Bioinformatics - Final Project Report

Introduction

From a data analysis perspective, the dataset can be characterized by its large amount of features compared to sample size ($p \gg n$) as well as the heterogeneity of its labels (more than 130 diseases, often labeled according to different taxonomies). Our approach for exploring the data was to first try to understand the data at a high level and then look at more nuanced differences:

- At a high level, can a classifier be trained to distinguish between samples taken from *healthy* and *unhealthy* subjects? This is an important task because it can be used to screen patients for early signs of any of the recorded illnesses by screening for gene interactions within certain genes.
- Then, we took a closer look at the gene expression in *specific types of cancer*, wishing to isolate the ~100 features from the dataset that can explain differences between different types of breast cancer and leukemia. Cancer research is one of the most important research areas in the medical sciences. A correct prediction of different tumor types has noticeable value in providing better treatment and toxicity minimizationon the patients.

Methodology

Overall structure of the data

Traditionally, the PCA procedure is employed to in order to reduce dimensionality and hence enable visualization of the data. However, it has been shown that the sample covariance matrix of a $p \gg n$ sample is a very poor approximation of the original population covariance, due to its high collinearity. More specifically, such covariance matrices are known as *spiked*, due to the fact that the first few components tend to explain a large proportion of the overall variance (Paul (2007)). This can be seen in the Scree Plot [Fig. 1]. The 2D and 3D PCA plots [Figs. 2 3] are characteristic of spiked covariance matrices.

An approach to obtain a more realistic approximation to the population covariance eigenvectors has been proposed as Sparse PCA (Zou, Hastie, and Tibshirani (2006)). sPCA is the same as PCA under the sparsity constraint that the resulting eigenvectors may only have k non-zero values. Unfortunately, sPCA is known to be NP-hard, and even approximation algorithms are computationally expensive or unfeasible. The sPCA of the data for $k = 10$ is shown in Fig. 4. It is a better visualization of the data in that it reflects the heterogeneity of the samples much better than PCA.

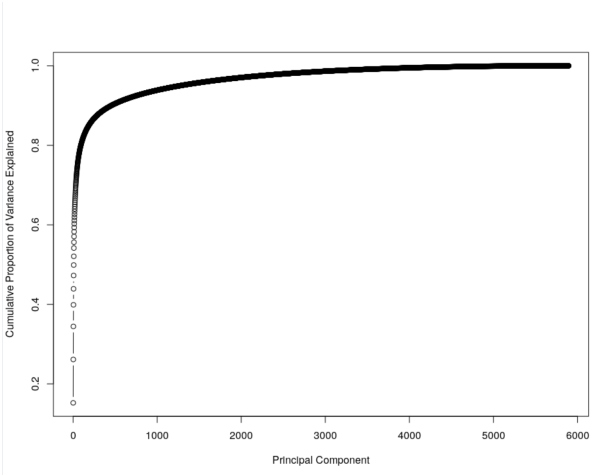

Figure 1: Scree plot reveals most components are highly correlated. The first 1271 components explain 95% of variance in the data

A second reason why PCA is of limited value as a visualization tool in this situation is because a large amount of information encoded within DNA is of no relevance to the labels. Hence, even if we had a larger sample size and some guarantee of PCA's stability, it is still possible that not much could be learnt from PCA visualization. Hence it is no surprise that the PCA plots reveal the data to be highly tangled.

Building a classifier for Healthy / Disease

Nevertheless, PCA is still of use as a way to reduce the dimensionality of the sample's input space. Other techniques can also be used, such as autoencoders and, more recently, Random Forests (Kong and Yu (2018)); however we chose to use PCA due to theoretical guarantees that information from the sample covariance is reliably captured as well as computational constraints.

