that use gene profiling such as the Pathwork Tissue of Origin (TOO), bioTheranostics Cancer type ID (CTID) or the miRview mets2 [3, 4, 5]. Accuracies on these commercially available models range from 72-95%, but they are known to be limited: for example, TOO is not ideal for sarcoma and CTID is not really feasible for pancreatic, colorectal and gastroesophageal cancers [6].

There is currently a range of studies into the use of different machine learning algorithms such as SVMs, Decision Trees, and Artificial Neural Networks, to classify the primary sites of cancers by using mRNA expression profiles (outlined in Section 2) [7, 8, 9, 2, 10, 11]. These studies have produced a maximum accuracy of 95% by using a deep learning method to classify 33 different tumour types [10].

1.3 Project Aims and Objectives

The aim of this project is to design a classifier capable of distinguishing different cancer types (Breast, Colon, etc) based on RNA-Sequencing Data. This will provide a pre-trained classifier capable of identifying the primary cancer of a CUP, which will provide physicians with guidance as to a treatment plan for the patient with the CUP.

1.4 Plan of Report

To achieve the above aim, we begin Chapter 2 by analysing the data taken from a variety of cancerous tumour samples, describing their structure and preprocessing steps, before they are suitable to be used to train and test the classification models described and formulated in Chapter 3. These models are then trained and tested for accuracy and ability to generalise, as described in Chapter 4. This chapter also contains details on parameter optimisation, and discussion on the results obtained. Chapter 5 presents conclusions drawn from this project, along with key developments and challenges associated with the problem approached. Finally, in the same chapter, we consider further work and potential extensions to this project.

Chapter 2

Data

2.1 Data Description

2.1.1 Preface on Genetic data

The simplest approach to quantifying gene expression by RNA-seq is to count the number of reads that map (i.e. align) to each gene (read count) using programs such as HTSeq-count [12].

Raw read counts are affected by factors such as transcript length as longer transcripts have higher read counts at the same expression level, and total number of reads [13]. To normalise each gene would require manual searching for each of the around 60,000 genes present in the data, and thus this transcript length bias is taken as a given for the sake of simplicity [14].

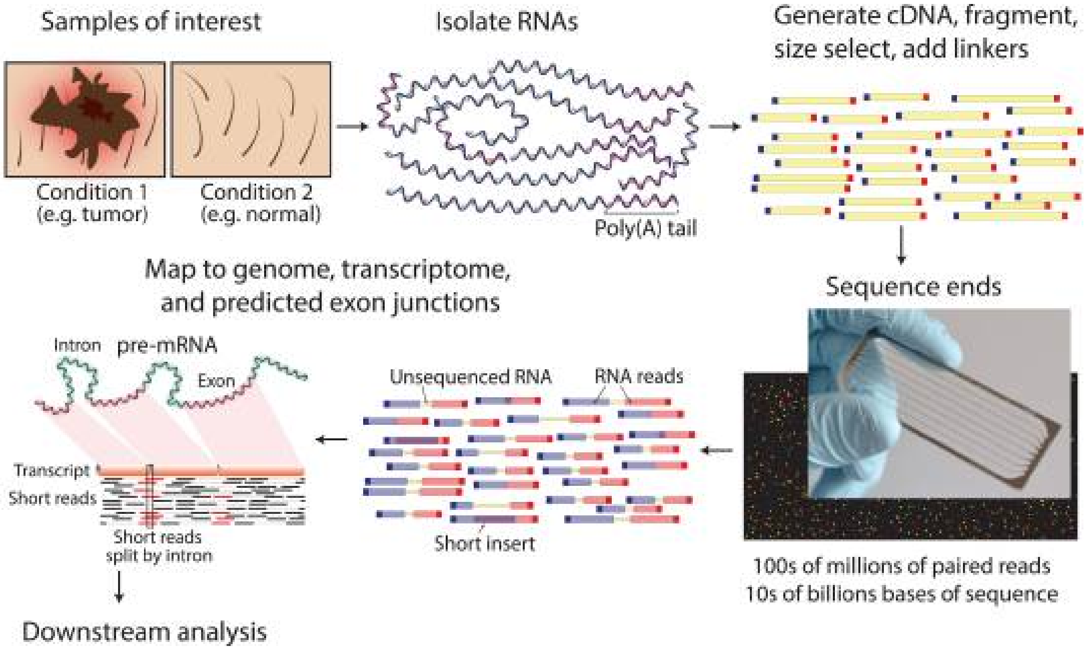

Samples of interest
Isolate RNAs
Generate cDNA, fragment, size select, add linkers
Condition 1 (e.g. tumor)
Condition 2 (e.g. normal)
Poly(A) tail
Map to genome, transcriptome, and predicted exon junctions
Sequence ends
Intron
pre-mRNA
Exon
Unsequenced RNA
RNA reads
Transcript
Short reads
Short reads split by intron
Short insert
100s of millions of paired reads
10s of billions bases of sequence
Downstream analysis

Figure 2.1: Figure demonstrating a typical RNA-seq experimental workflow [15]. This involves the isolation of RNA from samples of interest, generation of sequencing libraries, use of a high-throughput sequencer to produce hundreds of millions of short paired-end reads then alignment of reads against a reference genome or transcriptome. After this is followed by downstream analysis for expression estimation, differential expression, transcript isoform discovery, and other applications.

2.1.2 National Cancer Institute Genomic Data Commons Data Portal

The Genomic Data Commons (GDC) Data Model is the central method of organisation of all data artefacts in the GDC. The GDC is a research program of the National Cancer Institute (NCI). The mission of the GDC is to provide the cancer research community with a unified data repository that enables data sharing across cancer genomic studies in support of precision medicine [16].

The dataset used follows the internal tags from the GDC Data Portal described in Table 2.1.

Table 2.1: Data labels searched in the GDC when collecting datasets for each cancer.

Data Category	Transcriptome Profiling
Data Type	Gene Expression Quantification
Experimental Strategy	RNA-Seq
Workflow Type	HTSeq - Counts
Project	All projects were TCGA

The datasets collected by the GDC are formulated by The Cancer Genome Atlas (TCGA). This means the transcriptome data consists of TCGA-BRCA (Breast Invasive Carcinoma), TCGA-COAD (Colon Adenocarcinoma), TCGA-READ (Rectum Adenocarcinoma), and TCGA-PAAD (Pancreatic Adenocarcinoma) projects.

While the goal of this project is to identify tumours of unknown primary, as metastases are expected to retain the transcription signature of the primary tumour origin, it has been proven that predictors trained on primary tumour data can identify the primary tumour type from metastatic samples [11]. This is supported by several publications using this method [9, 17]. As a result, no samples used in this project for training or testing are actually from tumours of unknown primary.

2.1.3 Data Format

Due to several different datasets being trained and used, the general form of the data formatting is as follows.

For the purposes of training the classifiers outlined later, the integer HTSeq - counts for each sample of all cancers used are concatenated in a dataset of L rows and $m + 1$ columns, where L is the number of samples and m is the number of genes. The extra column is the label of the primary site of the cancerous sample. The genes are organised in ascending order by their ensemble reference [18], although the order of genes should not affect the quality of the classifier. The samples are ordered by their GDC file name with cancer labels at the end, however prior to training any classifier the samples were shuffled randomly to avoid any bias during validation from the training/test split of the holdout method, outlined in Section 3.2.

Table 2.2: An example of the structure of the datasets used, in this case, the Breast and Colon data. For this dataset $L = 921, m = 60483$

ENSG00000000003.13	*ENSG00000000005.5*	...	*ENSGR0000281849.12*	*Label*
3113	27	...	0	Breast
1894	14	...	0	Breast
8012	238	...	0	Breast
4623	2	...	0	Breast
⋮	⋮	⋱	⋮	⋮
8095	28	...	0	Colon
2337	16	...	0	Colon

2.2 Data Preprocessing

Prior to being used to train any classifier, the label was encoded for the binary classification task to be $y = \{+1, -1\}$ with $+1$ being whichever label appeared first in the dataset, and -1 being the other label.

