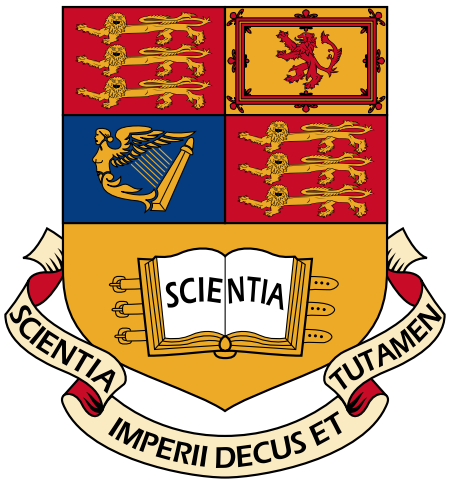
VISUALISING CANCER OVER A MANIFOLD

GEORGE TRIGEORGIS



MEng Artificial Intelligence
Computing Department
Imperial College London

Supervisor: Prof. Duncan Gillies
Second Marker: Prof. Yike Guo

ACKNOWLEDGEMENTS

First and foremost I would like to thank Prof. Duncan Gillies and Prof. Yike Guo for the supervision of my project. I would also like to thank Zena from Imperial College and 'Ed' from the Hammersmith Hospital who provided guidance and data for the completion of this project.

Lastly, but certainly not least, I would like to thank my family and friends for supporting me my time through university.

We have seen that computer programming is an art, because it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty.

— Donald E. Knuth (Knuth, 1974)

Microarray databases constitute a large source of biological data, which upon the proper analysis of the data, can enhance our understanding of biology and medicine. The main objective of this project is to provide means to researchers to visualise better the available information of a patient's genome in order to provide them with more suitable medical advice on the treatment of a particular form of cancer.

Although the analysis of microarray data has become popular in the research world the past 10 years, new approaches will be tested which may lead to a better visualisation of the available information.

To accomplish this we are investigating the use of manifold learning techniques on the microarrays to aid to the visualisation of the information but also create machine learning models that allow uncovering more of the information hidden in the noisy data of microarrays.

CONTENTS

1 INTRODUCTION

1.1 MOTIVATION

Cancer related diseases are unquestionably one of the most feared diseases of our time. In 2007, cancer caused about 13% of all human deaths worldwide (7.9 million). Although the term *cancer* refers to a broad group of diseases, (taken as a whole) about half of the people receiving treatment for invasive cancer, either die from it, or from the treatment itself. (Jemal et al., 2011).

In the context of using chemotherapy as the treatment of cancer, it is important to choose the correct dosage of cytotoxic drugs. Based on published data (Gurney, 2002), there is almost 20% relative reduction in survival for women receiving adjuvant chemotherapy for breast cancer, as a result of unrecognised under dosing. If we could better predict how rapidly the cancer will progress on a patient, we may also be able to provide a better treatment using a more appropriate drug dosage for the patient.

Using microarrays (2.1.1) we can look at the expression level of thousands of genes in a single experiment (Schena et al., 1995). Using this information we can compare the expression profile of tumour cells to their normal counterpart during different phases of the cancer process (initiation, progression and metastasis; see 2.1.2) which would give us an enhanced understanding of the process of tumour formation and its development (Kumaravel Somasundaram, 2002).

This is obfuscated by the fact that most cancers are highly heterogeneous, microarray results are noisy and there is large variation in the 'normal' levels of most genes.

The motivation behind this thesis is to build on top of the existing state-of-the-art methodologies using machine learning to provide a model of the microarray data obtained from patients. By doing so we aim to:

- Help the experts in the genetics to better visualise the information they have in hand.
- Uncover more information hidden in the noisy data, so machine learning algorithms can make better use of the data to make predictions.

1.1.1 *Ovarian Cancer*

Ovarian is one of the lesser known forms of cancer but still is a major issue nowadays.

An estimated 22,240 of new cases of ovarian cancer are expected just in US in 2013 and about 14,030 deaths. Ovarian cancer is responsible for about 3% of all

1 Source: http://www.ncbi.nlm.nih.gov/pubmed/

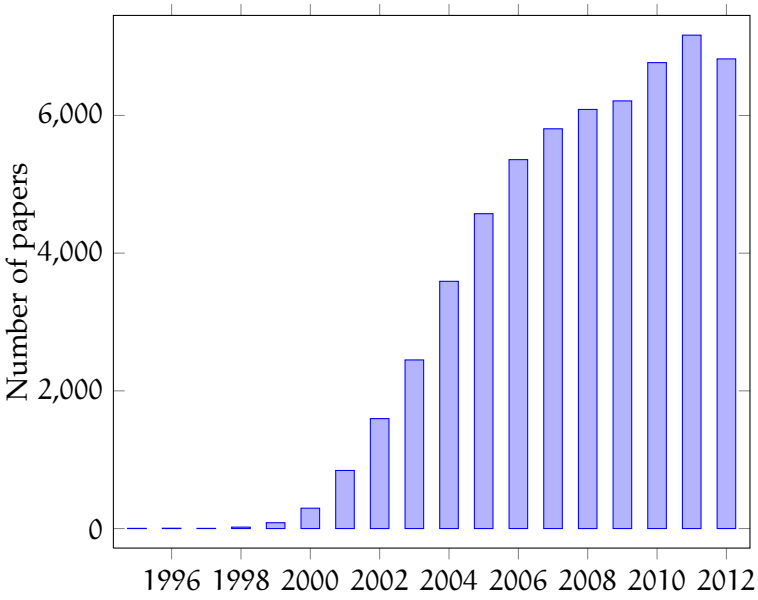


Figure 1: Trend of microarray related paper publications for the past 12 years. [1]

cancers amongst women accounting with about 5% of cancer caused deaths among women, producing more deaths than any other cancer of the female reproductive system.

Currently there is no sufficient accurate screening test of ovarian cancer. Examination of the pelvis only occasionally detects ovarian cancer and usually when the cancer is at an advanced state.

Unfortunately ~ 70% of the deaths are cause of patients presenting with advanced stage; high-grade serous ovarian cancer (HGS-OvCa).

The standard treatment is aggressive surgery followed by platinum-taxane chemotherapy. Surgery usually involves removal of one or even both ovaries and fallopian tubes (salpingo-oophorectomy), the uterus (hysterectomy), and the omentum. This kind of surgery has many risks and adverse effects such as small bowel obstruction (SBO) (Barmparas et al., 2010), premature death (Parker et al., 2009), cardiovascular disease, cognitive impairment or dementia (Rocca WA, 2007) and others.

For women with a very early stage of ovarian cancer who wish to have children only the involved ovary and fallopian tube may be removed in order to allow them to carry birth. After therapy, platinum resistant cancer recurs in approximately 25% of patients within six months, and the overall five-year survival probability is 31% (Jemal et al., 2009). Approximately 13% of HGS-OvCa is attributable to germline mutations in BRCA1/2 (Pal et al., 2005)), and a smaller percentage can be accounted for by other germline mutations. However, most ovarian cancer can be attributed to a growing number of somatic aberrations.

Unfortunately little can one tell using the microarray expression data available from patients that have ovarian cancer, so one of aims of this project is to help uncover more information from the ovary gene expression data that are available to us, just as for other types of cancer.

1.1.2 *Why not use RNA sequencing?*

In the last few years there has been much speculation whether using DNA microarrays should be the experiment of choice for transcriptome analysis (gene expression profiling).

RNA-seq "Whole Transcriptome Shotgun Sequencing" (Morin et al., 2008) is rapidly adopted in studies and it seems that this should be preferable to DNA microarrays as it allows for an accurate high-throughput sequencing of cDNA in order to get the RNA content of a sample. In contrast to microarrays, it allows for base-level resolution, overcoming many of the shortages of DNA microarrays.

Microarrays however are currently a fraction of the cost of RNA-seq ($10-100$ times cheaper) as the cost of RNA-seq only lies between $\$8,000$ - $\$20,000$ and also RNA-seq protocols still suffer from unknown biases such as those implied by the required ligation steps (Birzele et al., 2011).

There has been much speculation whether "Is this the beginning of the end for microarrays?" (Shendure, 2008) where the author concludes saying that although these new technologies may improve the quality of transcriptome profiling, we will continue to face what has probably been the largest challenge of microarrays - how best to generate biologically meaningful interpretations of complex data sets that are sufficiently interesting to drive follow-up experimentation.

Thus one can argue if being able to meaningfully interpret microarrays can give the much anticipated insight into these complex biological data sets.

1.2 REPORT STRUCTURE

Here follows a brief overview of the information available in the next chapters of the report.

BACKGROUND: gives an overview of the research done so far in the area of microarray analysis, the main methodologies of clustering the microarray data and the mathematical formulations of the various algorithms we will be using.

VISUALISING USING MANIFOLD LEARNING: we use traditional manifold learning techniques such as ISOMAP, Locally Linear Embedding and Spectral Embedding on microarray data focusing on the ISOMAP algorithm.

A PRIORI MANIFOLD LEARNING: we propose a novel way to enhance the performance of traditional manifold learning techniques by making use of *a priori* knowledge about our data sets, such as biological pathway information incorporation in the manifold analysis.

EVALUATION: evaluates, compares and contrasts the various manifold learning techniques with the aforementioned a priori manifold learning.

CONCLUSIONS & FUTURE WORK: summarises the most important results, but also the main difficulties found in the process of this project. We suggest future developments of this work and how it can potentially be extended to a Doctoral level.

2 BACKGROUND

2.1 MICROARRAYS

2.1.1 *What is a microarray?*

Etymology

Micro + Array

Micro: quantifyer prefix, multiplication by 10^{-6}

Array: order, arrangement

Wikipedia: "a collection of microscopic DNA spots attached to a solid surface"

Microarray technology has been used a lot and has become an indispensable tool that many biologists use to monitor genome-wide expression level of genes in a given organism.

Microarrays typically, are stranded DNA/RNA molecules that are anchored by one end on a solid surface, which is usually a glass.

Figure 2: Example of a commercial type microarray assay: the Affymetrix GeneChip® Genome-wide Human SNP Array 5.0

A microarray typically ranges from 1.3×1.3cm to 2.5×7.5cm and has thousands of spots, many of which are replicates, and control spots with a spot size of approximately 5µm. These spots (also called probes) are printed on to the solid surface by

a robot or are synthesised by the processes of photolithography. Photolithography is using light to create pattern, relying on UV masking and light-directed combinatorial chemical synthesis on a solid support to selectively synthesise the probes on the surface of the array.

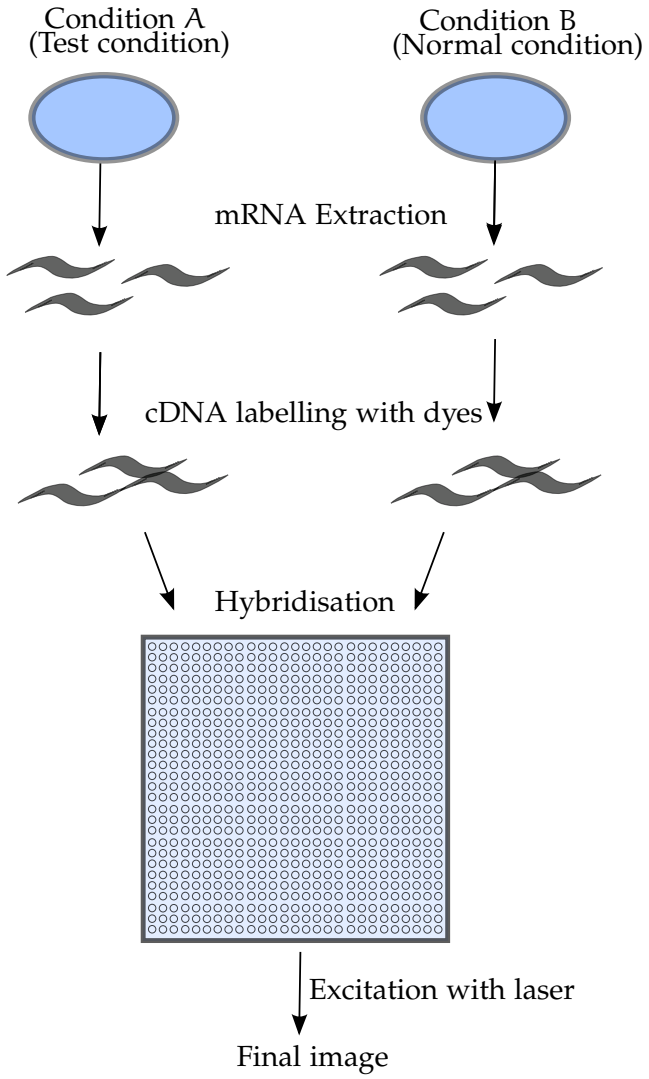


Figure 3: Schematic of the experimental protocol to study differential expression of genes. The organism is grown in two different conditions (a reference condition and a test condition). RNA is extracted from the two cells, and is labelled with different dyes (red and green) during the synthesis of cDNA by reverse transcriptase. Following this step, cDNA is hybridised onto the microarray slide, where each cDNA molecule representing a gene will bind to the spot containing its complementary DNA sequence. The microarray slide is then excited with a laser at suitable wavelengths to detect the red and green dyes. The final image is stored as a file for further analysis.

2.1.2 *Obtaining the gene expression data*

One of the most popular ways to measure gene expression is to compare the expression of a set of genes from a cell maintained in a particular condition A (test condition) to the same set of genes from a reference cell with condition B (normal condition).

These molecules will seek to hybridise with complementary stands floating in the solution.

Probes (probe is a fragment of DNA or RNA of variable length) are organised in clusters of spots and each probe usually corresponds to one gene/protein. Probe hybridisation is usually detected and quantified with the use of fluorophore, silver, or chemiluminescence labelled targets which detect the relative abundance of nucleic acid sequences in the cDNA sample.

Sources of Error

By doing the comparison of the gene expression levels of the two samples (A, B), one can often find that the average expression ratio of some genes - that should not change in the two experimental conditions - can deviate from an average expression ratio of 1. This can be due to numerous reasons such as differential labelling efficiency of the two fluorescent dyes or different amounts of starting RNA material in the two samples.

There are several factors which can lead to errors in microarray hybridisation data, such as the usual manufacturing defects in the microarrays themselves, but most importantly the experimental errors caused by humans.

In order to be able to take advantage of the microarray results and be able to analyse them, each gene sample is logarithmically normalised (i.e. using the correlation coefficient).

Robust Multi-array Average (RMA) is a popular normalisation approach for Affymetrix and other data where the raw intensity values are background corrected, $\log_2$ transformed and then quantile normalised (Irizarry et al., 2003).

2.2 CLUSTERING MICROARRAY DATA

Eisen et al. (1998) first demonstrated the ability of clustering microarray data results to reveal biologically meaningful patterns with the method of hierarchical clustering to identify groups of genes with similar behaviour.

Prior to that Weinstein et al. (1997) presented one of the first and most elegant applications of hierarchical clustering and other data-mining and visualisation techniques to the analysis of large-scale data in molecular biology.