The R programming language was used to compute the parameters for several classic Machine Learning

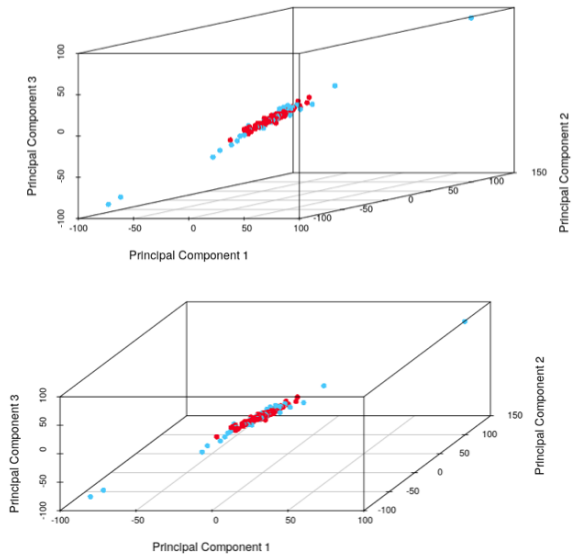

Figure 2: 3D scatterplots of the inputs in PC space using the first 3 components from different angles. Blue = healthy; red= unhealthy. The first 3 components capture 34% of the data. Unfortunately, due to spiking, PCA is not a suitable technique for understanding the data structure.

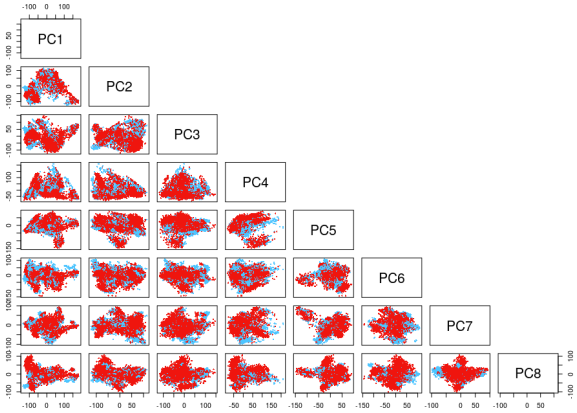

Figure 3: Pair scatterplots of the first 8 components (which explain 52% of the variance). Blue = healthy; red= unhealthy.

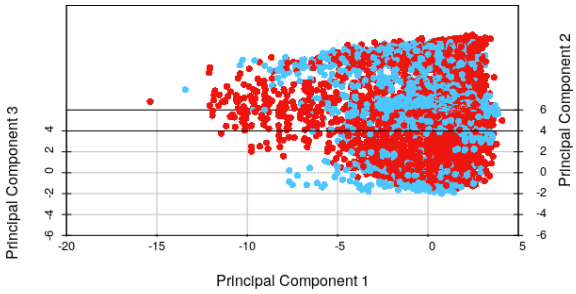

Figure 4: Sparse PCA projection on $k = 10$ vectors is a more realistic visualization of the heterogeneicity of the data.

algorithms. The R programming language was chosen for this task since all algorithms discussed had an implementation by their creators or affiliated researchers in this language. The Caret framework was used, which wraps most of R's ML algorithms to derive the desired statistics using 5-fold, 2-repetition CV. CV was preferred over bootstraping due to the small n and unbalanced classes. Due to the $p \gg n$ nature of the data, all fitting was done on the PCA feature space, with p treated as a parameter.

There is no consensus on a good heuristic to select the number of PC's to use as feature space. Based on past experience, a reasonable approach is to first be generous in our choice of p, by capturing 95% of the variance, then do a search on the 90-95% cumulative variance range on the most promising algorithms in order to try and reduce the total error by further reducing dimensionality. This is not a perfect heuristic, since some algorithms perform significantly better in lower dimensions (k-NN being the canonical example), but it is nevertheless a popular heuristic. Ideally, a search would be done on a wide range of p values, but this is unrealistic due to computational limitations and in practice is seldom done.

Model selection was done using the kappa value. This is the only reasonable choice since accuracy is blind to class imbalances; consider an algorithm that simply labeled all samples as unhealthy, it would be 67% accurate. Furthermore, an algorithm would only need to label 1/3 of the healthy individuals correctly to achieve ~80% accuracy. The kappa parameter, however, penalises this by comparing with random classifier, and is hence preferred (Hastie, Tibshirani, and Friedman (2001)) when dealing with class imbalances.