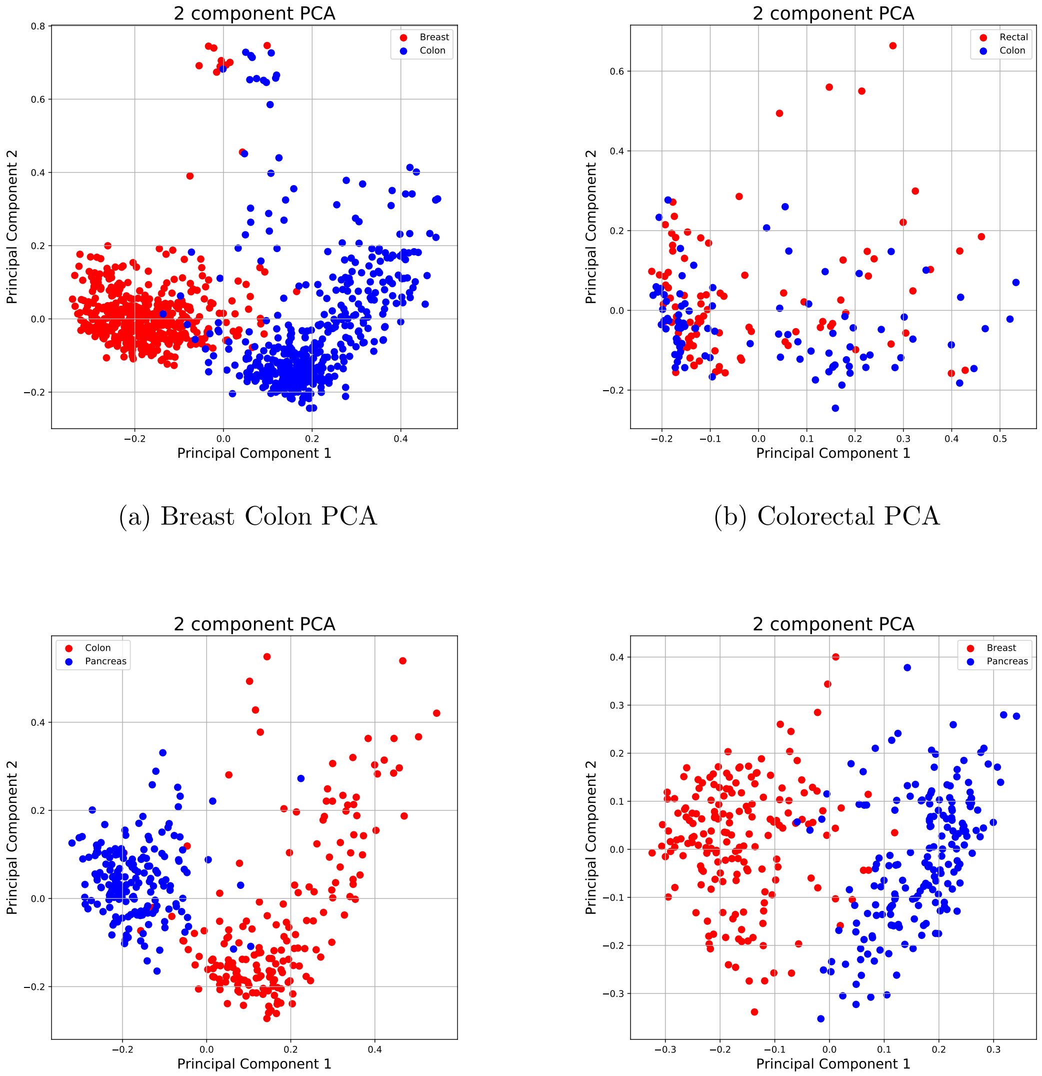
The genes were normalised

2.2.1 Feature Selection

2.3 PCA and Data Separability Representation

Principal component analysis (PCA) is a data compression method: it is used to reduce a large set of variables to a smaller set without the loss of key information. Mathematically, this is achieved by transforming a number of possibly correlated variable values into a (smaller) set of values of linearly uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component does the same with the remaining variability.

Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space is a useful tool for



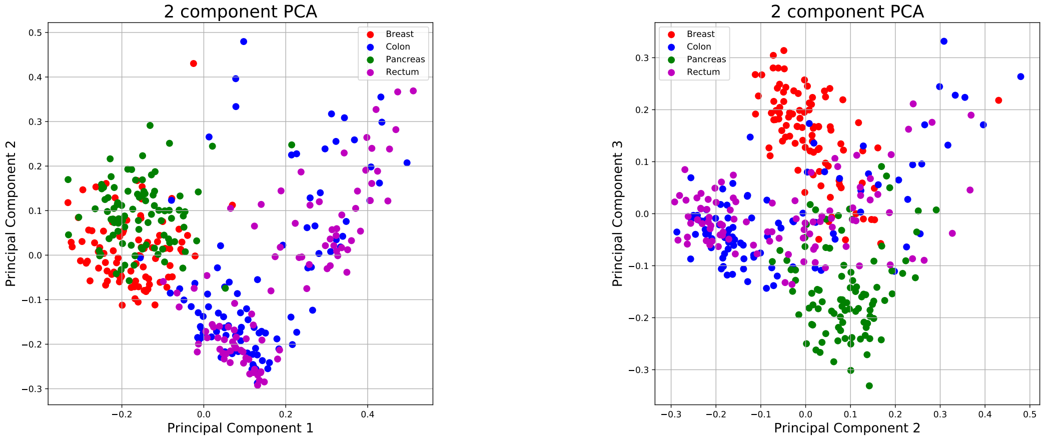
(a) Breast Colon PCA

(b) Colorectal PCA

(c) Colon Pancreas PCA

(d) Breast Pancreas PCA

Figure 2.2: PCA



(a) Breast Pancreas Colon Rectal PCA in 1st and 2nd principle components

(b) Breast Pancreas Colon Rectal PCA in 2nd and 3rd principle components

Figure 2.3: PCA separation of a full dataset.

Chapter 3

Classification Models

3.1 Support Vector Machines (SVM)

Support Vector Machines have become a well established tool within machine learning. They work well in practise and have been used across a wide range of applications from recognising hand written digits, to face identification, and most relevantly for this project, bioinformatics [19]. Conceptually, they have many advantages justifying their popularity. One such advantage of SVMs is their systematic approach, properly motivated by statistical learning theory. Training a SVM involves optimisation of a concave function: there is a unique solution. This contrasts with various other learning paradigms, such as neural network learning, where the underlying model is generally non-convex and we can potentially arrive at different solutions depending on the starting values for the model parameters.

The approach has many other benefits, for example, the constructed model has an explicit dependence on a subset of the data points, the support vectors, which assists model interpretation. Data is stored in the form of kernels which quantify the similarity of dissimilarity of data objects. Kernels can now be constructed for a wide variety of data objects from continuous and discrete input data, through to sequence and graph data. This, and the fact that many of data can be handled within the same model makes the approach very flexible and powerful. The kernel substitution concept is applicable to many other methods for data analysis. Thus SVMs are the most well known of a broad class of methods which use kernels to represent data and can be called kernel based methods [19].

3.1.1 Formalisation of problem to be completed by an SVM

An SVM is an abstract learning machine which learns from a training data set and attempt to generalise and make correct predictions on novel data (test data set).

For the training data, we have a set of input vectors denoted $\mathbf{x}_i$, with each input vector having a number of component features. These input vectors are paired with corresponding labels, which we denote y_i and there are L such pairs ($i = 1, ..., L$).

We consider a binary classification problem with training examples defined for L samples, m features, and $l < L$ training examples l by $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l)$, and test examples defined by $(\mathbf{x}_{l+1}, y_{l+1}), ..., (\mathbf{x}_L, y_L)$, $\mathbf{x}_i \in \mathbb{R}^m$, $y_i \in -1, +1$. We use $\mathbf{X} \in \mathbb{R}^{L \times m}$ to denote the matrix where examples are arranged in rows and $y \in \mathbb{R}^L$ the vector of labels. The matrix $K \in \mathbb{R}^{L \times L}$ denotes the complete kernel matrix containing the kernel values of each training and test data pair.

For this project, the input vectors are samples from the tumours of cancer patients with features consisting of gene expression counts for a large selection of genes. This is outlined in Chapter 2. The labels are typically a binary classification between two cancer types, e.g. Breast and Colon. This is extended to multiple types of cancer using the DAGSVM method as outlined in Section 3.4 [19].

For two classes of well separated data, the learning task amounts to finding a directed hyperplane, that is, an oriented hyperplane in m dimensional space such that on one side will be labelled as $y_i = +1$ and those on the other side as $y_i = -1$.