Hierarchical clustering is an algorithm which finds pairs of genes that are most similar, links them together continuing on to the next most similar pair of such genes, by using a given similarity metric. This is then repeated until all groups are linked to one cluster.

The problem with hierarchical clustering is that is a greedy algorithm and once a decision to link two clusters is made, it cannot be reversed in order to follow a better clustering outcome. It is additionally overfitting the data with no way to generalise the results. Finally, as microarrays are very error prone and hierarchical clustering is prone to noise and outliers, hierarchical clustering can give quite unexpected results.

A common problem with partitioning methods is that they end up assigning each data point to a group. This may not be desirable for gene clustering as genes may

Table 1: Brief history of microarrays

Year	Event
1953	Double helix, Watson & Crick
1961	DNA hybridisation and nature of the triplet code discovered
1970	Reverse transcriptase (Baltimore, 1970)
1975	Monoclonal antibodies (Kohler and Milstein, 1975)
1977	First DNA sequence of an organism (viral, Sanger)
1989	First microarray prototype using a microscope slide (Fodorand and coll.)
1993	Microarray containing over 1 million DNA sequences
1995	First microarray publication (Schena et al., 1995): Arabidopsis thaliana
1996	Commercialisation of arrays (Affymetrix)
1997	Genome-wide expression analysis in S. cerevisiae (DeRisi et al., 1997)
1998	First multicellular eukaryotic genome sequenced: (Elegans, 1998)
1999	First publication on microarrays for cancer classification (Golub et al., 1999): Leukaemia
2000	Portraits/signatures of cancer: First publications on molecular phenotyping in cancer (Perou et al., 2000)
2001	Human Genome published (Nature)
2004	Whole human genome on one microarray (Affymetrix)
2005	First FDA approved microarray based product

be involved in more than one active biological process or none at all. Clusters of genes which are not involved in any active process can be ignored, as we can filter out genes with near-constant expression profiles. During this process one might end up not using a large portion of the data set which could be used to obtain a better clustering of the data. Thus it is much more preferable to have a clustering algorithm that would leave "uninteresting" patterns unclustered.

We will talk about three types of clustering techniques:

ONE-WAY clustering methods where clusters are limited to either the rows(genes) or columns(samples) of the data set.

TWO-WAY clustering are used to find clusters combining both genes and samples.

BI-CLUSTERING methods to find two-dimensional clusters - a subset of genes which exhibit similar behaviour across a subset of samples or vice versa.

2.2.1 *One-way clustering*

Examples of one-way clustering can be accomplished using traditional machine learning clustering algorithms such as K-means, DBSCAN, Gaussian mixtures and others. Such methods can be problematic in clustering microarray data as usually a subset of the feature space (genes) can be associated with a subset of observations. It has been shown in the recent years that certain genes when over-expressed or down-expressed identify with high precision the different sample groups (i.e. cancerous patients and not). This means that by attempting to cluster the genes we will end up having unrelated clusters of genes of which will provide us no useful information.

2.2.2 *Two-way clustering*

Using one-way clustering we can obtain either gene-related clusters or sample-related clusters. Moreover one can argue though that there must be relationships between gene and sample clusters. As an example samples from patients taken on different stages of the cancer process can lead to different clustering of the genes.

An example of how this is useful was demonstrated by Tang et al. (2001), the genes are clustered and each one of the clusters used to cluster the samples. The concept was, as the dimensionality of the each sample vector is too large reduce to a reasonable level and then work with a clustering algorithm on the reduced data set. To achieve this Tang et al. (2001) did the following:

- to find a subset of genes (important genes) which are highly related to experiment conditions.
- cluster the samples into different groups (usually just two i.e. diseased and control samples)

These two tasks are related. By finding the most influential genes, then it is easier to cluster the samples due to the lower sample dimension (tens instead of thousands genes). Otherwise, if we cluster the samples, we can find the most important genes by sorting the genes using similarity scores such as correlation coefficient.

The advantages of this approach is to use the relationships between genes and samples to do an iterative clustering where by reducing the gene-dimension improves the classification accuracy.

Tang et al. (2001) was among the first to propose and use unsupervised two-way clustering techniques instead supervised techniques arguing that this is more suitable for problem domains with limited domain knowledge. In his paper, he presented a new framework for unsupervised analysis of gene expression data, which applies an interrelated two-way clustering approach on the gene expression matrices comparing the performance of the proposed method with various gene expression data sets.

2.2.3 *Bi-clustering*

Bi-clustering was coined by Mirkin (1996), although the technique was known much earlier on (Hartigan, 1972). A bicluster is defined to be a set of genes whose ex-

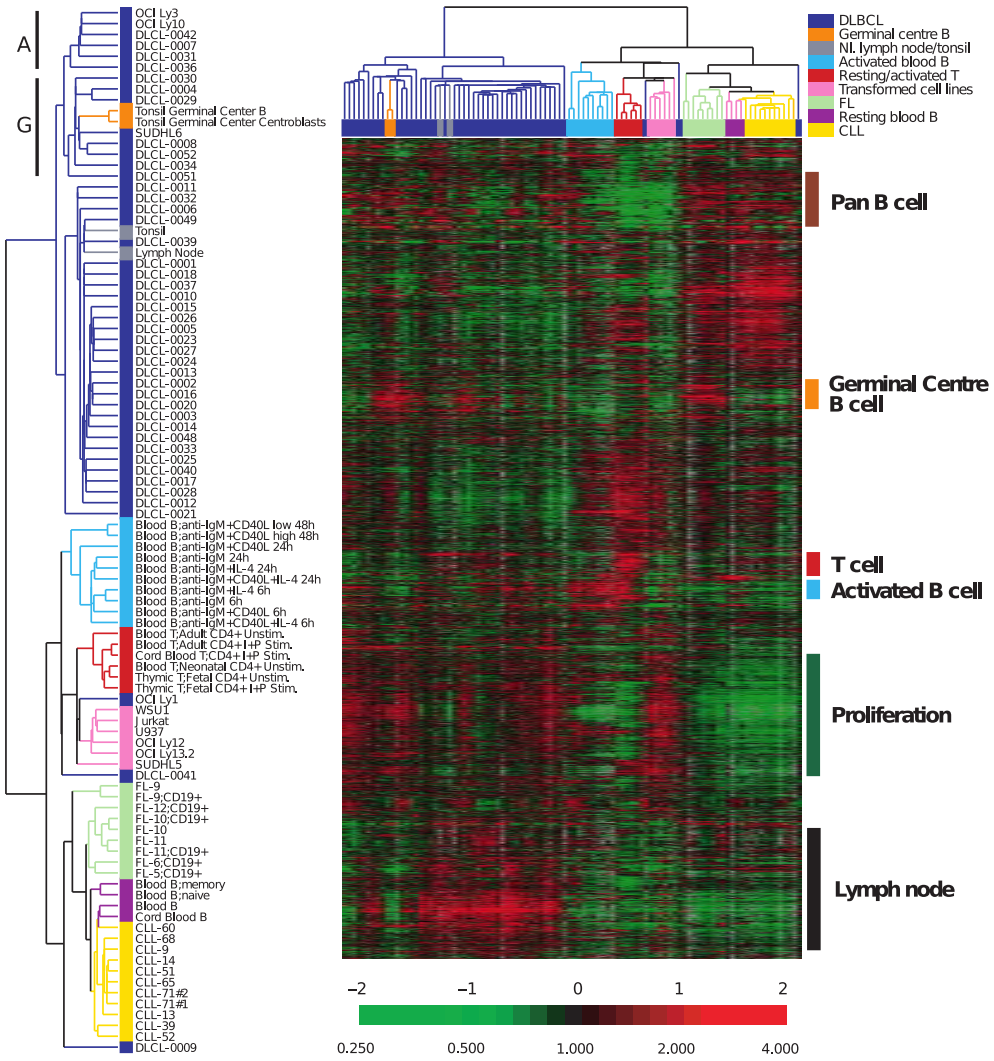


Figure 4: Hierarchical clustering of gene expression data. Depicted are the, 1.8 million measurements of gene expression from 128 microarray analyses of 96 samples of normal and malignant lymphocytes. The dendrogram at the left lists the samples studied and provides a measure of the relatedness of gene expression in each sample. The dendrogram is colour coded according to the category of mRNA sample studied (see upper right key). Each row represents a separate cDNA clone on the microarray and each column a separate mRNA sample. The results presented represent the ratio of hybridisation of fluorescent cDNA probes prepared from each experimental mRNA samples to a reference mRNA sample. These ratios are a measure of relative gene expression in each experimental sample and were depicted according to the colour scale shown at the bottom. As indicated, the scale extends from fluorescence ratios of 0.25 to 4 (-2 to +2 in log base 2 units). Grey indicates missing or excluded data. (Alizadeh et al., 2000)

pression profiles are mutually similar within a subset of experimental conditions (samples).

The intuition behind why one should prefer bi-clustering is that, due to the diversity of sample sources, functionally related genes may not exhibit a similar pattern in all samples but only in a subset of them. Bi-clustering addresses this by trying to find patterns between subset of columns and subset of rows of the microarray matrix. Since such subsets are initially unknown, bi-clustering can be seen as a simultaneous clustering of rows and columns.

Cheng and Church (2000) have done some significant ground work on bi-clustering and numerous bi-clustering algorithms have been suggested since then. Generally, there are two kinds of bi-clustering patterns; those that are defined on a single bicluster and deal with *local* patterns or those which are defined on all the *global* bicluster patterns.

A bicluster $\mathcal{B} = (I, J)$ is composed of a subset of rows $I \subset R$ and a subset of columns $J \subset C$, where all a_{ij}, for $i \in I$ and $j \in J$ are expected to fit to a predetermined target pattern for a given gene expression data matrix $A = (R, C)$. Of course it is usually the case that the bicluster will not fit exactly the predetermined target pattern so:

$$b_{ij} = \hat{b}_{ij} + \epsilon_{ij} \tag{2.2.1}$$

where $\hat{b}_{ij}$ is the expected value of b_{ij} that would match best to the target pattern and ϵ_{ij} the deviation from the expected value also known as residue. The residue is commonly used as a metric of how good a bicluster is. A popular bicluster quality metric is the Mean Squared Residue (MSR) which is defined to be:

$$\mathbf{MSR} = \frac{1}{|I||J|} \sum_{j \in J, i \in I} \epsilon_{ij}^2$$

If the bicluster would match exactly its target pattern then it would have a residue of 0, which also matches its MSR score.

Using 2.2.1 we can represent different types of algorithms, that perform searching for most bi-clustering methods using the appropriate expression for $\hat{b}_{ij}$.

The bicluster $\mathcal{B}$ is said to follow a *perfect shifting pattern* as its values can be obtained by adding a constant condition number β_i to a base value π_j. The shifting pattern can be fulfilled by using the following equation, where β_i is the *shifting* coefficient of condition i.

$$\hat{b}_{ij} = \beta_i + \pi_j$$

Similarly, instead of shifting the base value, we can use multiplication to scale the pattern. In this case, we say that the bicluster follows a *perfect scaling pattern*. In the following equation the term α_i is called the *scaling* coefficient, and represents a constant value for each sample (condition).

$$\hat{b}_{ij} = \alpha_i \times \pi_j$$

We can also form a *combined pattern* which is simply the shifting and scaling patterns put together.

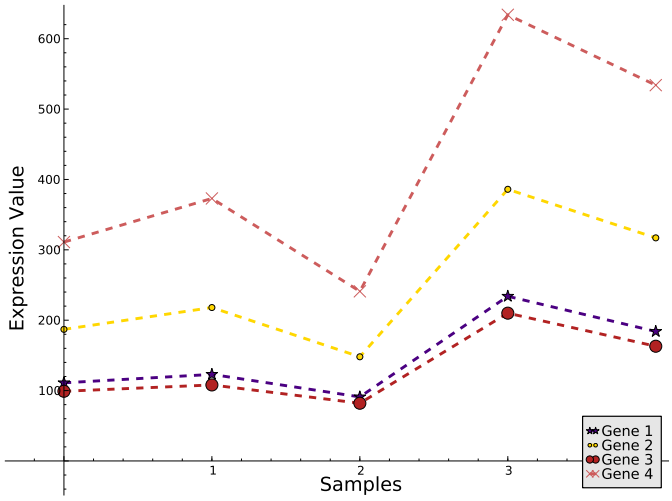
$$b_{ij} = \pi_j \times \alpha_i + \beta_i$$

Example

$$\pi = \begin{pmatrix} 23 \\ 42 \\ 20 \\ 73 \end{pmatrix}, \alpha = \begin{pmatrix} 4 \\ 5 \\ 3 \\ 8 \\ 7 \end{pmatrix}, \beta = \begin{pmatrix} 19 \\ 8 \\ 22 \\ 50 \\ 23 \end{pmatrix}$$

Forms the following bicluster:

$$\mathcal{B} = \begin{pmatrix} 111 & 187 & 99 & 311 \\ 123 & 218 & 108 & 373 \\ 91 & 148 & 82 & 241 \\ 234 & 386 & 210 & 634 \\ 184 & 317 & 163 & 534 \end{pmatrix}$$



Cheng and Church's Algorithm

One of the first bi-clustering algorithms was presented by Cheng and Church (2000). The model for a bicluster is represented by a submatrix A of the whole gene expression matrix.

Each a_{ij} in the bicluster is the summation of the background level, the gene effect (rows) and the sample effect (columns). The residue score of an element is given by

$$R(a_{ij}) = a_{ij} - a_{iJ} - a_{Ji} + a_{IJ}$$

Table 2: Various bi-clustering algorithms summarised from Kevin Yip, 2003

Method	**Publish**	**Cluster Model**	**Goal**
Cheng & Church	ISMB 2000	Background + row effect + column effect	Minimise mean squared residue of bi clusters
Getz et al. (CTWC)	PNAS 2000	Depending on plugin clustering algorithm	Depending on plugin clustering algorithm
Lazzeroni & Owen (Plaid Models)	Bioinformatics 2000	Background + row effect + column effect	Minimise modelling error
Ben-Dor et al. (OPSM)	RECOMB 2002	All genes have the same order of expression values	Minimise the p-values of biclusters
Tanay et al. (SAMBA)	Bioinformatics 2002	Maximum bounded bipartite subgraph	Minimise the p-values of biclusters
Yang et al. (FLOC)	BIBE 2003	Background + row effect + column effect	Minimise mean squared residue of biclusters
Kluger et al. (Spectral)	Genome Res. 2003	Background 'row effect' column effect	Finding checkerboard structures

where:

$$\bar{a}_{iJ} = \frac{\sum_{j \in J} a_{ij}}{|J|} \qquad \textbf{mean of row } i$$

$$\bar{a}_{Ij} = \frac{\sum_{I \in I} a_{ij}}{|I|} \qquad \textbf{mean of column } j$$

$$\bar{a}_{IJ} = \frac{\sum_{j \in J, I \in I} a_{ij}}{|I||J|} \qquad \textbf{mean of column } j$$

2.2.4 *Data mining approaches*

One can distinguish two main data mining approaches:

SUPERVISED LEARNING which defines a model which related one set of observations, called inputs, to another set of observations, called outputs.