Deep Learning was also used to learn a non-linear classification surface. The Keras framework for Python was used for this task, also using 5-fold 2-repetition CV. It surpassed in accuracy all other classifiers. The network structure was as follows: 1271 PCA dimensions in, then ReLU layers of 640, 370, 3 and 1 components.

Building a classifier to distinguish between different types of tumours

Given the shape of the data ($n \approx 100, p > 22,000$), many traditional ML techniques are known to fail. Both ANN's and Random Forests are prone to overfitting with n small. The spiked covariance phenomenon is even more pronounced in data of these characteristics.

In practice, Partial Least Squares is often used (Pérez-Enciso and Tenenhaus (2003), Zhang et al. (2012), Musumarra et al. (2005)). This is a classic technique which made its way into omics data analysis from the field of chemometrics (to which is bears close ties). While initially created for regression tasks, it is simple to adapt by encoding classes as 0/1 in the binary classification case and using the learnt regression hyperplane as a classification surface. The multiclass case is handled through hot encoding. Under the PLS model, the variates and the target are assumed to be of the form:

$$X = TP^T + EY = UQ^T + E$$

And different algorithms exist to estimate T, P, U, Q, most of which are in the EM-family.

One advantage of PLSDA is that adding sparsity constraints is reasonably trivial, as each predictor is given a weight which can be interpretted as its contribution to explaining variance. Hence sPLSDA was proposed for reasons similar to its PCA counterpart.

Tuning was done following the Balanced Error Rate (BER).

Results & Discussion

Healthy / Cancer / Other Illness

Table 1 summarises the results for both classic ML and Deep Learning algorithms. Although traditional algorithms worked well, Deep Learning gave the best results. This is in line with the hypothesis that the classification surface for microarray data is non-linear.

Table 1: Results using some classic supervised learning algorithms on the PC decomposition. $p = 1271$, capturing 95% of variance.

Algorithm	Parameters	Accuracy (standard deviation)	Kappa
Logit (elastic)	$\lambda = 0.01$ $\alpha = 0.2$	0.942 (0.008)	0.866
LDA		0.941 (0.009)	0.866
PLDA	$\ell_1 =$ 0.001 $C = 1$	0.92 (0.01)	0.813
QDA		0.859 (0.01)	0.649
SWDA	$\ell_1 =$ 0.003267 $\ell_2 = 0.1$	0.943 (0.01)	0.866
PLSDA	ncomp = 20	0.944 (0.008)	0.867
SVM RBF kernel	$\sigma =$ 0.000265 $C = 4.$	0.937 (0.005)	0.854
SVM Linear kernel		0.874 (0.008)	0.72
Deep Learning		**0.995**	-

Gene Expression of Different Types of Tumours

Two classifiers were built using PLSDA and sPLSDA to classify different types of Leukemias and different types of breast cancer, emulating the procedures in [cite someone]. While sPLSDA was of great use in building faithful visualizations of the data as well as highlighting which genes interaction are responsible for the most variation between samples of different classes (Figs. 6 and 8). Specifically, the Leukemia data is accurately represented with chronic and acute cases correctly separated, and types known to be similar closer in the plane than those to which they bear less relationship.

Overall, PLSDA performed very well in this classifier, specifically in the Leukemia case where it managed to classify all cases except for one correctly (see Table 2). The CIM plots show which sites were given the most weight by sPLSDA and can be used by researchers to

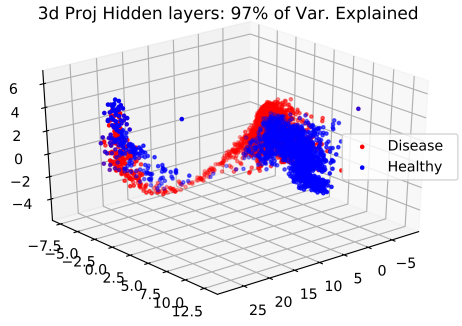

Figure 5: Projection of data using the last layer of Deep Learning algorithm

further investigate gene interactions from the cDNA that was bound at these sites.

Table 2: Results of PLSDA and sPLSDA on breast tumour and leukemia samples.