The directed hyperplane found by a Support Vector Machine is the hyperplane that is maximally distant from the two classes of labelled points located on each side. The closest such points on both sides have the most influence on this separating hyperplane and are therefore called support vectors. The separating hyperplane is given as

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{3.1}$$

where $\cdot$ denotes the scalar product. $b \in \mathbb{R}$ is the bias or offset of the hyperplane from the origin in input space, $\mathbf{x} \in \mathbb{R}^{1 \times l}$ are points located within the hyperplane and the normal to the hyperplane, the weights $\mathbf{w} \in \mathbb{R}^{l \times l}$, determine its orientation.

Of course, most the time, real data is not linearly separable (able to be neatly divided into its individual classes by a single oriented hyperplane, e.g. split into two for a binary classification task), like that in Figure 3.1. The PCA analysis in Figure 2.2a shows that the data points of the breast and colon data are not linearly separable, although the majority of the points are. This is, of course, when the data is reduced to 2 dimensions. Figure 2.2b shows a dataset that is not linearly separable at all. This situation is the motivation for testing different kernels as outlined in Section 3.1.3.

We can also see that stray data points could act as anomalous support vectors with a significant impact on the orientation of the hyperplane: we thus need to have a mechanism for handling noisy and anomalous data points. This is why the L1 error norm is introduced.

3.1.2 SVM for binary classification

The most important ability of a learning machine is that of being able to generalise. The motivation for considering binary classifier SVMs comes from a theoretical upper bound on the generalisation error, that is, the theoretical prediction error when applying the classifier to novel, unseen instances.

This generalisation error bound has two important features:

1. The bound is minimised by maximising the margin, γ, i.e., the minimal distances between the hyperplane separating the two classes and the closest data points to the hyperplane.
2. The bound does not depend on the dimensionality of the space.

The formalisation of the maximisation of this margin by the Lagrange multiplier method is readily available in Campbell et al [19].

It can, however be demonstrated that this minimisation task provides a Primal is equivalent to the maximisation of the Wolfe dual:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{3.2}$$

with respect to the α_i subject to the constraints $\alpha_i > 0$, $\sum_{i=1}^{m} \alpha_i y_i = 0$.

m, y_i and $\mathbf{x}$ are defined previously and α_i, α_j are Lagrange multipliers

This optimisation over α_i by the Kuhn-Tucker theorem gives a solution to the Primal, which provides an optimal $\mathbf{w}$ i.e. a directed hyperplane that separates the data [19].

Interestingly, the dimensionality of $\mathbf{w}$ is the number of features whereas α_i is indexed by the number of samples. This is valuable in practical applications, due to the less computationally intensive nature of optimisation over substantially fewer parameters. For example in this project, in which the number of features is exceeding 60,000, whereas the number of samples does not exceed 200.

Now, this is in the case where the data is linearly separable. When the data is not linearly separable, it may be separable with a non linear separator. This is the reason for trying different, non linear kernels.

3.1.3 Kernel Learning in SVMs

To separate the data with anything other than a linear hyperplane, we notice the datapoint $\mathbf{x}_i$ only appear inside an inner product. To get an alternate representation of the data, we could therefore map the data points into a space with a different dimensionality, called a feature space, through a replacement,

$$\mathbf{x}_i \cdot \mathbf{x}_j \to \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) \tag{3.3}$$

where Φ is the mapping function. The functional form of the mapping $\Phi(\mathbf{x}_i)$ does not need to be known since it is implicitly defined by the choice of kernel,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \tag{3.4}$$

or inner product in feature space.

Several Kernels meet the restrictions implicitly defined by the requirements of the feature space transformation[19]. But the main ones explored in this project are the

- Linear Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
- Homogeneous Polynomial Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$ where $d \in \mathbb{R}$ is the order of the polynomial
- Radial Basis Function (RBF) Kernel (Also known as the Gaussian Kernel): $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}) = exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$ where $\sigma \in \mathbb{R}$ is a kernel parameter and $\gamma = \frac{1}{2\sigma^2}$ for simplicity in later calculation.

Hence for binary classification, with a given choice of kernel, the learning task involves maximisation of

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} K(\mathbf{x}_i, \mathbf{x}_j) \tag{3.5}$$

with respect to α_i subject to $\alpha_i > 0$, $\sum_{i=1}^{m} \alpha_i y_i = 0$.

The bias b of the hyperplane from the origin in input space has not been featured so far, and can be given by

$$b = -\frac{1}{2} \left[\max_{i|y_i=-1} (\sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) + \min_{i|y_i=+1} (\sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right] . \tag{3.6}$$

Thus, to construct the SVM binary classifier, we place the data $(\mathbf{x}_i, y_i)$ into $W(\alpha)$ and maximise subject to it's constraints. From the optimal values of α_i denoted α_i^* we calculate the bias, b.

This means for a novel input vector $\mathbf{z}$, the predicted class is then based on the sign of

$$f(\mathbf{z}) = sign(\sum_{i=1}^{m} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{z}) + b^*) \tag{3.7}$$

where b^* denotes the value of the bias at optimality.

When the maximal margin hyperplane is found in feature space, only those points which lie closest to the hyperplane have $\alpha^* > 0$ and these points are *support vectors*. All other points have $\alpha_i^* = 0$ and the decision function is independent of these samples.

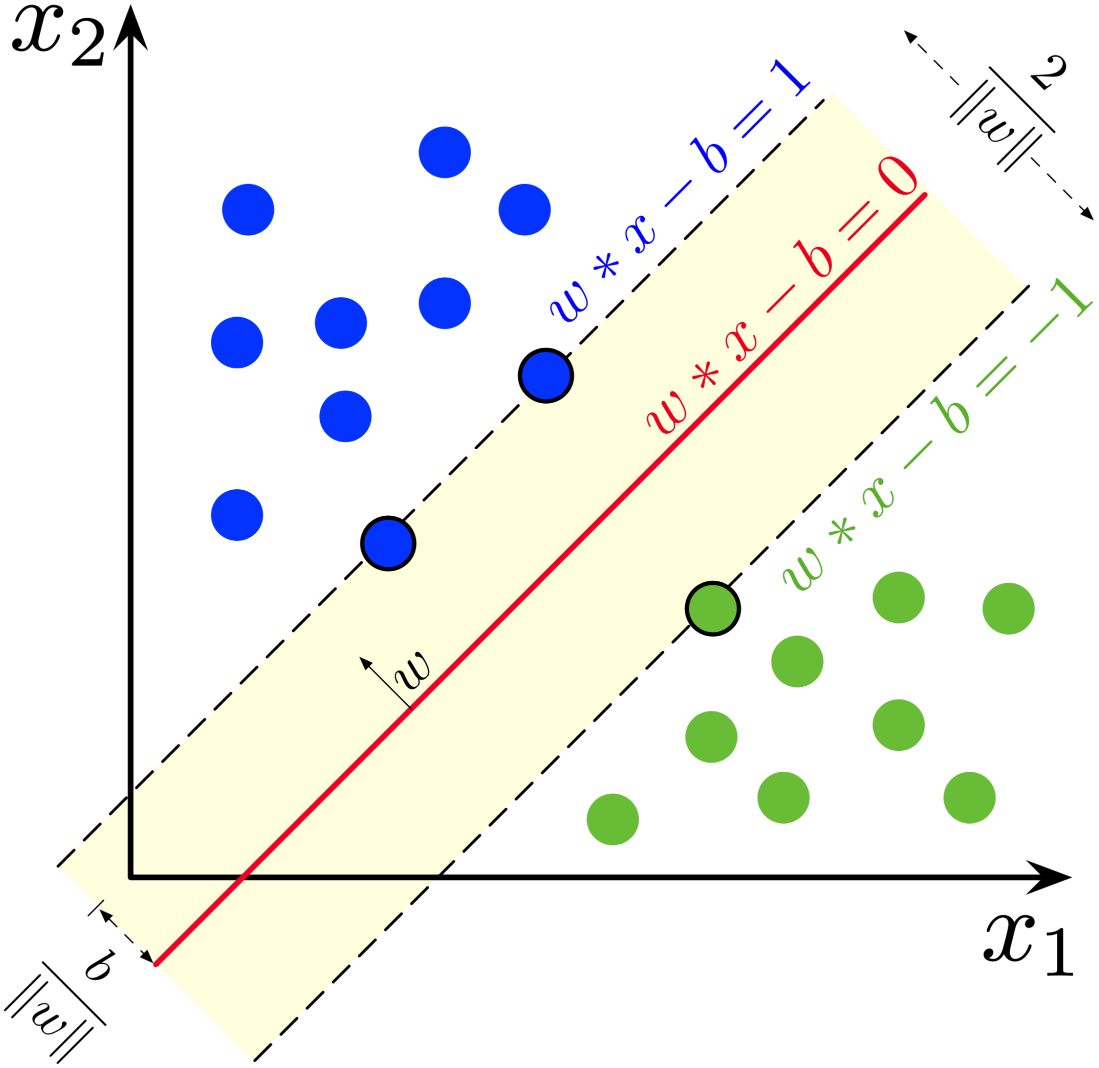


Figure 3.1: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors

This optimisation is suitable for suitable for neatly linearly separable data, but most real data contains noise, leading to poor generalisation.