UNSUPERVISED LEARNING which does not have any prior information about inputs associated with any outputs, but rather tries to uncover the structure of the data set from the input data provided.

Although often in microarray experiments we are given the labels of each observation (e.g. patient cancer regression status), we can opt to not use them while conducting a model of the data due to the small number of observations we have compared to the feature space (Klein et al., 2002; Tang et al., 2001). This means that we are less likely to overfit a model onto the data of the microrrays, but rather create a more general one.

Moreover a problem with supervised learning is that labelled samples are usually very difficult to obtain, as labelling is usually an expensive and time consuming job. There are numerous microarray databases with information that could go to waste simply because the samples were not labelled. A way to combat this is to use pairwise constrains information (side-information), which leads to a learning approach called semi-supervised learning.

In the next few chapters we will introduce and use unsupervised dimension reduction techniques of manifold learning on gene expression data but also introduce the concept of semi-supervised learning on the manifold learning techniques. Many machine-learning researchers have found that unlabelled data, when used in conjunction with a small amount of labelled data, can produce considerable improvement in learning accuracy. Instead of completely ignoring prior information we may have for our samples semi-supervised learning uses in some degree prior information, but not for all the data or it may not rely completely in the sample labels to build up the machine learning model. Such techniques we will use later on, in Chapter 4, where we introduce a novel method of semi-supervised learning called *a priori* manifold learning.

2.2.5 *Distance measures*

Inherent in all machine learning techniques, is the notion of similarity or of a distance function for a data set that needs to be classified or clustered. The choice of a distance metric is probably equally important to the choice of the machine learning algorithm, as it can greatly affect the outcome of the algorithm.

Although it may be common to use metaphors to describe the distance between two objects in high-dimensional data in various disciplines, such as the distance between two genes as a quantity measured in base pairs along a chromosome, this may not be always practical as genes may be present in different chromosomes.

Euclidean Distance

A classical measure of measuring distance is the Euclidean, also known as the Pythagorean distance. Euclidean is given by the Pythagorean and is defined as:

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(q_1-p_1)^2+\cdots+(q_n-p_n)^2} = \sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}. \tag{2.2.2}$$

Pearson correlation coefficient

One of the most commonly used metrics to measure similarity between expression profiles is the Pearson correlation coefficient (PCC) (Eisen et al., 1998). PCC between two samples X and Y can be computed as follows:

$$r = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \tag{2.2.3}$$

where $\bar{X}$ ($\bar{Y}$) is the sample mean of the sample X (Y)

Curse of dimensionality

When comparing observation with observation though a distance metric is argued not to play much of a role in comparing the distances. When a measure such as a Euclidean distance is defined on a lot of dimensions, there is little difference in the distances between different pairs of samples (Beyer et al., 1999).

2.3 MICROARRAY DATA FORMAT

Gene expression data on G genes for N samples are usually summarised by a $G \times N$ matrix X, where x_{ij} denotes the expression measure of gene i of mRNA/DNA sample j. The expression value may be either absolute (e.g. Affymetrix oligonucleotide

arrays) or relative to the expression levels. (R.). The data structure will be referred to as X.

$$X(t) = \begin{bmatrix} x_1(1) & \cdots & x_1(N) \\ \vdots & \ddots & \vdots \\ x_G(1) & \cdots & x_G(N) \end{bmatrix} \tag{2.3.1}$$

Usually this set of data will be associated with groups' labels vector $y(t)$ which maps each sample $x(t)$ to a label $y(t)$.

2.4 DIMENSIONALITY REDUCTION

In machine learning as the dimensionality of the data rises, the amount data we would require to be able to support the outcome of the machine learning algorithm often grows exponentially. Richard E. Bellman referred to this phenomenon as 'curse of dimensionality' when considering problems in dynamic optimisation (Bellman, 1957).

Not only considering more genes (variables) adds to the computation cost as mentioned but it also decreases the statistical significance of obtaining a good result (Kung and Mak., 2009). In order to combat the 'curse of dimensionality' several methods have been tested from researchers.

By being able to choose the most influencing genes amongst the set we can greatly increase the effectiveness of our machine learning system. In machine learning this process is also known as *Feature Selection*, but in the context of microarrays is know as *Gene Selection*.

OPEN-LOOP Filter methods which select features based on between-class separability criterion, which is not involved in classification performance in the process of feature selection.

CLOSED-LOOP Wrapper methods select features using classification performance as a criterion of feature subset selection.

We will use both Open-Look and Closed-Loop techniques to determine the criterion of the feature set reduction.

2.5 NONLINEAR DIMENSIONALITY REDUCTION

Another approach to dimensionality reduction is to assume that the data (genes of interest) lie on an embedded non-linear manifold. Algorithms based on manifold learning are based on the idea that the high dimensionality of some data sets is only artificially high; although each point consists of multiple features (e.g. thousands) it can be described as a function of just a few parameters. In other words samples are actually samples from a low-dimensional manifold that is embedded in a high-dimensional space. (Cayton, 2005)

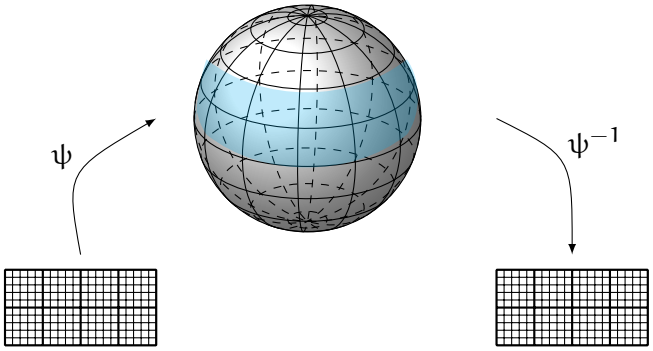


Figure 5: In manifold learning we usually assume that a low dimensional manifold is embedded in a higher dimensional space, which is in this case a sphere. Our aim, in this example, is to transform the manifold from the higher dimensional space back to its original form, the 2-dimensional plane.

The rational behind using manifold learning it to discover the hidden structure of the high-dimensional microarray data so by visualisation of the low-dimensionality output of the machine learning process we can uncover previously unseen features of the data.

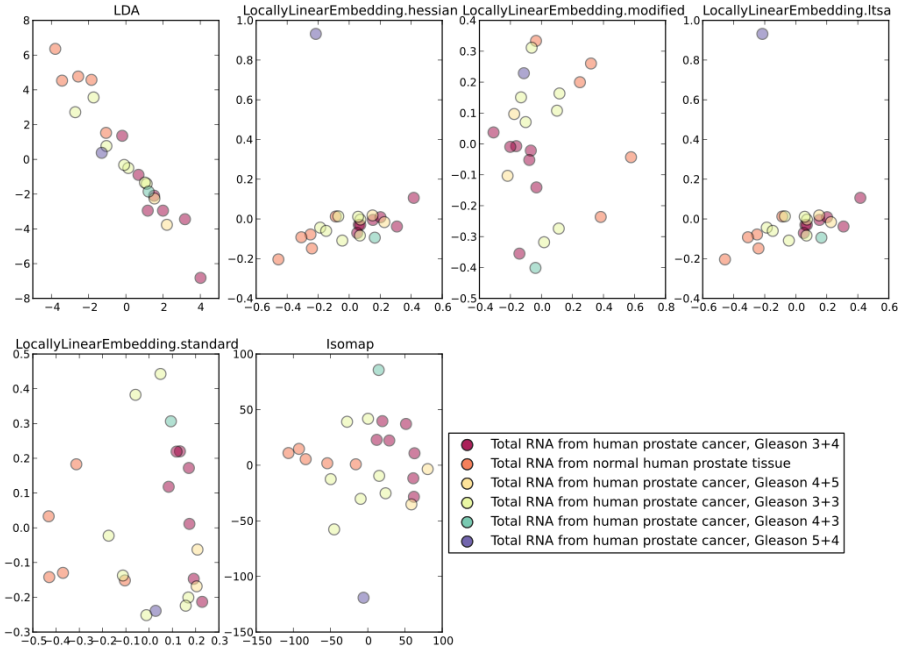


Figure 6: Example output of various non linear dimensionality reduction algorithms of a data-set of expression profiles of 18 prostate samples (7 with Gleason 6, 8 with Gleason 7 and 3 with Gleason score equal or higher than 8) and 5 non-neoplastic prostate samples, using the GeneChip® Human Exon Array 1.0 ST of Affymetrix.

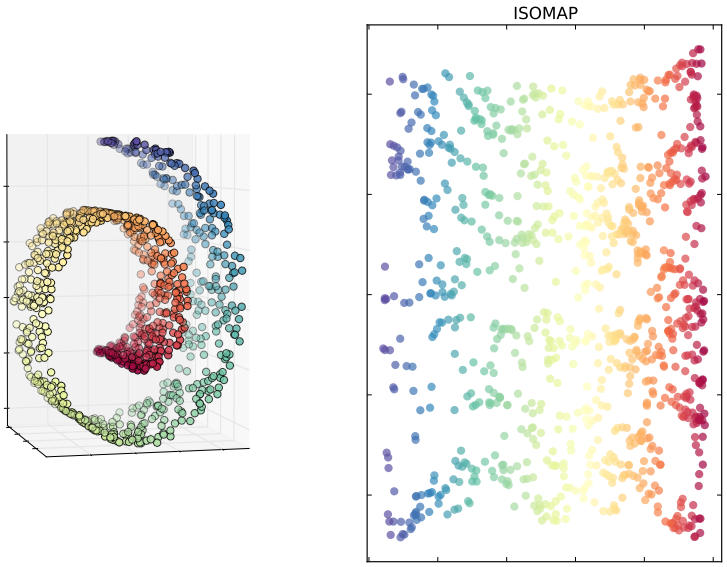


Figure 7: An example application on a 3D Swiss roll data-set with N = 1000 data points. The original data-set is illustrated on the left in the original three-dimensional space. The ISOMAP projection to the two-dimensional space on the right was done, by taking a number of neighbours equal to 8. It can be seen that nearby points in the 2D embedding are also nearby points in the original 3D manifold, as desired.

2.5.1 *ISOMAP*

The complete isometric feature mapping (ISOMAP) Tenenbaum et al. (2000) is a manifold learning algorithm works by getting a set of data points represented as M-dimensional vectors $x \in \mathbb{R}^{M \times n}$ and yielding a d-dimensional representation of the original data set such as the Euclidean distance between two points of the final projection approximates their geodesic distance along the underlying manifold as much as possible.

ISOMAP is based on the idea of the classical multidimensional scaling technique also known as MDS. Classical MDS constructs first the pairwise similarity matrix using a distance measure such as the Euclidean distance function. Afterwards it computes the reduced dimensional mapping that preserves as much as possible the similarity matrix in predefined reduced dimension. On the other hand ISOMAP constructs the pairwise similarity matrix based on the geodesic distance estimated by the shortest path in the neighbourhood graph of the data set.

Geodesic distance is the number of edges in the shortest path between two vertices of a graph

Unfortunately in practice there is often no guarantee of the existence of a well-defined underlying manifold structure in the data, and thus, one may not be sure if manifold learning techniques such as ISOMAP are suitable for the data at hand. In any case one can still try to apply the algorithm to see if it fits our purpose Tenenbaum et al. (2000).

ISOMAP can be summarised in three main steps as follows:

1. Construct neighbourhood graph Define the graph G over all data points by connecting points i and j (as measured by distance metric $d_X(i, j)$) (i, j)] they

are closer than ε (ε-ISOMAP), or if i is one of the K-nearest neighbours of j (K-ISOMAP). Set edge lengths equal to $d_X(i,j)$.

2. Compute shortest paths between each of the nodes of the graph (for example using Floyd-Warshall or Dijkstra's algorithm) and thus obtaining the geodesic distance between all pairs of points on the manifold and let $\Delta \in \mathbb{R}^{n\times n}$ be this matrix of the pairwise geodesic distances.

3. Finally we can compute the d-dimensional embedding by computing the eigenvalues of the matrix Θ (Equation 2.5.1). Sorting the eigenvalues in decreasing order we can form the d-dimensional embedding such that for each dimension with $1 \leqslant k \leqslant d$:

$$\Theta = -\frac{1}{2} H \Delta^2 H \tag{2.5.1}$$

where H is the centring matrix

$$H = I_n - \frac{1}{N} e_N e_N^T, \text{with } e_N = [1,\dots,1]^T$$

Let λ_k be the k^{th} eigenvalue and v_k be the k^{th} eigenvector. We construct the k^{th} component of the embedding Π by setting it to $\sqrt{\lambda_k} v_k$.

$$\Pi = \begin{pmatrix} \sqrt{\lambda_1} v_1 \\ \sqrt{\lambda_2} v_2 \\ \sqrt{\lambda_3} v_3 \\ \vdots \\ \sqrt{\lambda_d} v_d \end{pmatrix}$$

Complexity

As described before the ISOMAP algorithm has a three step process.

NEIGHBOUR-SEARCH The naive approach using a linear search of the data space, would have a running time of $\mathcal{O}[D \times N^2]$, but this can be dramatically reduced using techniques that use space partitioning such as ball trees.

Ball trees (Omohundro, 1989) are geometric data structures designed to provide fast nearest neighbour searching in high-dimensional spaces. The structure is similar to other hierarchical representations such as *k-d* trees, but has the advantage that performs better in higher dimensions by partitioning data in a series of nesting hyper-spheres.

The cost is $\mathcal{O}[D\log(k) \times N\log(N)]$ (Omohundro, 1989)

GEODESIC DISTANCE CALCULATION There are two main algorithms for computing the inter-node path distance which are Dijkstra's algorithm (Dijkstra, 1959) and Floyd-Warshall's Algorithm (Floyd, 1962). Each of the algorithms has a computational cost of $\mathcal{O}[N^2(k + \log(N))]$ and $\mathcal{O}[N^3]$ respectively.

EIGENVALUE DECOMPOSITION After the construction of the geodesic distance matrix, ISOMAP needs to find the d largest eigenvalues. Using an algorithm such as Power Iteration this can be computed in $\mathcal{O}[dN^2]$, but for our needs we can use the ARPACK[1] solver to improve this.

To conclude ISOMAP has a complexity of the order:

$$\mathcal{O}[Dlog(k) \times Nlog(N) + N^2(k + log(N)) + dN^2]$$

where
D: Number of dimensions
N: Number of data points
k: Number of nearest neighbours to search for
d: The number of the final reduced dimensions.

2.5.2 *Locally Linear Embedding*

Locally Linear Embedding (LLE) attempts to solve the nonlinear dimensionality reduction problem by computing a low-dimensional neighbourhood preserving embeddings of the high-dimensional data. It discovers nonlinear structure in high dimensional data by exploiting the local symmetries of linear reconstructions (Roweis and Saul, 2000).