Data subsample	PLSDA (BER)	sPLSDA (BER)
Breast tumour	0.0966	0.125
Leukemia	0.00134	0.164

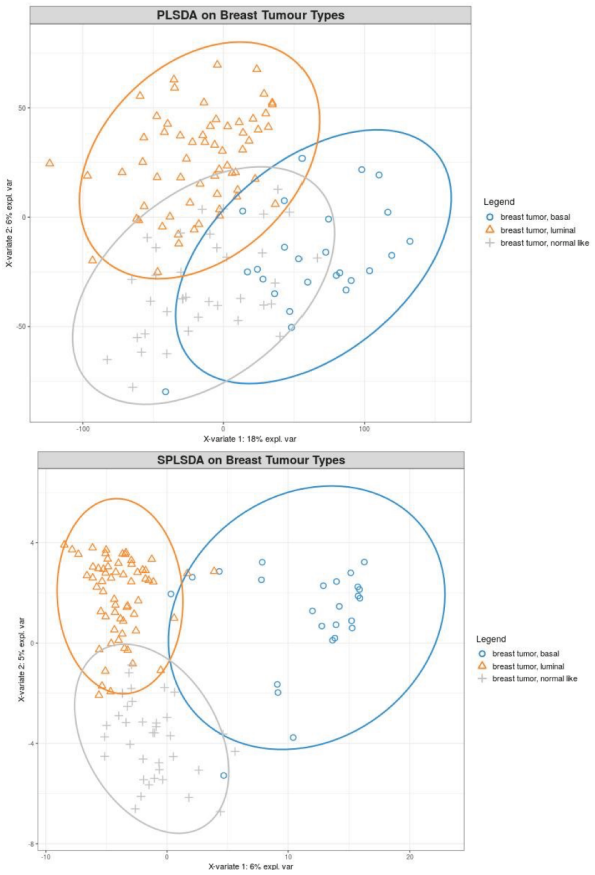

Figure 6: PLSDA (top) and sPLSDA (bottom) projections of the Breast Tumours data

Appendix: Tuning Plots

Tuning plots are provided for completion

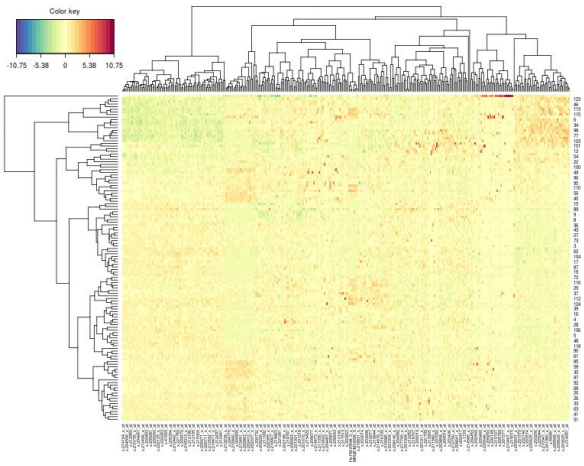

Figure 7: Clustered Image Map showing the most active sites in Breast Tumour classification

Bibliography

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Kong, Yunchuan, and Tianwei Yu. 2018. "A Deep Neural Network Model Using Random Forest to Extract Feature Representation for Gene Expression Data Classification." *Scientific Reports* 8 (1): 16477.

Musumarra, Giuseppe, Vincenza Barresi, Daniele F Condorelli, Cosimo G Fortuna, and Salvatore Scirè. 2005. "Genome-Based Identification of Diagnostic Molecular Markers for Human Lung Carcinomas by Pls-Da." *Computational Biology and Chemistry* 29 (3): 183–95.