To avoid this problem, soft margins are introduced with the L_1 error norm.

L_1 error norm is the same as the optimisation problem outlined previously in Equation 3.5 but with the box constraint:

$$0 \leq \alpha_i \leq C \tag{3.8}$$

This implies as $C \to \infty$, the soft margin is equivalent to a hard margin.

The most suitable value of C can be found by means of a validation study, such as by using leave-one-out cross validation, as outlined in Section 3.2.

3.2 Assessment of classifier quality

In order to determine the quality of the classifier that has been trained, the classifier must be tested to see how accurate it is at predicting the label of a sample based on its features.

Attempting to predict the same samples as the classifier has been trained on should, for a high C value and hence a hard margin, give zero error, this value called the training error. This however, is not a good measure of classifier quality, as the main requirement of the classifier is that it is capable of accurately predicting the label of an previously unseen sample. In addition, this training error is not informative of any kind of overfitting. This is when the classifier hyperplane is too closely fitted to the training data, and hence is less representative of an ideal, generalised, separator between classes.

To determine how well the classifier generalises to novel, unseen data, cross validation to find a validation error is performed.

The holdout method is the simplest kind of cross validation. The data set is separated into two sets, called the training set and the testing set. The classifier fits a function using the training set only. Then the classifier is asked to predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is as quick to compute as it is to train a single SVM and count how many correctly predicted samples there are in the test set. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

The leave-one-out procedure consists of removing one example from the training set, constructing the decision function on the basis only of the remaining training data and then testing on the removed example [7]. In this fashion one tests all examples of the training data and measures the fraction of errors over the total number of training examples. This is a very computationally intensive form of cross validation, as L (number of samples) classifiers must be trained separately. It does however provide a very representative value of how generalised an SVM is, as it tests the parameters used and the model, rather than the training/test data split. A LOOCV validation error approaching 0 implies an SVM trained any selection of samples less one can be used to classify the final, unseen, sample and hence is well generalised.

However, due to the fact that the parameters in the various models must be optimised for best performance, both measures will be used. LOOCV will be used to find the optimal parameters for the model and the overall accuracy of the model will be tested on a test set

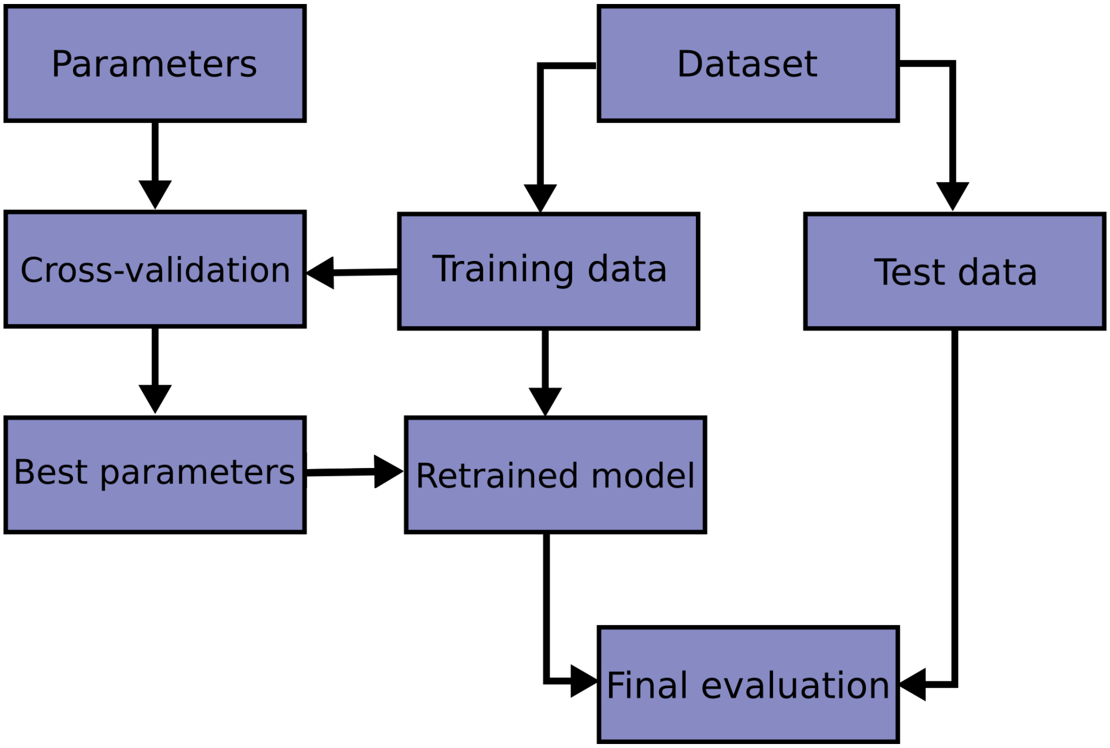


Figure 3.2: Demonstration of the flow of data used.

3.3 Multiple Kernel Learning

An SVM can be considered the base case of multiple kernel learning, in which there is one linear kernel [19].

But there are formulations of learning machines that consist of several kernels that are combined. The reasoning is similar to combining different classifiers: Instead of choosing a single kernel function and putting all our eggs in the same basket, it is better to have a set and let an algorithm do the picking or combination. In this project, different kernels correspond to different notions of similarity and instead of trying to find which works best, a learning method does the picking for us, or may use a combination of them. Using a specific kernel may be a source of bias, and in allowing a learner to choose among a set of kernels, a better solution can be found [20].

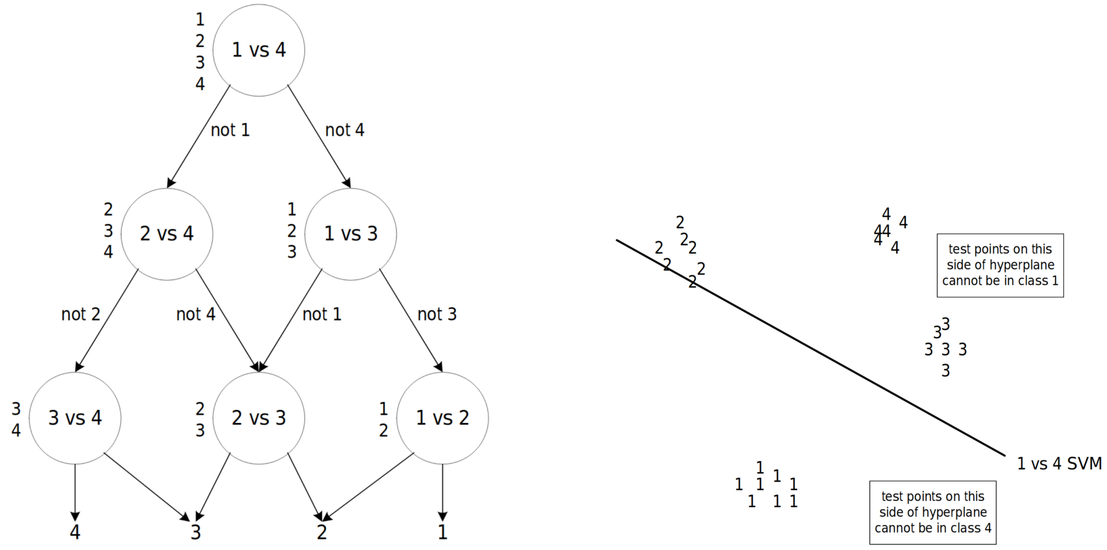
We focus on multiple kernel learning with positive and linear combination parameters, that is, MKL in the form

$$K = \sum_{r=1}^{R} \eta K_r, \eta_r \geq 0. \tag{3.9}$$

This new, combination kernel is then used as any other kernel, in Equation 3.7 [21].