For data which consist of N real-valued vectors $\vec{X}_i \in \mathbb{R}^D$ which is assumed to be sampled from a lower dimensionality d ($d \ll D$) manifold. Provided enough data we would expected that each data point along with its neighbours would lie on or nearby a locally linear patch of the manifold.

The LLE procedure can be separated in three main steps: building a neighbourhood graph for each point in the data, by using the k-NN algorithm as in ISOMAP, finding the weights for linearly approximating the data in the neighbourhood and finding the low-dimensional coordinates that best reconstruct these weights and then returning the low-dimensional embedding of the original space.

Lets assume that the manifold was linear around each data point $\vec{X}_i$. Then that means that, $\vec{X}_i$ along with its neighbours form a linear subspace of a certain dimension. But if the subspace is linear it means that there is a combination of weights for each of the neighbours as such to reconstruct $\vec{X}_i$ exactly.

$$\vec{X}_i = \sum_j w_{ij}\vec{X}_j$$

Using these weights those one can characterise how the manifold looks like in both the high and low dimensional space given small neighbourhood sizes.

The idea then is to minimise the reconstruction error needed to explain the each data point using the locally linear manifold of its neighbours.

$$E(W) = \sum_i \left|\vec{X}_i - \sum_{j \neq i} \mathbf{W}_{ij}\vec{X}_j\right|^2 \tag{2.5.2}$$

1 http://www.caam.rice.edu/software/ARPACK/

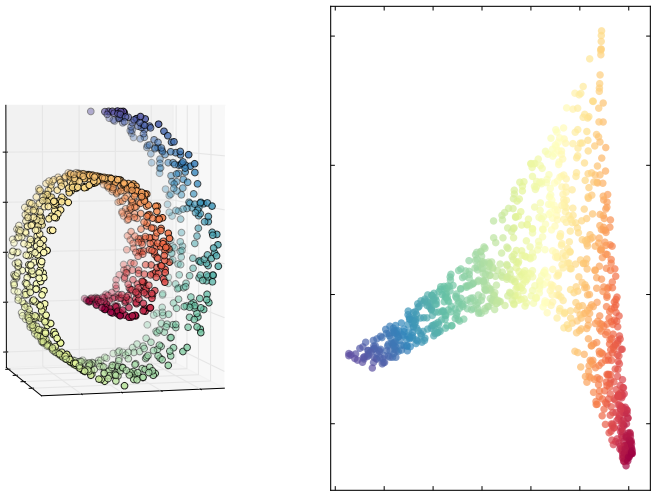


Figure 8: A demonstration of the standard Locally Linear Embedding on a three dimensional Swiss-roll data-set to two dimensions using `n_neighbours=8`

which adds up the squared distances between all the data points and their reconstructions. The weights W explain the contribution of the jth data point to the ith reconstruction. While minimising the reconstruction error one should also ensure that the weights sum up to one: $\sum_j W_{ij} = 1$. The reason for this is that the weights should have invariance to rotations, rescaling and translation of that data point and its neighbours. Invariance to rotations and rescaling is deduced from 2.5.2, the invariance under translation is enforced by ensuring that the weights sum to one, as if we add any vector $\vec{c}$ to $\vec{x}_i$ and its neighbours, nothing happens to the optimisation function

$$\vec{x}_i + \vec{c} - \sum_j w_{ij}(\vec{X}_j + \vec{c}) = \vec{x}_i + \vec{c} - \sum_j w_{ij}\vec{X}_j - \sum_j w_{ij}\vec{c} = \vec{x}_i - \sum_j w_{ij}\vec{X}_j \quad (2.5.3)$$

where $\sum_j w_{ij} = 1$.

To calculate the reconstruction weights W we follow the following procedure (Roweis and Saul, 2000):

for $i = 1 : N$ **do**
  create matrix Z consisting of all neighbours of X_i
  subtract X_i from every column of Z
  compute the local covariance $C = Z^T Z$
  solve $C\vec{w} = 1$
  set $W_{ij} = 0$ if j is not a neighbour of i
  set the remaining elements in the i-th row of W equal to $w/\sum w$
end

Note that when computing the local covariance C, if $K > D$ then the local covariance will not be full rank and thus we should a regularisation method to approx-

LLE Algorithm

The procedure can be summarised in the following steps:

1. Find the neighbours of each data point $\vec{X}_i$, either by choosing the k-nearest neighbours or by choosing all samples within a fixed distance ϵ.

2. Compute the weights that best summarise the contribution of the j-th data point to the i-th reconstruction by minimising the cost function:

$$E(W) = \sum_i \left| \vec{X}_i - \sum_{j \neq i} \mathbf{W}_{ij} \vec{X}_j \right|^2$$

3. Each high-dimensional observation $\vec{X}_i$ is mapped to a low-dimensional vector $\vec{Y}_i$ representing global internal coordinates on the manifold by minimising the cost function:

$$C(Y) = \sum_i \left| \vec{Y}_i - \sum_{j \neq i} \mathbf{W}_{ij} \vec{Y}_j \right|^2$$

imate the result. In the original paper the following regularisation is used (2.5.4), although we will study additional LLE algorithms that solve this problem in a different way.

$$\mathbf{C} = \mathbf{C} + \varepsilon \times \mathbf{I} \tag{2.5.4}$$

where:
ε is a small constant of the order $0.001 \times tr(\mathbf{C})$
I is the identity matrix

Complexity

The first and third steps of the algorithm are similar to the ISOMAP algorithm which was discussed in 2.5.1.

The second step of the LLE has to compute the weight matrix which requires the solution of a $k \times k$ linear equation system for every one of the N local neighbourhoods computed using the nearest neighbours algorithm. This has a cost of $\mathcal{O}[DNk^3]$.

Thus the complexity of the standard LLE algorithm is of the order of:

$$\mathcal{O}[Dlog(k) \times Nlog(N) + DNk^3 + dN^2]$$

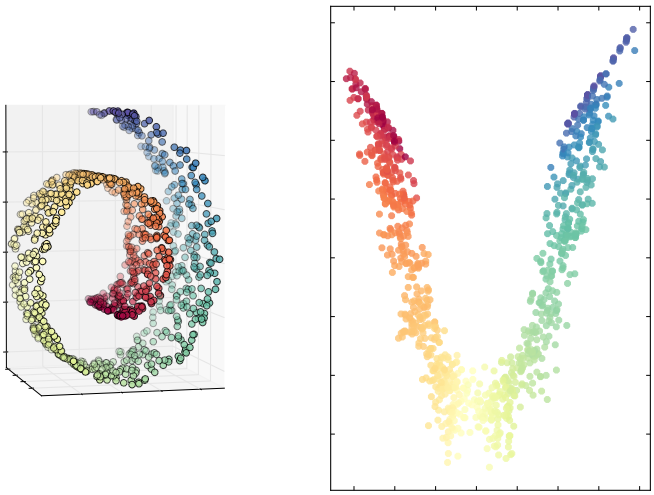


Figure 9: An example application on a 3D Swiss roll data-set with $N = 1000$ data points. The original data set is illustrated on the left in the original three-dimensional space. The Spectral Embedding projection to the two-dimensional space on the right was done, by taking a number of neighbours equal to 10. It can be seen that nearby points in the 2D embedding are also nearby points in the original 3D manifold, as desired.

2.5.3 *Spectral Embedding*

Belkin and Niyogi (2003) proposed the method of Laplacian Eigenmaps (also known as Spectral embedding) that resorts to the notion of the Laplacian operator on the neighbourhood graph of the data to computer the low dimensional embedding of the high-dimensional space. The justification of using the Laplacian is that it can be viewed as an approximation of the Laplace-Beltrami operator defined on the manifold.

The algorithm

CONSTRUCTING THE GRAPH As in ISOMAP and LLE we construct a nearest neighbour graph by choosing the nearest neighbours of a node where we put an edge between nodes i and j if they are close enough. As before we can choose between connecting two nodes if they are within a threshold ϵ or using the n nearest neighbours of each node.

CHOOSING THE WEIGHTS The are two variations of weighting the edges of the graph.

HEAT KERNEL with a parameter $t \in \mathcal{R}$. If nodes i and j are connected then:

$$W_{ij} = -\frac{\|x_i - x_j\|^2}{t}$$

which uses the Gaussian kernel function.

NAIVE APPROACH (simple minded) sets instead $W_{ij} = 1$

EIGENMAPS Assume that the graph G computed in the first step is connected, otherwise apply this step for each connected component.

The projections ϕ_i of the data points in the reduced space can be computed by minimising the function

$$\sum_{ij} \|\phi_i - \phi_j\|^2 W_{ij} \tag{2.5.5}$$

The solution of 2.5.5 is reduced to the following optimisation problem, using the Laplacian matrix D.

$$\textbf{min} \quad \operatorname{trace}(\Phi^T L \Phi) \tag{2.5.6}$$

$$\text{with respect to:} \quad \Phi^T D \Phi = \mathbf{1} \tag{2.5.7}$$

$$\Phi^T D \mathbf{1} = \mathbf{0} \tag{2.5.8}$$

The constraint in Equation 2.5.7 removes an arbitrary scaling factor in the embedding and (2.5.8) is needed to eliminate the possibility of taking the weight of 1 at each vertex ($\mathbf{1}$ is an eigenvector with eigenvalue of 0 and if the graph is connected it creates a trivial solution of collapsing all vertices of G onto the real number 1, thus we put an extra constrain of orthogonality.

To compute the solution of the optimisation problem we can compute the eigenvalues and eigenvectors for the generalised eigenvalue problem (2.5.9)

$$L\vec{y} = \lambda D\vec{y} \tag{2.5.9}$$

where D is the diagonal weight matrix, its entries are column sums of W ie.

$$Dii = \sum_j Wji$$

$L = D - W$ is then called the Laplacian matrix.

Two main advantages of Spatial Embedding is the computational efficiency and the capability of emphasising the natural clusters in the data.

Laplacian eigenmaps have successfully used in the past, especially in the fields of face (He et al., 2005) and speech recognition (Jafari and Almasganj, 2010).

Complexity

2.5.4 *Hessian Eigenmaps*

Hessian Eigenmaps (Donoho and Grimes, 2003) also known as Hessian locally linear embedding tries to solve the regularisation problem of the LLE [Section 2.5.2]. It assumes that the data lie on a manifold which is locally isometric to an open, connected subset of the euclidean space.

It is similar to the method of the Laplacian Eigenmaps, in which the graph Laplacian is replaced by an estimator of the Hessian matrix.

Algorithm

NEAREST NEIGHBOURS Construct the nearest neighbours matrix using the methods described as before.

OBTAIN TANGENT COORDINATES by performing a singular value decomposition of M^i and thus obtaining the matrices $\mathbf{U}, \mathbf{D}$ and $\mathbf{V}$

HESSIAN ESTIMATOR For least squares estimation of the Hessian (H^i) form a matrix X^i of the following columns for the case $d = 2$.

$$X^i = \begin{bmatrix} 1 & U_{\cdot,1} & U^2_{\cdot,2} & (U_{\cdot,1} \times U_{\cdot,2}) \end{bmatrix}$$

In the general case $d > 2$ we create a matrix with $1 + d + \frac{d(d+1)}{2}$ columns, the first $d + 1$ of which consist of a vector of ones, then the next d columns of U and the last $\frac{d(d+1)}{2}$ columns are the cross products and squares of those d columns. The Gram-Schmidt process is then applied to X^i to obtain the orthonormal matrix X'^i and the target Hessian $\mathbf{T_i}$ is then extracted from the last $\frac{d(d+1)}{2}$ columns of X'^i.

EIGEN-DECOMPOSITION Using $\mathbf{L} = \sum_i \mathbf{T_i}^\top \mathbf{T_i}$ the d-dimensional representation is then obtained by computing the d smallest eigenvalues of $\mathbf{L}$ and setting the projection matrix to $\Phi = \mathbf{V}\sqrt{\mathbf{V}^\top \mathbf{V}}$.

Complexity

2.5.5 *Modified Locally Linear Embedding*

It is widely reported that LLE can fail as its local geometry exploited by the reconstruction weights is not well-determined, since the constrained least squares problem involved for determining the local weights may be ill-posed. A solution to this is to use Tikhonov regulisation (as known as ridge regression in statistics), but it may be the case that a regularised solution may be not a good approximation to the exact solution if the regularisation parameter is not suitably selected.

Zhang and Wang (2006) resolves the issue of regularisation of LLE by making use of multiple local weight vectors. It can be shown these linearly independent

Modified LLE Algorithm

The procedure can be summarised in the following steps:

1. For each data point i such that $1 \leqslant i \leqslant N$
   a) Compute the neighbour set N_i using k-NN or ϵ-NN
   b) Compute the regularised solution $w_i(\gamma)$ by using $G^T G + \gamma \|G\|_F^2 I)y = \mathbf{1}_k, w = y/\mathbf{1}_k^T y$
   c) Compute the eigenvalues $\lambda_1^{(i)}, \ldots, \lambda_k^{(i)}$ and set $\rho_i = \sum_{j=d+1}^{k} \lambda_j^{(i)} / \sum_{j=1}^{(i)} \lambda_j^{(i)}$.
2. Sort ρ_i, $1 \leqslant i \leqslant N$ in increasing order and set $\eta = \rho_{\pi_{[N/2]}}$
3. For each data point i such that $1 \leqslant i \leqslant N$
   a) Set $s_i = \max_l \left\{ l \leqslant k_i - d, \frac{\sum \lambda_j^{(i)}}{\sum \lambda_j^{(i)}} \right\} < \eta$ and let V_i be the eigen vectors
4. Compute the $d+1$ smallest eigenvectors of Φ set the projection Π to be

$$\Pi = \begin{pmatrix} \vec{u}_2 \\ \vdots \\ \vec{u}_{d+1} \end{pmatrix}$$

weights can be approximately optimal to the true embedding of the data in the smaller dimensionality space.

Complexity

NEAREST NEIGHBOURS SEARCH Same as before in ISOMAP and LLE. $\mathcal{O}[Dlog(k)Nlog(N)]$

WEIGHT MATRIX CONSTRUCTION

EIGENVALUE DECOMPOSITION Same as ISOMAP and LLE. $\mathcal{O}[DNk^3]$

This the overall complexity of MLLE is characterised by:

$$\mathcal{O}[Dlog(k) \times Nlog(N) + DNk^3 + dN^2]$$

where
D: Number of dimensions
N: Number of data points
k: Number of nearest neighbours to search for
d: The number of the final reduced dimensions.

Note that the computational cost of MLLE is approximately the same as that of LLE.

2.6 EVALUATION METHODS

2.6.1 *Adjusted Rand Index*

Due to the small amount of samples, there is a difficulty obtaining balanced samples, e.g. same amount of cancerous and non-cancerous samples, which makes comparing clustering outcomes difficult. For example if 10 patients have cancer and 90 do not, and our leave-one out prediction accuracy is 85% we perform worse than a method which predicts an outcome based on chance.