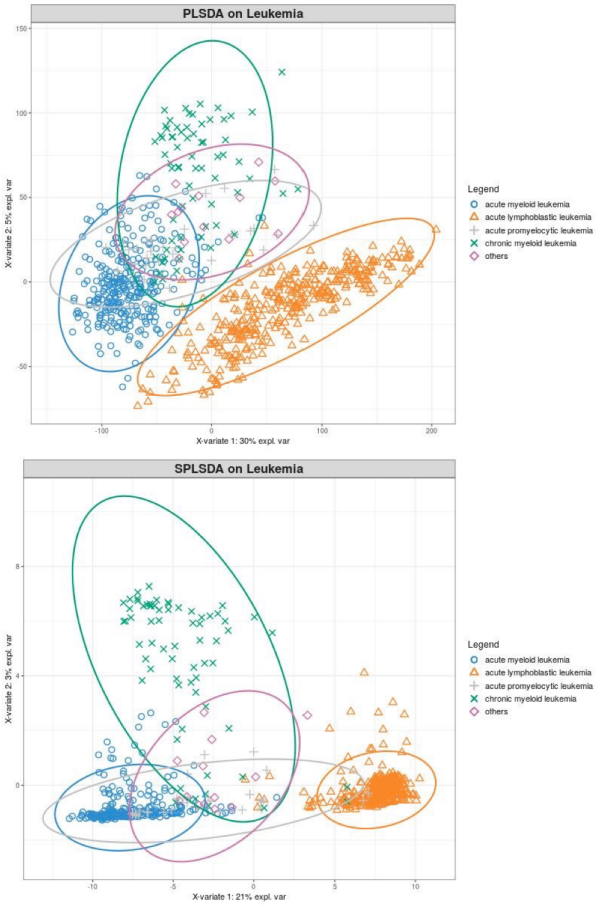

Figure 8: PLSDA (top) and sPLSDA (bottom) projections of the Leukemia data

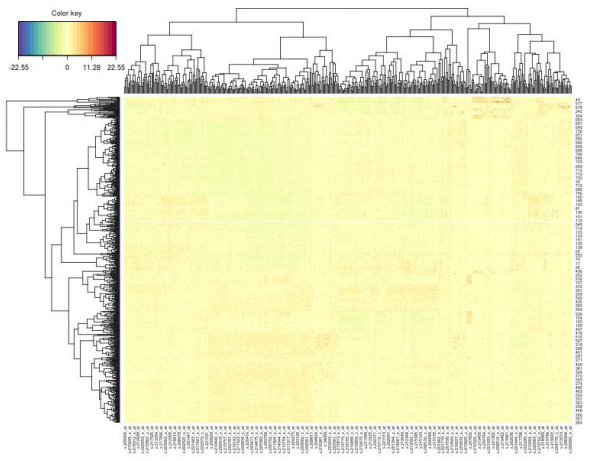

Figure 9: Clustered Image Map showing the 100 most active sites in Leukemia classification (obtained using sPLSDA)

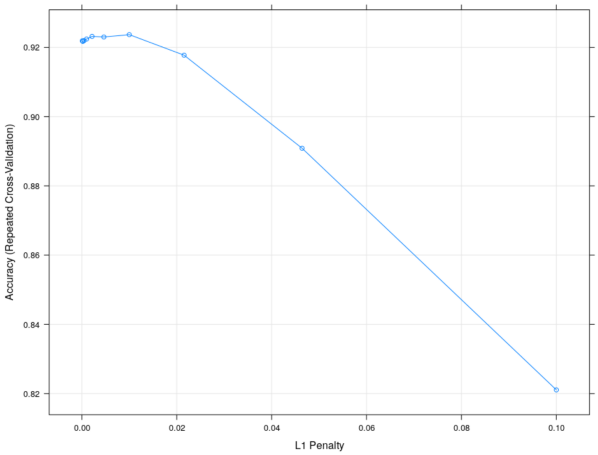

Figure 10: Penalized LDA tuning plot.

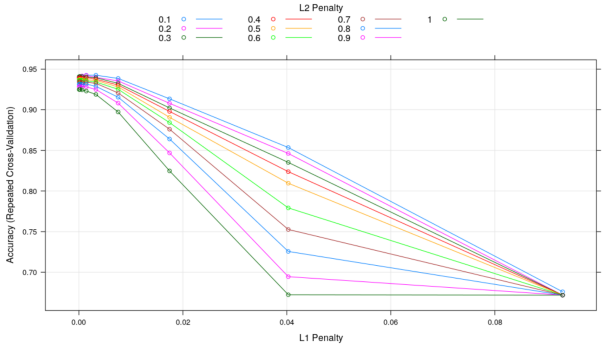

Figure 11: SWDA tuning plot.