3.4 Multi-Label Classification

The Decision Directed Acyclic Graph (DDAG) is a learning architecture used to combine many two-class classifiers into a multiclass classifier. For an N-class problem, the DDAG contains $N(N-1)/2$ classifiers, one for each pair of classes. The Directed Acyclic Graph SVM (DAGSVM) algorithm combines the results of 1-v-1 SVMs with the choice of the class order in the list (or DDAG) being arbitrary. The DAGSVM Algorithm is proven to be superior to other multiclass SVM algorithms in both training and evaluation time [22].


1 2 3 4
1 vs 4
not 1
not 4
2 3 4
2 vs 4
1 2 3
1 vs 3
not 2
not 4
not 1
not 3
3 4
3 vs 4
2 3
2 vs 3
1 2
1 vs 2
4
3
2
1
test points on this side of hyperplane cannot be in class 1
1 vs 4 SVM
test points on this side of hyperplane cannot be in class 4

(a) The decision DAG for finding the best class out of four classes. The equivalent list state for each node is shown next to that node.

(b) A diagram of the input space of a four-class problem. A 1-v-1 SVM can only exclude one class from consideration.

Figure 3.3: Figure demonstrating tree like structure (Directed Acyclic Graph) of the series of binary classification tasks used to create a multi-class classifier [22].

3.5 Implementation

All SVM models with assorted Kernels were implemented in Python 3.6.9 with the use of MKLpy as a framework, which uses the Scikit-Learn package for preprocessing heavily [23, 24].

The EasyMKL algorithm is used to solve the optimisation problem proposed by the training of the SVMs in this project due to the its simplicity, proven low memory requirements and speed of training [21]. These factors are useful given the use of leave-one-out cross validation requiring training of multiple SVMs of large kernel size in parallel.

3.6 Alternative classifiers

The task outlined in this project is to create the best classifier for CUPs. It is therefore necessary to consider how the SVM and MKL models compare to other, state of the art machine learning models. Two other models were formulated and tested to see how they compare in terms of training and validation error.

3.6.1 AdaBoost

Boosting is an approach to machine learning based on the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules. The AdaBoost algorithm of Freund and Schapire was the first practical boosting algorithm, and remains one of the most widely used and studied, with applications in numerous fields [25].

The standard form of AdaBoost uses decision trees as the weak learners, however it can be any ensemble of weak classifiers, hence it is possible to use an combination of SVMs to form a new classifier [26]. For the purposes of comparability, SVMs are used as the learners when using AdaBoost going forward.

Pseudocode for AdaBoost is shown in Figure 1. Here we are given m labelled training examples $(x_1, y_1), ..., (x_m, y_m)$ where the x_i's are in some domain H, and the labels $y_i \in \{-1, +1\}$. On each round $t = 1, ..., T$, a distribution D_t is computed as in the figure over the m training examples, and a given weak learner or weak learning algorithm is applied to find a weak hypothesis $h_t : H \rightarrow \{-1, +1\}$, where the aim of the weak learner is to find a weak hypothesis with low weighted error $\eta_t$relative to D_t. The final or combined hypothesis H computes the sign of a weighted combination of weak hypotheses $F(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$. This is equivalent to saying that H is computed as a weighted majority vote of the weak hypotheses h_t where each is assigned weight α_t.

Algorithm 1 The boosting algorithm AdaBoost

```
Given: $(x_1, y_1), ..., (x_m, y_m)$ where $x_i \in H$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, ..., m$.
for $t = 1, ..., T$ do
    Train weak learner using distribution $D_t$.
    Get weak hypothesis $h_t : H \rightarrow \{-1, +1\}$.
    Aim: select $h_t$ with low weighted error: $\eta_t = Pr_{i\ D_t}[h_t(x_i) \neq y_i]$.
    Choose $\alpha_t = \frac{1}{2} \ln(\frac{1-\eta_t}{\eta_t})$.
    Update, for $i = 1, ..., m$: $D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ where $Z_t$ is a normalisation
    factor (chosen so that $D_{t+1}$ will be a distribution).
end for
return the final hypothesis: $H(x) = sign(T \sum_{t=1}^{T} \alpha_t h_t(x))$.
```

AdaBoosting can be subject to overfitting if too many rounds of boosting are performed hence the number of boosts must be selected to minimise validation error.

AdaBoost has been implemented using the Scikit-Learn toolbox in Python 3.6.9 [24].

3.6.2 Deep Learners

Deep learning is a highly frequented method for data classification, including in cancer bioinformatics [10]. Deep learning consists of the use of Artificial Neural Networks to solve a given supervised learning problem.

The general algorithm for training a neural network consists randomly initialising the weights (represented in Figure 3.4 as lines between neurons) randomly, then passing a sample through the input layer where the weighted sum at each node is fed forward until the output node receives a value which is then tested for accuracy. If the predicted label is different from the actual label for the sample, the network feeds back corrections to each of the weights in a process called backpropagation. Then another sample is fed through and the cycle continues until all the samples are fed forward, this is a single epoch of training. Training a neural network consists of continued training for many epochs until the network has fallen into some minima in the multidimensional space formed by the feature space, local, or global. At this point, the accuracy has plateaued. One of the flaws of the artificial neural network is its capacity to fall into local minima and be unable to classify further samples correctly [27].

For this project a simple artificial neural network architecture was designed consisting of 2 fully connected layers, the first having 2000 neurons with ReLu activation, the second being 66 neurons also with ReLu activation. This then feeds into an output neuron with binary step activation to give a binary output. This architecture was inspired by the successful model described in recent literature [28].

Due to the time taken to train a single neural network model, LOOCV was not an option for this classifier. As a result, to test for overfitting, validation loss was calculated at each epoch (iteration of backpropagation).

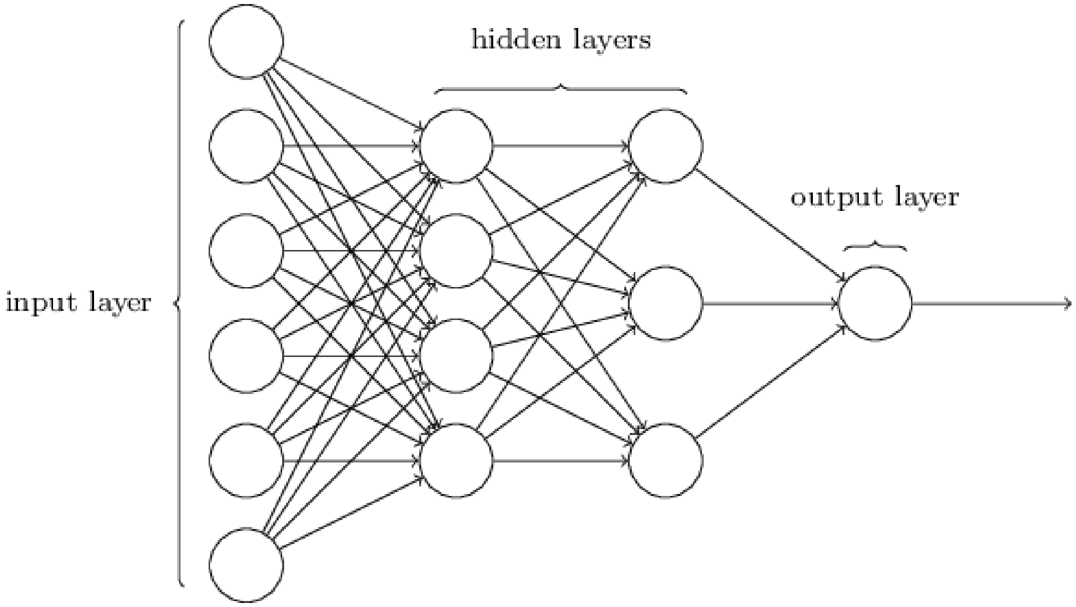


Figure 3.4: General structure of a neural network. For our architecture, the number of neurons, or nodes, in the input layer is the number of genes used as features, with each node being fed the HT-Seq-count of that gene. The number of neurons in the first hidden layer, visible as the first column on the left, is 2000, then the second layer has 66 neurons.