To take into account this observation, that is, the necessity of correction for chance for information theoretic based measures for clustering, (Vinh et al., 2009) proposed the Adjusted Rand Index (ARI). This measure was derived by assuming a hypergeometric model of randomness, and has several advantages over other measures.

- It has a bounded range of $[-1, 1]$ where values close to 1 indicate a good match score (with 1.0 being the absolute perfect match score) and values closer to -1.0 a bad match score.
- Random assignment have a score close to 0.0
- No assumption was made on the cluster structure, which is preferred for gene clustering.

Mathematical Formulation

The adjusted form of the Rand Index is defined as:

$$AdjustedIndex = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex}$$

If $\mathcal{O}$ are the ground truth class members and Ψ the derived clustering the unadjusted Rand Index is defined by:

$$RI = \frac{a + b}{C_2^{n_{samples}}}$$

where
a is the number of pairs of elements that are in the same set in $\mathcal{O}$ and in the same set in Ψ.
b is the number of pairs of elements that are in different sets in $\mathcal{O}$ and in different sets in Ψ. $C_2^{n_{samples}}$ is the number of all possible pairs in the data-set.

To take into account the fact that random label assignments should have a value close to zero, we discount the expected RI of random labels.

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}$$

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}$$

2.6.2 *Dunn Index*

The Dunn index (DI) (Dunn, 1973) is a metric for evaluating clustering algorithms which tries to identify compact (but yet 'rich' in features) clusters that are well separated with the means of different clusters sufficiently far apart compared to the within cluster variance.

$$\Delta_i = \frac{\sum_{x \in C_i} d(x, \mu)}{|C_i|} \qquad \mu = \frac{\sum_{x \in C_i} x}{|C_i|} \tag{2.6.1}$$

$$DI_m = \min_{1 \leqslant i \leqslant m} \left\{ \min_{1 \leqslant j \leqslant m, j \neq i} \left\{ \frac{\delta(C_i, C_j)}{\max_{1 \leqslant k \leqslant m} \Delta_k} \right\} \right\}$$

where $\delta(C_i, C_j)$ is the intercluster distance metric, between clusters C_i and C_j.

2.6.3 *Davies-Bouldin Index*

Davies-Bouldin Index (DBI) is another metric for evaluating how good a clustering scheme is. Davies and Bouldin (1979) claims that it can be used to infer the appropriateness of data partitions and therefore be used to compare relative appropriateness of various divisions of data.

If the distance between clusters remains constant while the dispersions increase then the similarity should increase as well. It is suggested that using Equation 2.6.2

$$R_{ij} = \frac{S_i + S_j}{M_{ij}} \tag{2.6.2}$$

where M_{ij} is the distance between vectors of clusters i and j.

$$M_{i,j} = ||A_i - A_j||_p = \sqrt[p]{\sum_{k=1}^{n} |a_{k,i} - a_{k,j}|^p} \tag{2.6.3}$$

and S_i is a measure of scatter within the cluster which characterised by the distance from the centroid of the cluster and the individual feature vectors. In Equation 2.6.4 when $q = 2$ we would have the euclidean distance function, but this can generalise easily to any distance metric according to the application.

$$S_i = \sqrt[q]{\frac{1}{T_i}\sum_{j=1}^{T_i} |X_j - A_i|^q} \tag{2.6.4}$$

According to Equation 2.6.2 if we have a large intercluster distance and small intracluster index then we will have a low $R_{i,j}$ value. Thus we can define the Davies-Bouldin Index as:

$$DB = \frac{1}{N}\sum_{i=1}^{N} \max_{j:i\neq j} R_{i,j}$$

which is the average (in this case) worst $R_{i,j}$ for each cluster i.

2.6.4 *Silhouette index*

Rousseeuw (1987) proposed a metric which can be used to evaluate clustering validity, and to select an 'appropriate' number of clusters for algorithms such as the k-means method.

Each cluster is represented by a *silhouette* based on the comparison of its tightness and separation. This shows how many and which members of the cluster lie within it, and which lie outside between other clusters.

The silhouette of a cluster is then defined as:

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{if } a(i) < b(i) \\ 0, & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{if } a(i) > b(i) \end{cases}$$

which can be simplified into one equation:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

3

VISUALISING USING MANIFOLD LEARNING

3.1 ACUTE LYMPHOBLASTIC LEUKAEMIA

Acute lymphoblastic leukaemia (ALL) is a form of leukaemia characterised by excess lymphoblasts.

There are two main types of acute leukaemia are T-cell ALL and B-cell ALL. *T-Cell* acute leukaemia is aggressive and progresses quickly but is more common in older children and teenagers.

B-Cell ALL leukaemia is another type of ALL, originated in a single cell and characterised by the accumulation of blast cells that are phenomenologically reminiscent of normal stages of B-cell differentiation (Cobaleda and Sanchez-Garcia, 2009).

To show off the potential of manifold learning we will first apply the techniques on a microarray data-set which was first used by Brunet et al. (2004). The paper uses the popular dimensionality reduction technique Non-negative matrix factorisation (NMF) which was found to have huge success in lowering the dimensionality of microarrays.

3.1.1 *ISOMAP*

The data set includes 38 samples where each one has a dimensionality space of 5000 probes. Using the ISOMAP algorithm, we construct a Euclidean distance matrix of the distances between the nearest neighbours of each sample. To decide which are the nearest neighbours of each sample one can use either ϵ-nearest neighbours or k-nearest neighbours. Using ϵ-nearest neighbours it can be thought of a more realistic approach as it chooses only points which are in close proximity to each node, whereas k-NN may choose points which are not quite close to the node if the node is isolated. The problem with ϵ-NN though, is that often is quite difficult to choose the right threshold value ϵ and more over can lead to disconnected neighbour graphs which cause problems with manifold learning techniques. If such a case arises, that is if the neighbourhood graph consists of two or more connected components, then we could treat each component as different and analyse each one separately. Doing so though we lose information about the global structure of the manifold which is undesired.

Taking $k = 3$ number of neighbours we can reduce the previous features of the data down to dimensions, while keeping well defined clusters between the two different types of ALL and normal behaviour samples as seen in Figure 10.

We can see that ISOMAP can reduce the dimensionality of the initial dataset very well, while keeping the three given clusters (Normal, T-cell ALL, and B-cell ALL) well separated. *What does it mean to have well defined clusters though? What is the best way to evaluate our lower-dimensional embedding?*

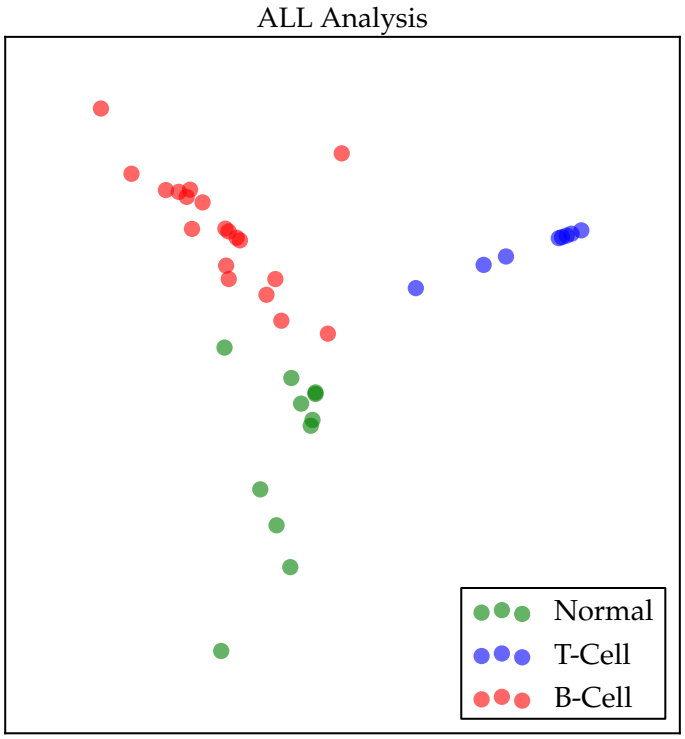


Figure 10: Example application of the ISOMAP algorithm on the ALL data-set using number of neighbours $k = 3$

One of the advantages of using unsupervised learning techniques such as the algorithms of manifold learning we use is that it is quite difficult to over-fit our algorithms onto the dataset as we are not using the labels to reduce the dimensionality of our dataset.

3.1.2 *Evaluation*

A way to reason how good the lower-dimensional embedding is, would be to use the eigenvalues used in its calculation (Figure 11) as larger eigenvalues indicate the importance of including the corresponding eigenvector into the embedding when using the ISOMAP algorithm (as a larger eigenvalue means a smaller reconstruction error). This way we can determine approximately a good cut-off for the number of components to use, but also whether the information of the microarray can actually be represented in a smaller number of dimensions.

Unfortunately, this method does not allow us to compare different algorithms together, nor does it give us a way to tell if the embedding gives us useful information. Instead we will use the clinical information of each patient and see if we can deduce any useful pattern in the lower-dimensionality embedding, hopefully by the form of clusters that distinguish the different types of cancer in the ALL dataset.

The solution to this is to make use of the k-NN algorithm which can be fine tuned quite easily as it has only one free parameter. Although SVM methods are generally

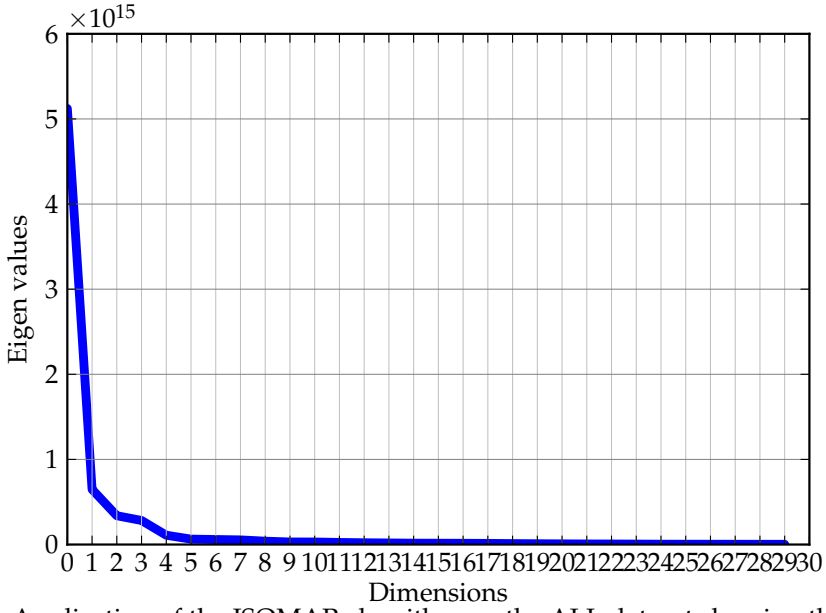


Figure 11: Application of the ISOMAP algorithm on the ALL dataset showing the change of eigenvalues according to the dimensionality of the embedding

less computationally expensive methods and simpler than k-NN, k-NN can deliver better results in most applications although is more prone to over-fitting, which is not important, as we would like the best possible classification in possible in each case to be able to compare and contrast each of the algorithms. Additionally as the resulting embedding lives in the Euclidean space gives us more reasons to choose the k-NN classifier.

Therefore, in order to assess the results we make use of the k-nearest neighbours classifier using the Adjusted Rand Index (ARI) measure as defined in Section 2.6.1. ARI lets us easily compare different microarray datasets, which are often hugely imbalanced as we can have many more non-cancerous, than cancerous tissue samples which can mislead someone into thinking that the algorithm performs better than it actually is. For an example if a dataset contains 900 normal and 100 prostate samples then a classifier that assigned random labels would yield an average of 90% accuracy, whereas the corresponding ARI value would be 0.

Of course in the case several algorithms can perform similarly and produce almost identical classification accuracies, what would be the best one to choose from then? Traditional machine learning techniques such as cross-validation for validating the model cannot be very reliable due to the small number of samples available. In conjunction to these techniques one can use cluster validation measures.

It is extremely important to be able to evaluate the results of the classification process correctly. This can be troublesome as there the number of samples available is limited so it is difficult to ensure that assess the quality of the clustering in gene expression data.

3.1.3 *Evaluation process*

In order to be able to evaluate properly the results of our system we are proposing to use the following benchmark system as illustrated in Figure 12. In the first step

of dimensionality reduction we have to fit the parameter of k nearest neighbours to our algorithm of choice (ISOMAP, LLE, etc.) needed for computing the nearest neighbours graph. This k (not to be confused with the parameter k of the classifier) will be chosen according to the classification accuracy and cluster validation metrics as follows:

In *Classification Accuracy* we will be using the k-NN classifier which we will fit after a 10-fold validation onto the embedding produced from the manifold learning process. In 10-fold cross validation the embedding produced gets partitioned in 10 subsets, one of them is used for testing and the other 9 are used as the training data. The process is repeated 10 times so that every subset is used as validation exactly once. The results are averaged along the 10 times the algorithm run and a single estimation is produced. Using these results we can estimate the optimal value for k and thus use it to measure the ARI index using Leave-One-Out (LOO) cross-validation.

Additionally to the classification accuracy, it is also useful to use *cluster validation* indexes. As the samples are usually of a small number, it will not be unusual to get almost identical ARI values, so we would want to choose the embedding that most separates the different classes of our data set.

Intuitively we would like the clusters generated by the clustering process to be easily separable with each cluster 'as far' as possible from every other cluster so the classification algorithm can provide as good results as possible. In the case of ALL we want the three classes (Normal, T-cell ALL, and B-cell ALL) well separated.

As we know the labels of each sample in the resulting embedding we will use the Dunn Index (2.6.2), the Davies-Bouldin Index (2.6.3) and the Silhouette Index (2.6.4) as indication of how well separated the clusters are. Note these indices are not an absolute measure of how good the embedding is, rather than a guide we will use when the ARI measures are really close with each other. This happens as the values reported from these indices are heavily influenced by the use of distance metric (Euclidean, Manhattan, Mahalanobis etc.), and the type of intra and inter cluster measure it uses for combining the information (average, centroid, complete, single etc.). We will use the Euclidean distance metric along with the use of the *centroid* for intra-cluster measures and the *average* distance for the intra-cluster measures.