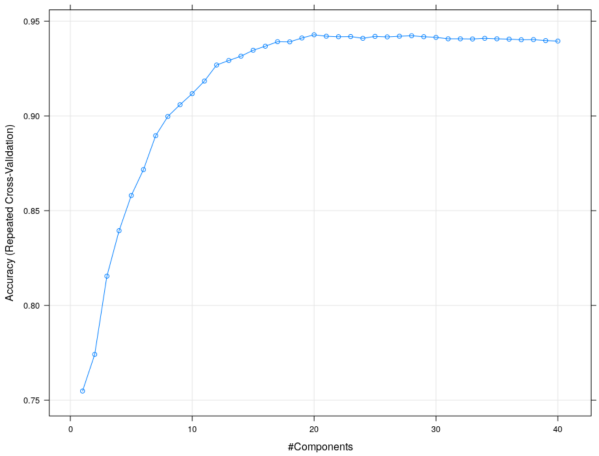

Figure 12: Partial Least Squares Discriminant Analysis tuning plot.

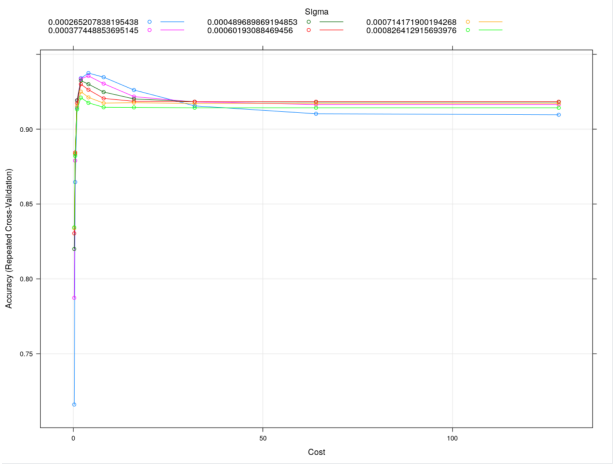

Figure 13: SVM Radial Basis Function tuning plot.

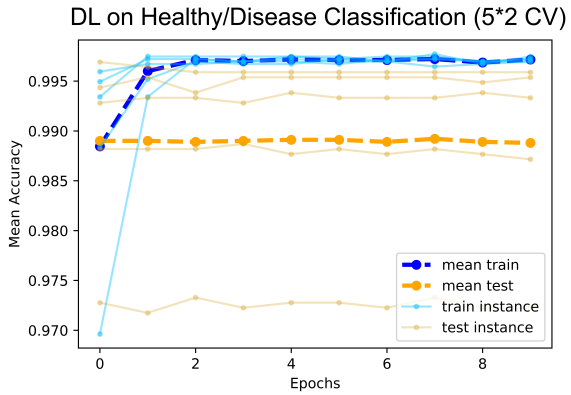

Figure 14: Deep Learning Tuning Plot.

Paul, Debashis. 2007. "Asymptotics of Sample Eigenstructure for a Large Dimensional Spiked Covariance Model." *Statistica Sinica*, 1617–42.

Pérez-Enciso, Miguel, and Michel Tenenhaus. 2003. "Prediction of Clinical Outcome with Microarray Data: A Partial Least Squares Discriminant Analysis (Pls-Da) Approach." *Human Genetics* 112 (5-6): 581–92.

Zhang, Tao, Xiaoyan Wu, Mingzhu Yin, Lijun Fan, Haiyu Zhang, Falin Zhao, Wang Zhang, et al. 2012. "Discrimination Between Malignant and Benign Ovarian Tumors by Plasma Metabolomic Profiling Using Ultra Performance Liquid Chromatography/Mass Spectrometry." *Clinica Chimica Acta* 413 (9-10): 861–68.

Zou, Hui, Trevor Hastie, and Robert Tibshirani. 2006. "Sparse Principal Component Analysis." *Journal of Computational and Graphical Statistics* 15 (2): 265–86.