Chapter 4

Results

Outlined below are the results from training and testing the models described in Section 3 with the data, after preprocessing, as described by Section 2. All models were trained on 75% of the data as a training data, with 25% held out for final testing for an accuracy value, as outlined in Section 3.2.

After training, the training accuracy was calculated. This involved performing classification on the data used to train the classifier. While the classifier should be generalised to new data for all the models outlined, they should still be able to classify the data points used to train them accurately. In addition, for the Area under ROC curve value was calculated as another measure of performance for the classifiers.

4.1 Seperable Cancer Classification

The majority of the cancer pairings that were tested, showed extremely high accuracies ($\geq 98\%$) in the SVM model. These accuracies are held as comparisons to the state of the art classification accuracies of around 95% described in Section 1.2. The exclusion to this is the Colon and Rectum classification, which will be discussed in Section 4.2. The data was proven previously to be highly separable as seen from the PCA analysis in Figure 2.3.

The first dataset tested was the Breast and Colon cancer dataset. This dataset is thought to be the most separable, due to literature describing it being fairly unlikely (5%) for a patient to have both breast and a form of colorectal cancer [29]. This implies that the two cancers are fairly distinct and do not easily incite each other. In addition, PCA analysis on the dataset, as seen in Figure 2.2a, displays a visual representation of the separability of the dataset, and appeared to be separable due to the distinct formation of clusters in the feature space. This is is taken to imply that in the feature space, different cancers have higher HTSeq-counts for certain genes. This is supported by literature [30].

Similarly tested are the Pancreas/Breast and Colon/Pancreas datasets, which have similar separability, as shown in Figure 2.2d and Figure 2.2c respectively. Pancreas and Breast cancers are similarly unlikely (5%) to incite each other in literature [31]. The Colon and Pancreas are both parts of the digestive system, and were initially hypothesised to have similar HTSeq-counts, and therefore are more difficult to distinguish [32]. These two cancers show the lowest accuracy of the SVM, but are still a high value.

4.2 Colon and Rectum Cancer Classification

This was the most ambitious dataset tested. Colon and Rectum cancers are often grouped together as a subclass of cancer. Because of the anatomic continuity of the colon into the rectum, cancers affecting these organs have historically been considered equivalent [33].

It is worth testing how well the classifiers perform on a very similar cancer.

4.3 Multi-Label Classification

When all the datasets are concatenated to see how the classifiers utilise the DDAG method outlined in Section 3.4 it is interesting to see that the deep learner gets the highest training accuracy substantially above the other classifiers. It is possible that with the neural networks highly nonlinear higher dimensional weights, it has found some other metric by which to seperate the data.

Model	Train acc	Validation acc	Balanced acc
SVM	1	0.878136	0.739872
MKL	1	0.874552	0.748206
AdaBoost	0.810036	N/A	0.752688
Deep Learn	0.953405	N/A	0.827957

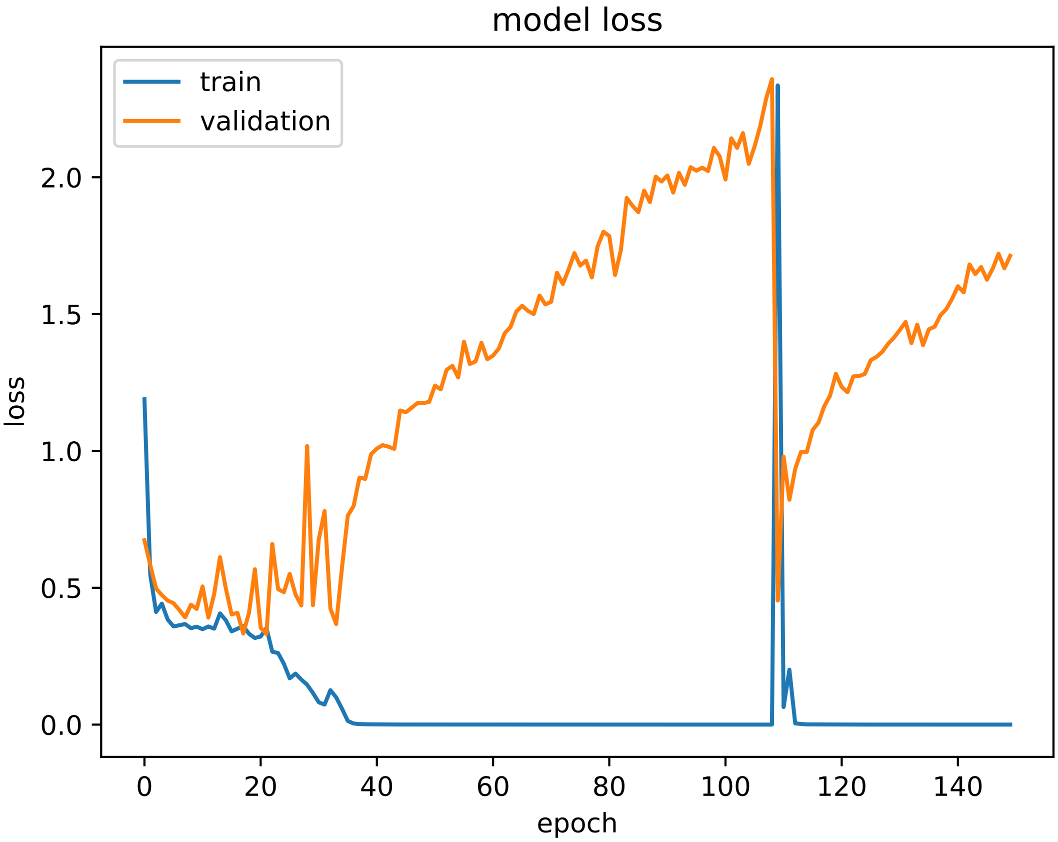


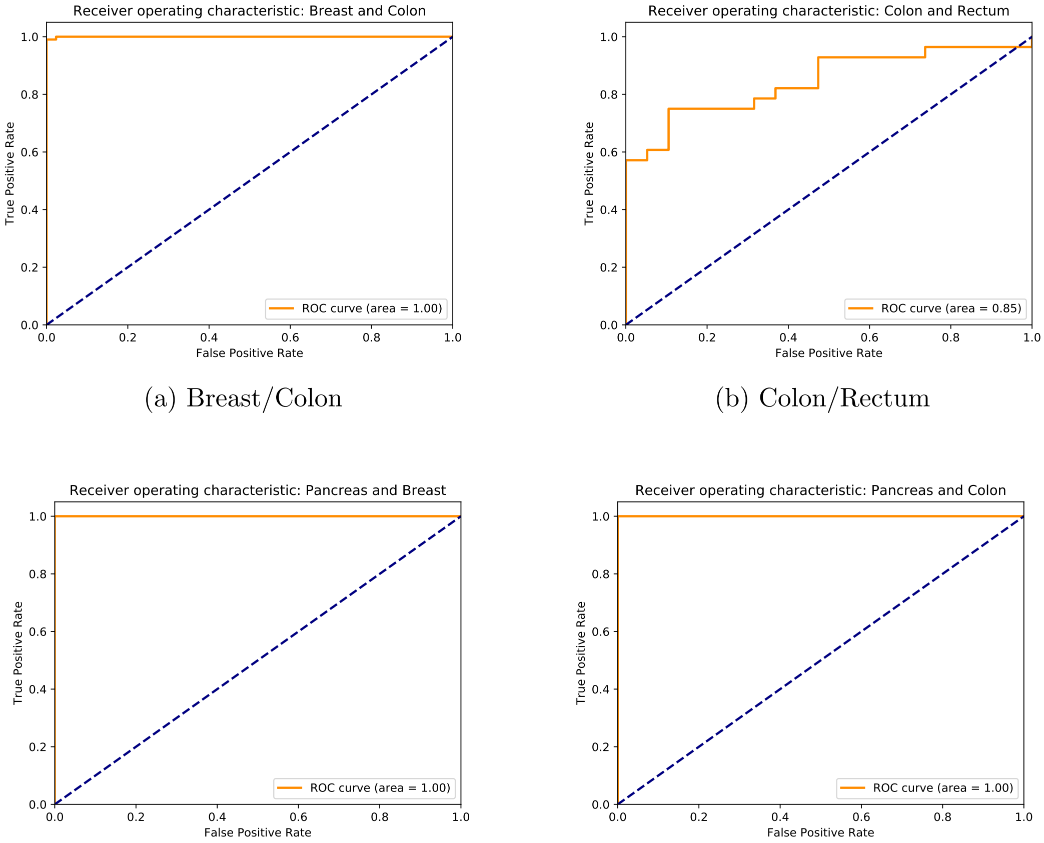
Figure 4.1: Training and validation loss for Deep Neural Network. Lower values are better so it shows there is likely some overfitting to the training data.