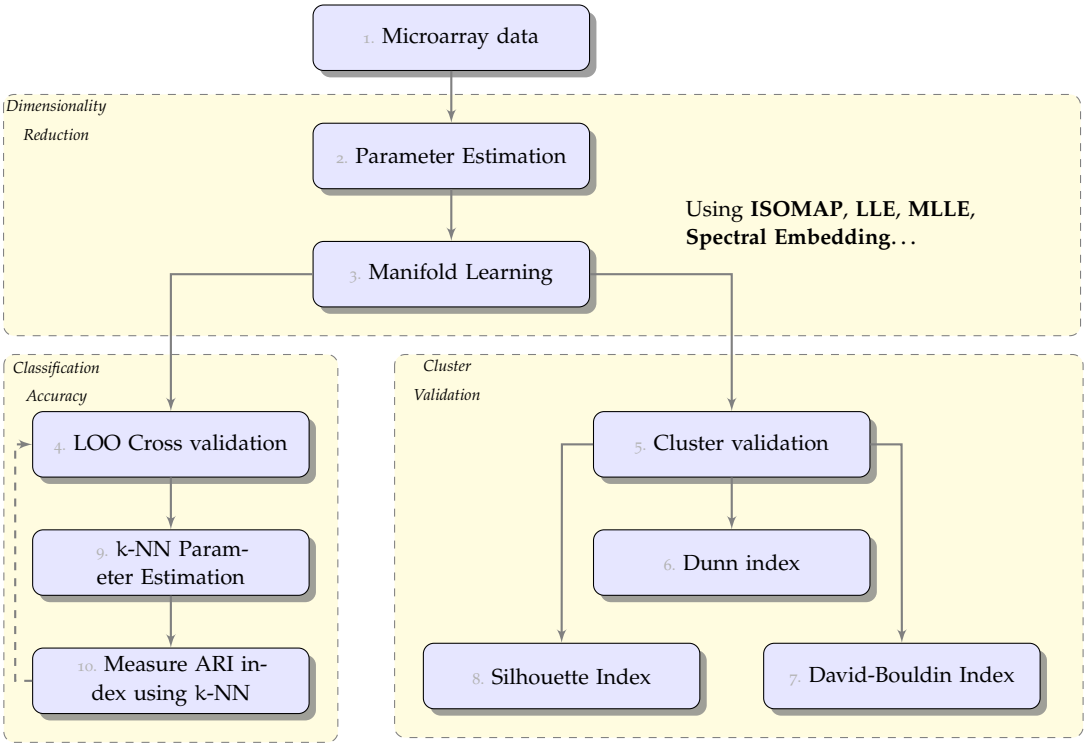


Figure 12: The evaluation process for dimensionality reduction, including two ways of evaluation; classification accuracy and cluster validation

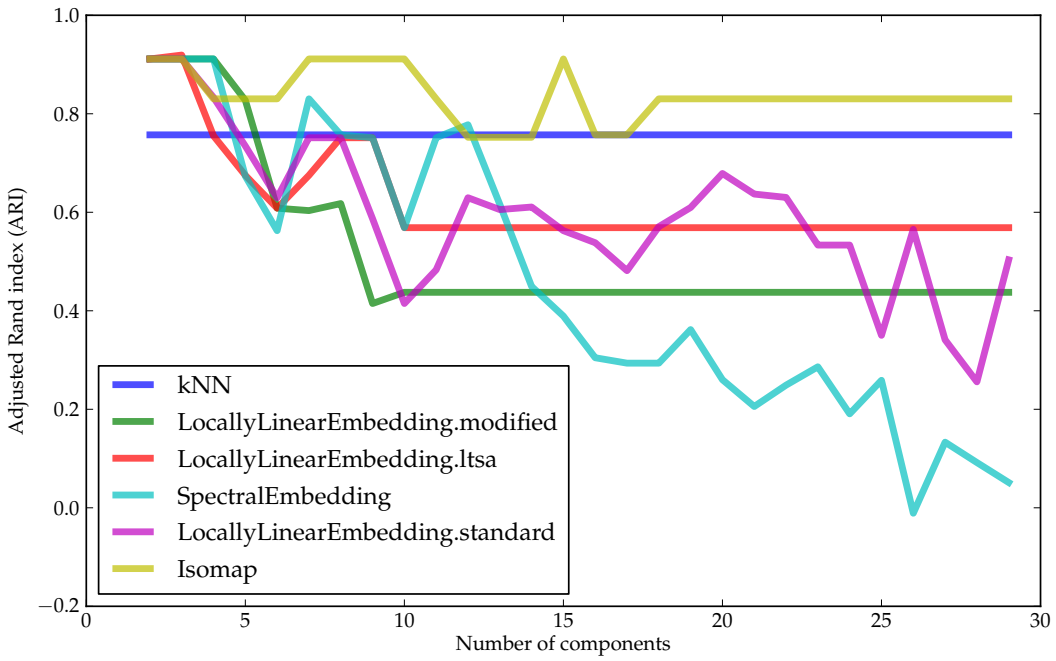


Figure 13: Visualisation of the performance of each of the algorithms compared to k-NN for the ALL dataset. The ARI index was measured using leave-one-out cross-validation using the optimal number of neighbours for each case

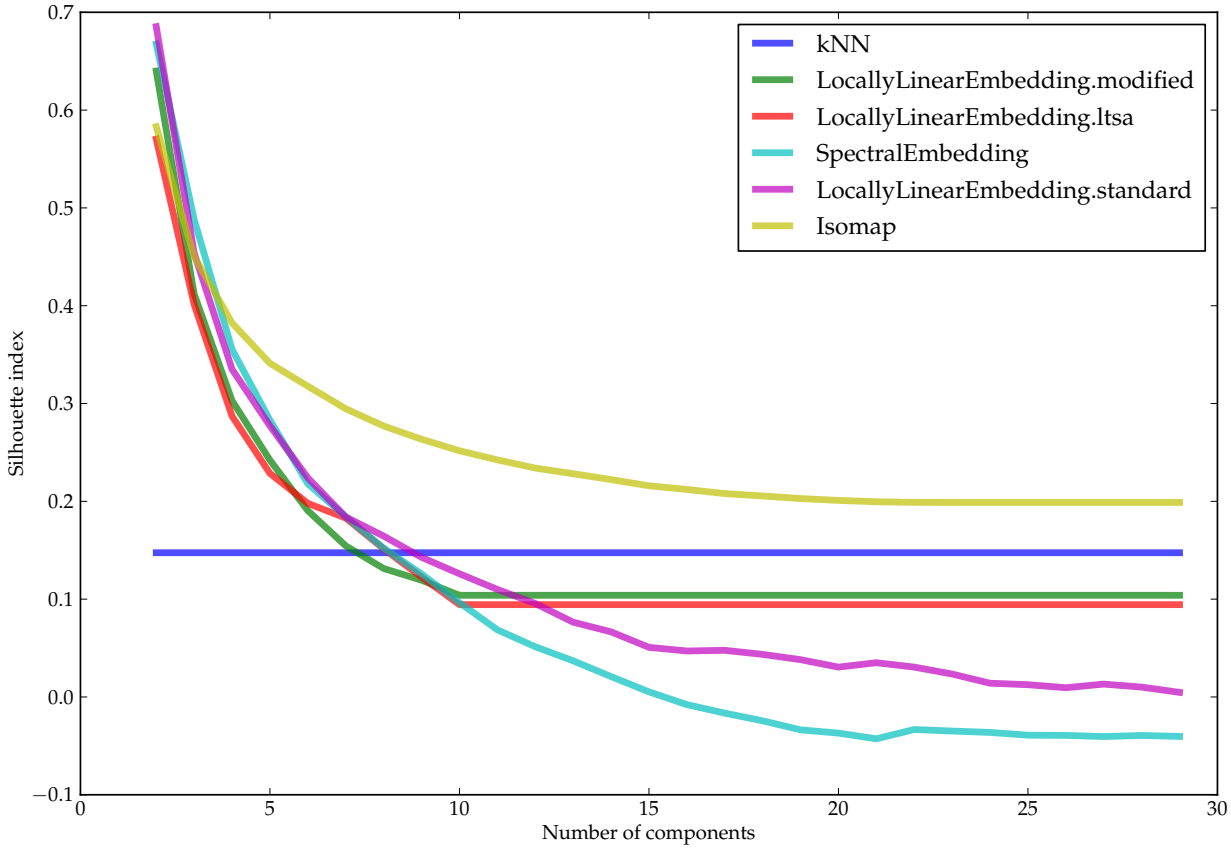


Figure 14: Visualisation of the performance of each of the algorithms compared to k-NN with a variable number of dimensions. The Silhouette index was measured using leave-one-out cross-validation using the optimal number of neighbours for each algorithm

3.2 OVARIAN CANCER

As described in Section 1.1.1, Ovarian Cancer is one of the few types of cancer that is difficult to diagnose, so our aim is to apply the manifold learning techniques to help uncover more information from the microarray data. For this purpose we will use two well-annotated large microarray datasets.

3.2.1 *Tothill et. al microarray publication*

Tothill et al. (2008) used a microarray gene expression profiling of 285 serous and endometrioid tumours of the ovary. Their technique compromised of unsupervised K-means clustering. To evaluate the clustering they concentrated on the patient survival analysis within the identified K-means groups using Cox proportional hazards models (Breslow, 1975). Instead of using patient survival probability analysis using the Cox method, we will attempt to identify other critical details such as Grade or the survival status of the patient.

First, using ISOMAP we reduce the dimensionality of our data set from 285×54621 down to a lower dimension $285 \times d$. Again using the visualisation of the eigenvalues Figure 15 we can tell that most of the information about our dataset can be contained in a small number of dimensions, as we would want.

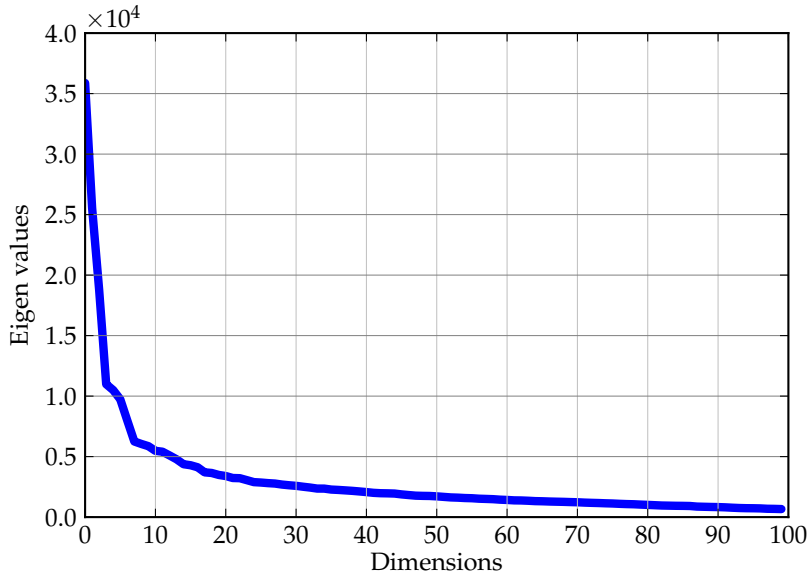


Figure 15: Application of the ISOMAP algorithm on the microarray Tothill et. al paper using number neighbours $k = 50$ showing the change of eigenvalues according to the dimensionality of the embedding

Next we will attempt, using the clinical data of the patients to use the k-NN classifier, using LOO validation, to see if using manifold learning to reduce the dimensionality of the data helps the classification process. Using ISOMAP ($k = 5$), we have managed to reduce the dimensionality of the data down to 5 dimensions where we had the best classification rates as depicted in Table 3. Although using manifold learning for dimensionality reduction clearly outperforms just using k-NN classifier, we could not obtain a good enough representation of the dataset to get good classification rates. The fact that using the ISOMAP improves this much the learning abilities of our classifier suggests that approximates quite well the data, and removes some of the noise present in the data.

Field	ISOMAP (k = 5, d = 5)		k-NN
	ARI	**Accuracy**	**ARI**
Arrayed Site	0.114	0.94	4.33e-14
Grade	0.07	0.45	-0.003
Histologic Subtype	0.1	0.60	0.08
Patient survival status	0.064	0.71	0.02
Primary Site	0.067	0.64	0.038
Residual Disease	0.056	0.86	3.6e-14
Stage	0.041	0.72	-0.002
Type	0.159	0.32	0.03

Table 3: Shows the classification rates of clinical data of the Tothill et al. dataset with and without the application of ISOMAP, using k-NN and Leave-One-Out classification

3.2.2 *The Cancer Genome Atlas Research Network (TCGA)*

TCGA presented an analysis of m-RNA expression, microRNA expression, promoter methylation and DNA copy number in 489 high-grade serous ovarian adenocarcinomas and the DNA sequences of exons from coding genes in 316 of these tumours (TCGA, 2011).

Again in a similar manner as in 3.2.1 we can plot the eigenvalues of the embedding using the ISOMAP algorithm. This time though, it seems that we need much more dimensions to represent our data in (Figure 16). Using the same procedure as for the Tothill microarray dataset we will use the k-NN classifier on the 20 dimensional embedding computed with ISOMAP (k = 5) and the original dataset (Table 4). Again although we can see minor improvements we obtain better results than without the use of manifold learning, which is quite promising.

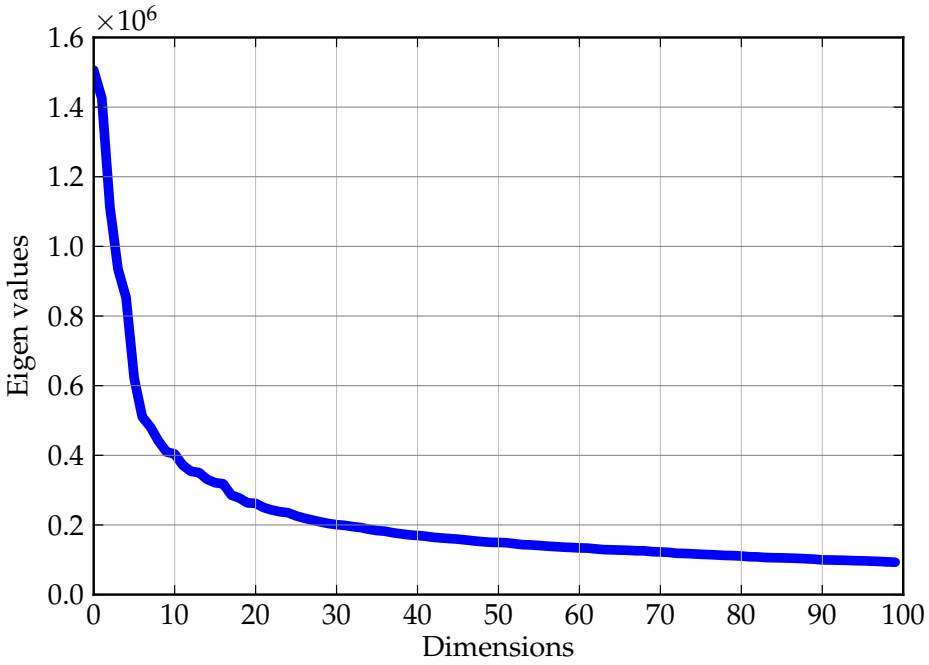


Figure 16: Eigenvalue distribution according to dimensionality on the TCGA Ovarian Cancer dataset.