4.4 Overall

The results of all tests are described in Table 4.1. For the MKL classifier, a combination of a linear kernel and a RBF (gaussian) kernel where $\gamma = 1$

Data	*Model*	*Train Acc*	*LOOCV Acc*	*Balanced Acc*	*ROC AUC*	*Parameters*
Breast/Colon	SVM	1	0.998551	0.99	1	$C = 1$
Colon/Rectum	SVM	1	0.669065	0.77	0.846	$C = 100$
Pancreas/Breast	SVM	1	0.989011	1	1	$C = 1$
Colon/Pancreas	SVM	1	0.989011	0.988889	1	$C = 100$
Breast/Colon	MKL	1	1	0.995146	0.999697	$\gamma = 1$, $C = 100$
Colon/Rectum	MKL	1	0.726619	0.708	0.722	$\gamma = 1$, $C = 100$
Pancreas/Breast	MKL	0.992701	0.992674	1	1	$\gamma = 1$, $C = 1$
Colon/Pancreas	MKL	1	0.992674	0.988889	1	$\gamma = 1$, $C = 100$
Breast/Colon	AdaBoost	1	N/A	0.987013	N/A	
Colon/Rectum	AdaBoost	1	N/A	0.723404	N/A	
Pancreas/Breast	AdaBoost	1	N/A	1	N/A	
Colon/Pancreas	AdaBoost	1	N/A	1	N/A	
Breast/Colon	Deep Learner	1	N/A	0.995671	N/A	$Epoch = 150$
Colon/Rectum	Deep Learner	0.94964	N/A	0.617021	N/A	$Epoch = 15$
Pancreas/Breast	Deep Learner	1	N/A	0.989011	N/A	$Epoch = 150$
Colon/Pancreas	Deep Learner	0.996337	N/A	0.989011	N/A	$Epoch = 150$

Table 4.1: Results of all classifiers on various pairings of data.



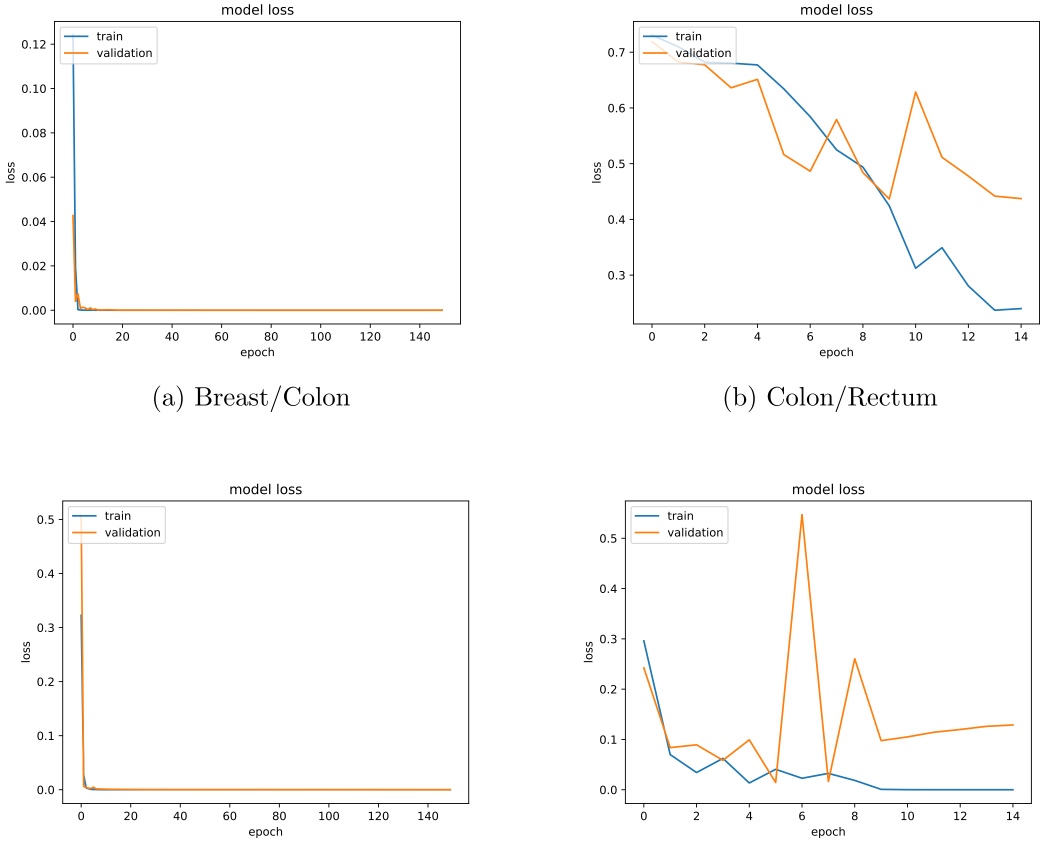
(a) Breast/Colon

(b) Colon/Rectum

(c) Pancreas/Breast

(d) Pancreas/Colon

Figure 4.2: SVM ROCs



(a) Breast/Colon

(b) Colon/Rectum

(c) Pancreas/Breast

(d) Pancreas/Colon

Figure 4.3: Training and Validation loss curves for each of the neural networks trained. Colon/Rectum and Pancreas/Colon featured lowest validation loss (desirable) at 15 epochs. After that point, data began overfitting to training data, and validation data dropped

Chapter 5

Conclusion

In this project, several classifier are described, subsequently trained and tested to classify a series of binary Gene Expression data sets.

These classifiers are shown to be highly accurate with little distinguishing features between their accuracies. This is a reflection on the separability of the dataset, rather than the efficacy of the classifiers outlined.

5.1 Future Work

The feature spaces the classifiers are trained on, after removing zeroes, is 14,398, which is reduced from 60,484 genes. This dramatic reduction implies a huge sparsity in the feature space. In addition, literature shows that reducing the feature space by many genes can lead to increases in accuracy [7]. As a result, more pruning of the feature space should be explored to increase accuracy. Feature space reduction methods could include using the previous utilised PCA as a feature space rather than a visualisation tool, this is common in input dimension reduction in gene expression analysis [34].

One of the most logical next steps with the classifiers that have been outlined here is to apply them to samples that were previously Carcinoma of Unknown Primary (CUP) samples, but the primary is now known. In testing on this dataset, the true accuracy of the classifier will be able to be viewed. Previous studies have shown that testing real CUP samples will often have a lower accuracy (likelihood of correct classification) than non CUP tumour samples [17]. It would be valuable to see how this accuracy still compares to the prediction accuracies from the methods in current circulation, rated as low as 35% [17].

Bibliography

[1] Patricia L. Kandalaft and Allen M. Gown. "Practical Applications in Immunohistochemistry: Carcinomas of Unknown Primary Site". In: *Archives of Pathology & Laboratory Medicine* 140.6 (June 2016), pp. 508–523. ISSN: 0003-9985. DOI: 10.5858/arpa.2015-0173-CP. URL: http://www.archivesofpathology.org/doi/10.5858/arpa.2015-0173-CP.

[2] John D. Hainsworth and F. Anthony Greco. *Gene expression profiling in patients with carcinoma of unknown primary site: From translational research to standard of care*. Apr. 2014. DOI: 10.1007/s00428-014-1545-2.

[3] Paolo Boscolo-Rizzo et al. *The prevalence of human papillomavirus in squamous cell carcinoma of unknown primary site metastatic to neck lymph nodes: a systematic review*. Dec. 2015. DOI: 10.1007/s10585-015-9744-z.

[4] Panagiota Economopoulou et al. *Cancer of Unknown Primary origin in the genomic era: Elucidating the dark box of cancer*. July 2015. DOI: 10.1016/j.ctrv.2015.05.010.

[5] Panagiota Economopoulou and George Pentheroudakis. *Cancer of unknown primary: time to put the pieces of the puzzle together?* Oct. 2016. DOI: 10.1016/S1470-2045(16)30377-1.