Field	ISOMAP (k = 5, d = 20)		k-NN
	ARI	Accuracy	ARI
Platinum Status	0.046	0.443	0.009
Progression Free Status	0.104	0.691	0.000
Tumour Residual Decease	0.060	0.459	0.014
Event Relapse	0.104	0.691	0.000
Event Death	0.054	0.589	0.018
Tumour Grade	0.222	0.864	0.183
Person Neoplasm Cancer Status	0.092	0.626	0.000
Vital Status	0.054	0.589	0.018

Table 4: Classification rates on the TCGA Ovarian Cancer dataset with and without the use of manifold learning. The k-NN classifier was used with Leave-One-Out classification

4

A PRIORI MANIFOLD LEARNING

4.1 BIOLOGICAL PATHWAYS

Biological pathways represent the biological reactions, which are identified with enzymes (which are in turn encoded onto genes) and interaction network in a cell.

Researchers find most of the biological pathways through laboratory studies of bacteria, fruit flies, mice and other organisms. Fortunately most of these model systems have been found to have similar counterparts in the human organism as well.

Still, it is not well understood how these pathways work together. This is quite an open problem in the area of genetics and bioinformatics that will take a lot more years to fully understand it (NHGRI, 2012).

Biological pathways, technically, are usually directed graphs with labelled nodes and edges representing associations of genes participating in a biological process. Using this information we can determine which genes are most significant for the development of a certain type of cancer by identifying genes that are participating in such a process. This information can be used as background knowledge to either reduce the initial data set or either pay more attention (by attributing weights) to those genes during the clustering process.

KEGG Kyoto Encyclopedia of Genes and Genomes (KEGG) provides data primarily centred on biological pathways. Each pathway is associated with a name (i.e. Prostate cancer metabolism Figure 17) and an identifier and also genes are associated with a pathway for each species.

4.2 BIOLOGICAL PATHWAY BASED WEIGHTING

Of course due to the technically difficult and time consuming task to find these pathways there are not always available to us, so instead we propose a more general scheme for how to use them as background knowledge.

Our intuition is that some genes tend to co-express when are participating in the same biological process and these pairs of genes are more likely to display co-expression in other biological processes. That is, the more information we have that a gene displayed c Thus we can use the information to create weights in order to favour genes that participated in the same biological pathway.

Using the annotation data of a genome array we can associate each probe participating with a KEGG pathway. The mappings for each probe are based on the

1 http://www.genome.jp/kegg-bin/show_pathway?map=hsa05215
2 http://pinguin.biologie.uni-jena.de/bioinformatik/

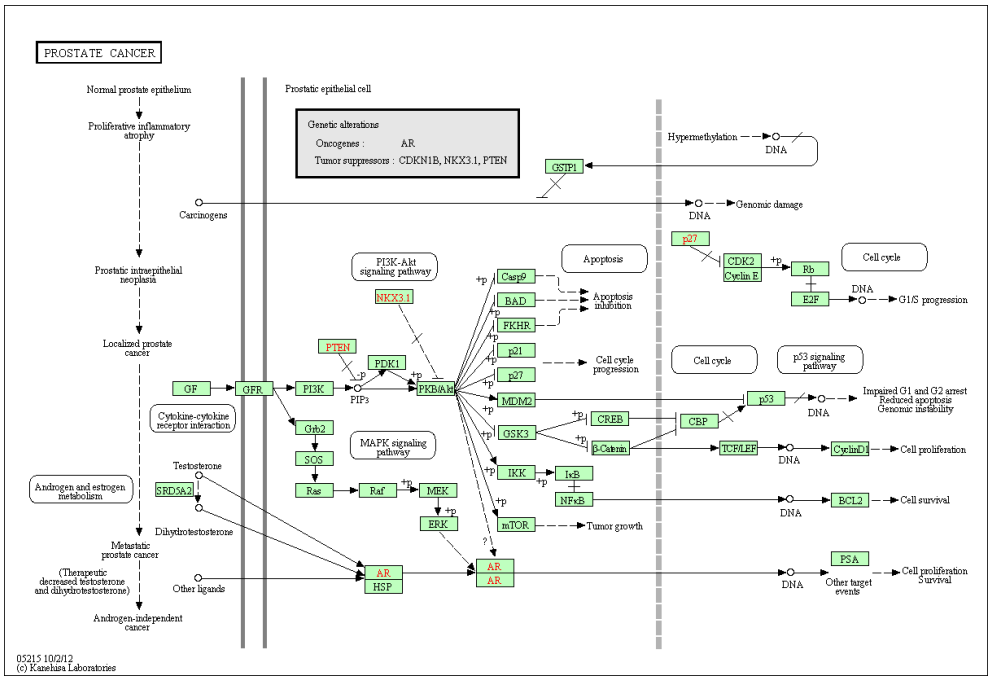


Figure 17: The biological pathway of prostate cancer of Homo Sapiens as illustrated on KEGG[1].

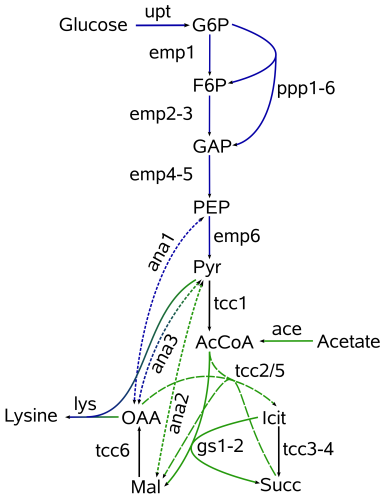


Figure 18: Demonstration of a metabolic pathway[2]

data provided by KEGG[3] and then associated with the annotations provided by the manufacturer of each microarray.

The simple explanation is that we have each probe of the microarray to be associated with a number of pathways (4.2.1).

$$\xi\colon G \to \mathcal{P}(\Pi) \tag{4.2.1}$$

3 ftp://ftp.genome.jp/pub/kegg/genomes/genome

Jaccard Index

Given a pair of probes we would like to evaluate the similarity of pathways they share together. A suitable measure for this will be to use the Jaccard coefficient. This index coined by Paul Jaccard is a statistic commonly used for comparing similarity and diversity of sample sets.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.2.2}$$

Therefore using the Jaccard Index we can define the pathway similarity of two probes i and j as:

$$R(i, j) = \frac{|\xi(i) \cap \xi(j)|}{|\xi(i) \cup \xi(j)|} \tag{4.2.3}$$

Of course we do not know how much influence should the weighting have on the prediction of the nearest neighbours (NN) needed for the construction of the manifold. Therefore (4.2.4) uses a learning parameter η which is used to maximise (minimise) the influence of the biological pathways prior knowledge to the NN learning process. When η changes in the dataset are exponentially reflected on the weights.

$$w_{ij} = exp(-\eta \times R(i, j)) \tag{4.2.4}$$

Exploiting the use of these weights we can modify the classical k-NN algorithm using the weighted Euclidean (4.2.5) as a distance metric for determining which points of the original data space are close to one another.

$$D(a, b) = \sqrt{w_{ij} \times \sum_{i=1}^{n} (a_i - b_i)^2} \tag{4.2.5}$$

Data: gene expressions geneData, biological pathways map ξ
Result: k-nearest neighbours of each probe
initialisation;
for *each probe* i *in probes* **do**
  for *each probe* j *in probes* **do**
    $R(i,j) = \frac{|\xi(i) \cap \xi(j)|}{|\xi(i) \cup \xi(j)|}$
  end
end
$R = R / \sum_{i,j} R(i,j)$
for *each probe* i *in probes* **do**
  for *each probe* j *in probes* **do**
    $w_{ij} = \exp(-\eta \times R(i,j))$
    $\text{distances}(i,j) = \sqrt{w_{ij} \times \sum_{i=1}^{n} (a_i - b_i)^2}$
  end
  nearestNeighbours(i) = sorted(distances (i))
end

Algorithm 1: Calculation of the k-Nearest neighbours of the manifold

After having the k-nearest neighbours of each probe we can calculate the euclidean distances between each probe of the data set.

Using the pairwise distances between the genes we can calculate the geodesic distances (i.e. the shortest paths) and then the eigen values, eigenvectors and thus obtain the embedding in the gene-to-gene space using the steps of the standard ISOMAP algorithm. We would like to project the resulting embedding to sample-to-sample space, in order to be able to evaluate our algorithm and visualise the results by viewing how the patients (samples) are distributed on the plane. To do so we can use the equation (4.2.6) to project the gene-to-gene space back to the sample-to-sample space.

$$\Pi_{S \times S} = \textbf{expressionData} \cdot \Pi_{G \times G} \tag{4.2.6}$$

4.2.1 *Choosing the optimal weights*

In order to find the optimal weight parameter η we will be using the benchmarking system as proposed before for various values of η. In Figure 19 we can see runs of the benchmarking suite for $0 \leqslant \eta \leqslant 80000$.

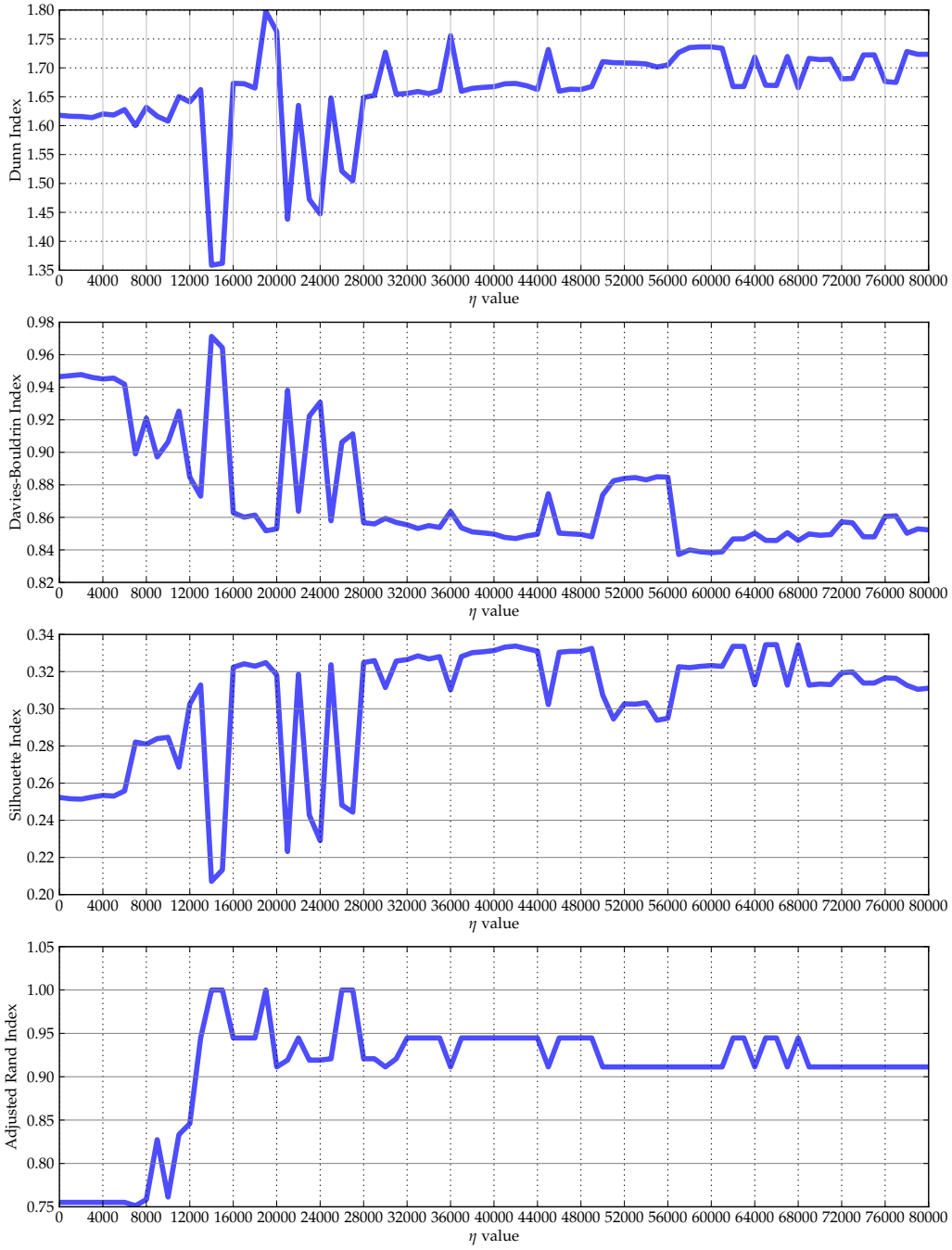


Figure 19: Choosing a suitable η value using the ALL microarray dataset. Note that in contrast with the other measures a low Davies-Bouldrin Index is preferred to a high one.

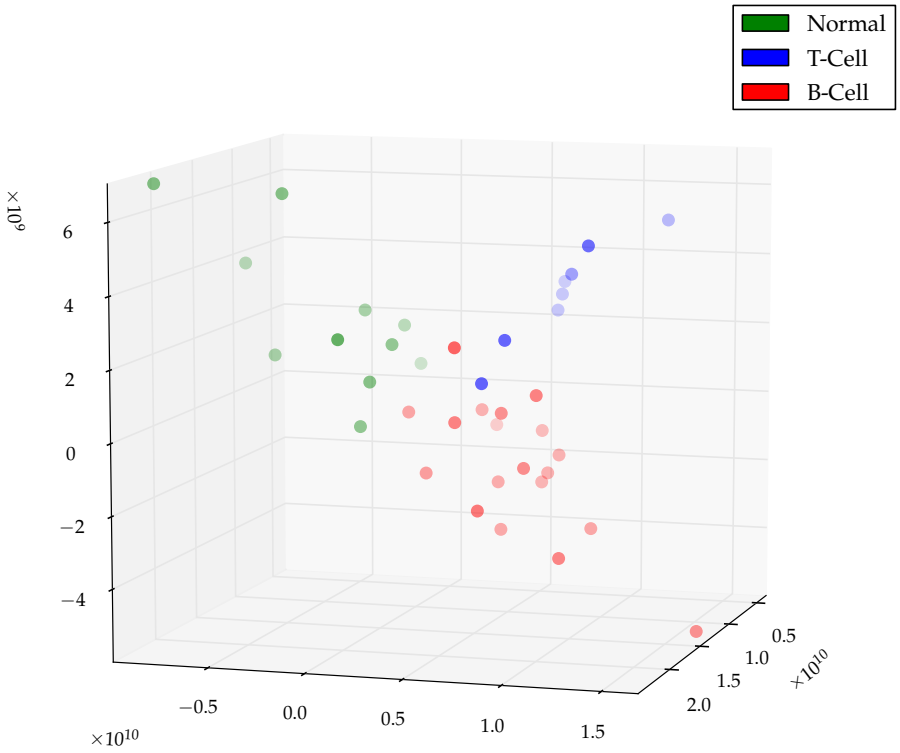


Figure 20: Visualisation of the ALL microarray dataset using *a priori* learning in three dimensions ($\eta = 2.6e4, k = 5$)

5

EVALUATION

In the previous chapters we have proposed two areas of techniques that can be used on the analysis and visualisation of microarray data. To confirm the results we will be using the ALL microarray dataset to compare and contrast ISOMAP with *a priori* ISOMAP and then use additional larger microarray datasets to replicate the procedure and confirm the results, while comparing it to current state of art techniques.