[6] Ciprian Tomuleasa et al. "How to diagnose and treat a cancer of unknown primary site". In: *Journal of Gastrointestinal and Liver Diseases* 26.1 (Mar. 2017). ISSN: 18418724. DOI: 10.15403/jgld.2014.1121.261.haz. URL: http://www.jgld.ro/wp/archive/y2017/n1/a13.

[7] Isabelle Guyon et al. "Gene Selection for Cancer Classification using Support Vector Machines". In: *Machine Learning* 46.1/3 (2002), pp. 389–422. ISSN: 08856125. DOI: 10.1023/A:1012487302797. URL: http://link.springer.com/10.1023/A:1012487302797.

[8] Richard W. Tothill et al. “An Expression-Based Site of Origin Diagnostic Method Designed for Clinical Application to Cancer of Unknown Origin”. In: *Cancer Research* 65.10 (May 2005), pp. 4031–4040. ISSN: 0008-5472. DOI: 10.1158/0008-5472.CAN-04-3617. URL: http://cancerres.aacrjournals.org/lookup/doi/10.1158/0008-5472.CAN-04-3617.

[9] Kalle A Ojala, Sami K Kilpinen, and Olli P Kallioniemi. “Classification of unknown primary tumors with a data-driven method based on a large microarray reference database”. In: *Genome Medicine* 3.9 (Oct. 2011), p. 63. ISSN: 1756-994X. DOI: 10.1186/gm279. URL: http://www.ncbi.nlm.nih.gov/pubmed/21955394%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3239238%20http://genomemedicine.biomedcentral.com/articles/10.1186/gm279.

[10] Boyu Lyu and Anamul Haque. “Deep Learning Based Tumor Type Classification Using Gene Expression Data”. In: (2018). DOI: 10.1145/nnnnnnn.nnnnnnn. URL: https://www.researchgate.net/publication/327217294_Deep_Learning_Based_Tumor_Type_Classification_Using_Gene_Expression_Data.

[11] William F. Flynn et al. “Pan-cancer machine learning predictors of primary site of origin and molecular subtype”. In: *bioRxiv* (July 2018), p. 333914. DOI: 10.1101/333914. URL: https://www.biorxiv.org/content/10.1101/333914v2.

[12] S. Anders, P. T. Pyl, and W. Huber. “HTSeq–a Python framework to work with high-throughput sequencing data”. In: *Bioinformatics* 31.2 (Jan. 2015), pp. 166–169. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu638. URL: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btu638.

[13] Laura Huerta and Melissa Burke. “Functional genomics (II): Common technologies and data analysis methods”. In: *Embl-Ebi* Ii (2016), pp. 1–34. URL: http://www.ebi.ac.uk/training/online/course/functional-genomics-ii-common-technologies-and-data-analysis-methods.

[14] Liyan Gao et al. “Length bias correction for RNA-seq data in gene set analyses”. In: *Bioinformatics* 27.5 (Mar. 2011), pp. 662–669. ISSN: 13674803. DOI: 10.1093/bioinformatics/btr005.

[15] Malachi Griffith et al. “Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud”. In: *PLoS Computational Biology* 11.8 (Aug. 2015). ISSN: 15537358. DOI: 10.1371/journal.pcbi.1004393.

[16] Robert L. Grossman et al. “Toward a Shared Vision for Cancer Genomic Data”. In: *New England Journal of Medicine* 375.12 (Sept. 2016), pp. 1109–1112. ISSN:

0028-4793. DOI: 10.1056/NEJMp1607591. URL: http://www.ncbi.nlm.nih.gov/pubmed/27653561%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6309165%20http://www.nejm.org/doi/10.1056/NEJMp1607591.

[17] Richard W. Tothill et al. "Development and validation of a gene expression tumour classifier for cancer of unknown primary". In: *Pathology* 47.1 (Jan. 2015), pp. 7–12. ISSN: 00313025. DOI: 10.1097/PAT.0000000000000194. URL: https://linkinghub.elsevier.com/retrieve/pii/S003130251630160X.

[18] Sarah E. Hunt et al. "Ensembl variation resources". In: *Database : the journal of biological databases and curation* 2018 (Jan. 2018). ISSN: 17580463. DOI: 10.1093/database/bay119.

[19] Colin. Campbell and Yiming. Ying. *Learning with support vector machines.* Morgan & Claypool, 2011, p. 83. ISBN: 1608456161.

[20] Mehmet Gönen and Ethem Alpaydın. "Multiple Kernel Learning Algorithms". In: *Journal of Machine Learning Research* 12.Jul (2011), pp. 2211–2268. ISSN: ISSN 1533-7928. URL: http://www.jmlr.org/papers/v12/gonen11a.html.

[21] Fabio Aiolli and Michele Donini. "EasyMKL: a scalable multiple kernel learning algorithm". In: *Neurocomputing* 169 (Dec. 2015), pp. 215–224. ISSN: 0925-2312. DOI: 10.1016/J.NEUCOM.2014.11.078. URL: https://www.sciencedirect.com/science/article/pii/S0925231215003653.

[22] John C Platt, Nello Cristianini, and John Shawe-Taylor. "Large margin DAGs for multiclass classification". In: *Advances in Neural Information Processing Systems.* 2000, pp. 547–553. ISBN: 0262194503.

[23] Ivano Lauriola. *IvanoLauriola/MKLpy: A package for Multiple Kernel Learning in Python.* URL: https://github.com/IvanoLauriola/MKLpy.

[24] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830. ISSN: ISSN 1533-7928. URL: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

[25] Robert E. Schapire. "Explaining adaboost". In: *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik* (2013), pp. 37–52. DOI: 10.1007/978-3-642-41136-6{_}5.

[26] Elkin García and Fernando Lozano. "Boosting Support Vector Machines". In: *Revista de Ingeniería* unknown.24 (2006), pp. 62–70. ISSN: 0121-4993. DOI: 10.16924/riua.v0i24.328.

[27] Wojciech Marian Czarnecki and Razvan Pascanu DeepMind London. *LOCAL MINIMA IN TRAINING OF NEURAL NETWORKS.* Tech. rep.

[28] Jasleen K. Grewal et al. "Application of a Neural Network Whole Transcriptome–Based Pan-Cancer Method for Diagnosis of Primary and Metastatic Cancers". In: *JAMA Network Open* 2.4 (Apr. 2019), e192597. ISSN: 2574-3805. DOI: 10.1001/jamanetworkopen.2019.2597. URL: http://jamanetworkopen.jamanetwork.com/article.aspx?doi=10.1001/jamanetworkopen.2019.2597.

[29] S Toma et al. "Association between breast and colorectal cancer in a sample of surgical patients." In: *European journal of surgical oncology : the journal of the European Society of Surgical Oncology and the British Association of Surgical Oncology* 13.5 (Oct. 1987), pp. 429–32. ISSN: 0748-7983. URL: http://www.ncbi.nlm.nih.gov/pubmed/3666159.

[30] Jayne L Dennis and Karin A Oien. "Hunting the primary: novel strategies for defining the origin of tumours." In: *The Journal of pathology* 205.2 (Jan. 2005), pp. 236–47. ISSN: 0022-3417. DOI: 10.1002/path.1702. URL: http://www.ncbi.nlm.nih.gov/pubmed/15641019.

[31] Stefano Amore Bonapasta et al. "Metastasis to the pancreas from breast cancer: Difficulties in diagnosis and controversies in treatment". In: *Breast Care* 5.3 (June 2010), pp. 170–173. ISSN: 16613791. DOI: 10.1159/000314249.

[32] Meghan L. Underhill, Katharine A. Germansky, and Matthew B. Yurgelun. *Advances in Hereditary Colorectal and Pancreatic Cancers.* July 2016. DOI: 10.1016/j.clinthera.2016.03.017.

[33] Theodore S. Hong, Jeffrey W. Clark, and Kevin M. Haigis. "Cancers of the colon and rectum: Identical or Fraternal Twins?" In: *Cancer Discovery* 2.2 (Feb. 2012), pp. 117–121. ISSN: 21598274. DOI: 10.1158/2159-8290.CD-11-0315.

[34] Feng Chu and Lipo Wang. *APPLICATIONS OF SUPPORT VECTOR MACHINES TO CANCER CLASSIFICATION WITH MICROARRAY DATA*. Tech. rep. 6. 2005, pp. 475–484. URL: http://research.nhgri.nih.gov/microarray/.