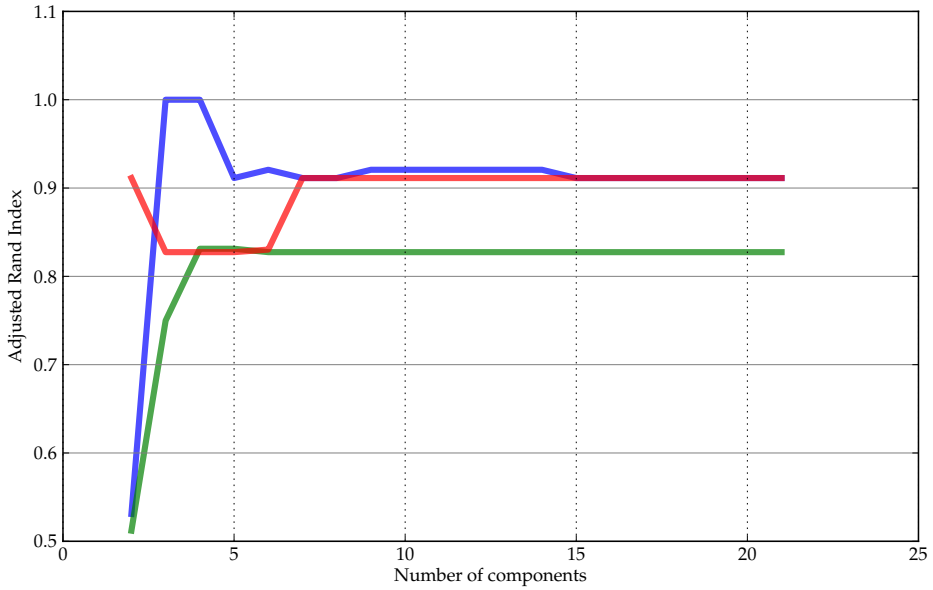


Figure 21: Comparison of the ISOMAP algorithm on the ALL dataset.
Blue: *a priori* manifold learning ($\eta = 2.6e4, k = 5$)
Green: classical ISOMAP algorithm ($k = 5$)
Red: ISOMAP using sample-to-sample distance matrix ($k = 3$)

Firstly we will compare the performance of the microarray dataset of ALL. From Figure 21 we can see that *a priori* manifold learning clearly outperforms applications of the ISOMAP algorithm on the dataset on all measures and additionally it reaches an ARI of 1.0 thus giving a perfect LOO score.

When we apply the ISOMAP algorithm using as observations the patient samples (sample-space), it clearly delivers much better results than regarding each gene as a different observation (gene-space). The reason for this could be attributed to the fact that the patients are more likely to be sampled from a low-dimensional manifold, than genes are, or the assumption of a Euclidean space is completely off. Fortunately we gain much better classification accuracy to the true labels using the *a priori* knowledge, which is encouraging.

Using the optimal settings obtained from the ARI values we can plot the corresponding cluster validation measures, in Figure 22. Although the measures of cluster validation seem to indicate that a sample-space application should be preferred, the actual classification accuracy disagrees, confirming the drawback of clustering indices where a 'good' value may not necessarily imply the best information retrieval.

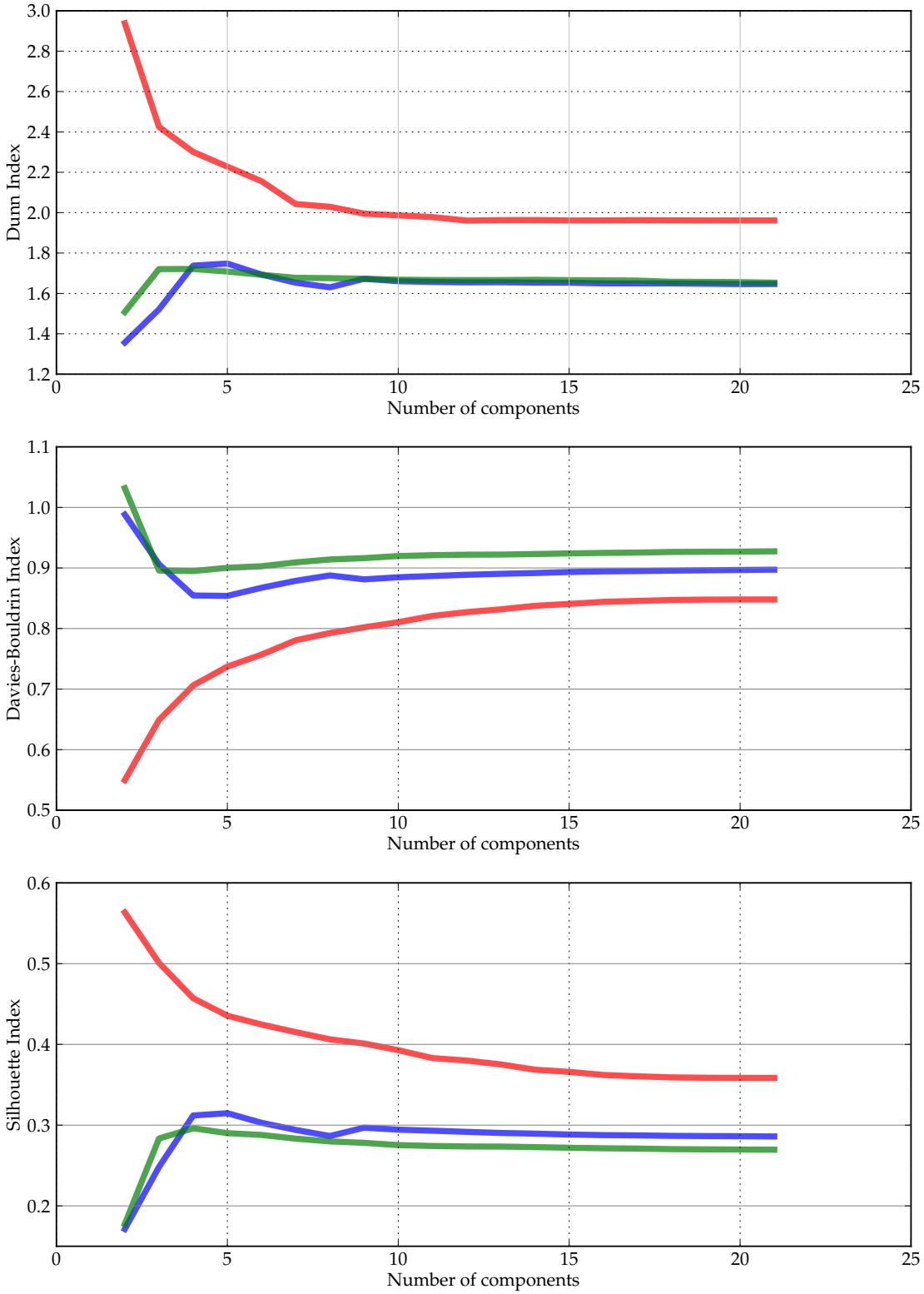


Figure 22: Comparison of the ISOMAP algorithm.
Blue: *a priori* manifold learning ($\eta = 2.6e4, k = 5$)
Green: classical ISOMAP algorithm ($k = 5$)
Red: ISOMAP using sample-to-sample distance matrix ($k = 3$)

5.0.2 *GEMLeR – Gene Expression Machine Learning Repository*

Having established some initial intuition about the applicability of manifold techniques on the ALL microarray dataset we will expand our knowledge to microrray data found from Gene Expression Machine Learning Repository (GEMLeR). GEMLeR provides a collection of gene expression datasets that can be used for benchmarking gene expression oriented machine learning algorithms. Each of the gene expression samples in GEMLeR came from a large publicly available repository named Expression Project For Ontology (expO)[1]. *expO* was mainly preferred as:

- The processing procedure of tissue samples is consistent
- The same Affymetrix microarray assay platform is used (Affymetrix GeneChip U133 Plus 2.0)
- There is large number of samples for different tumour types
- Availability of additional information for combined genotype-phenotype studies

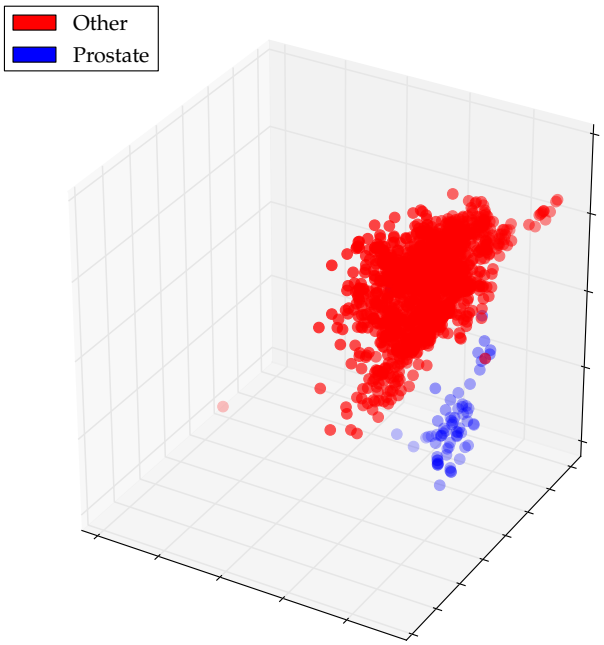


Figure 23: Application of manifold learning (ISOMAP; $k = 3$) on a dataset of 69 prostate patients compared to 8 other cancer types, each containing 10935 gene expression measurements reducing the dimensionality of the data space to three dimensions. We can clearly distinguish two clusters; those of prostate tissue samples and the rest of the tissue samples.

In order to lower memory requirements and allow faster computation times we made use of a shorter version of each gene expression dataset containing only 10935

1 http://www.intgen.org/expo/

probes, instead of 54681. To achieve this, a method of unsupervised highest variance filter was applied to avoid the so called "selection bias" and thus be able to remove genes that have a practically constant signal.

We applied the manifold learning ISOMAP algorithm on the 9 datasets and compared it to the results provided by GEMLeR which used a Support Vector Machine (SVM) classifier and another state of the art feature reduction algorithm (Support Vector Machines - Recursive Feature Elimination (SVM-RME)).

1. Breast cancer (344 samples) vs. other
2. Colon cancer (286 samples) vs. other
3. Kidney cancer (260 samples) vs. other
4. Ovary cancer (198 samples) vs. other
5. Lung cancer (126 samples) vs. other
6. Uterus cancer (124 samples) vs. other
7. Omentum cancer (77 samples) vs. other
8. Prostate cancer (69 samples) vs. other
9. Endometrium cancer (61 samples) vs. other

In figures 24 and 25 we can see that the manifold learning ISOMAP does as well as the SVM-RME feature reduction. Although using a basic k-NN classifier, we can see that our dimensionality reduction techniques perform as well as state of the art techniques used already (Stiglic and Kokol, 2010). In order to compare more accurately the two different approaches of dimensionality reduction it would be more appropriate to use a paired t-test, or a Wilcoxon signed-ranks test, as recommended by Demsar (2006), which is a non-parametric alternative to paired t-test.

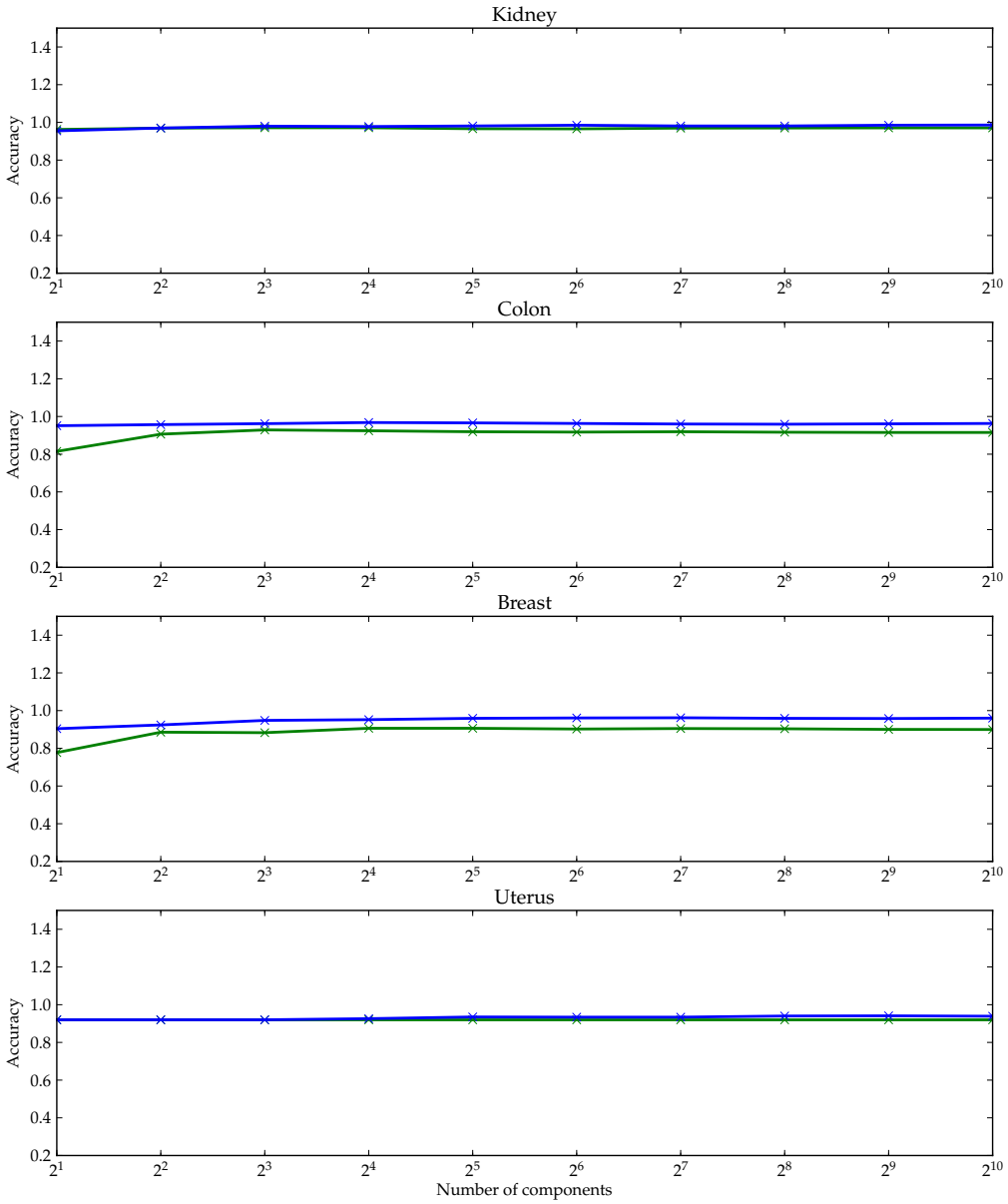


Figure 24: The accuracy on the GEMLeR dataset against the change in dimensionality of the SVM-RFE (blue) against ISOMAP (green) with $k = 5$ dimensionality reduction methods.

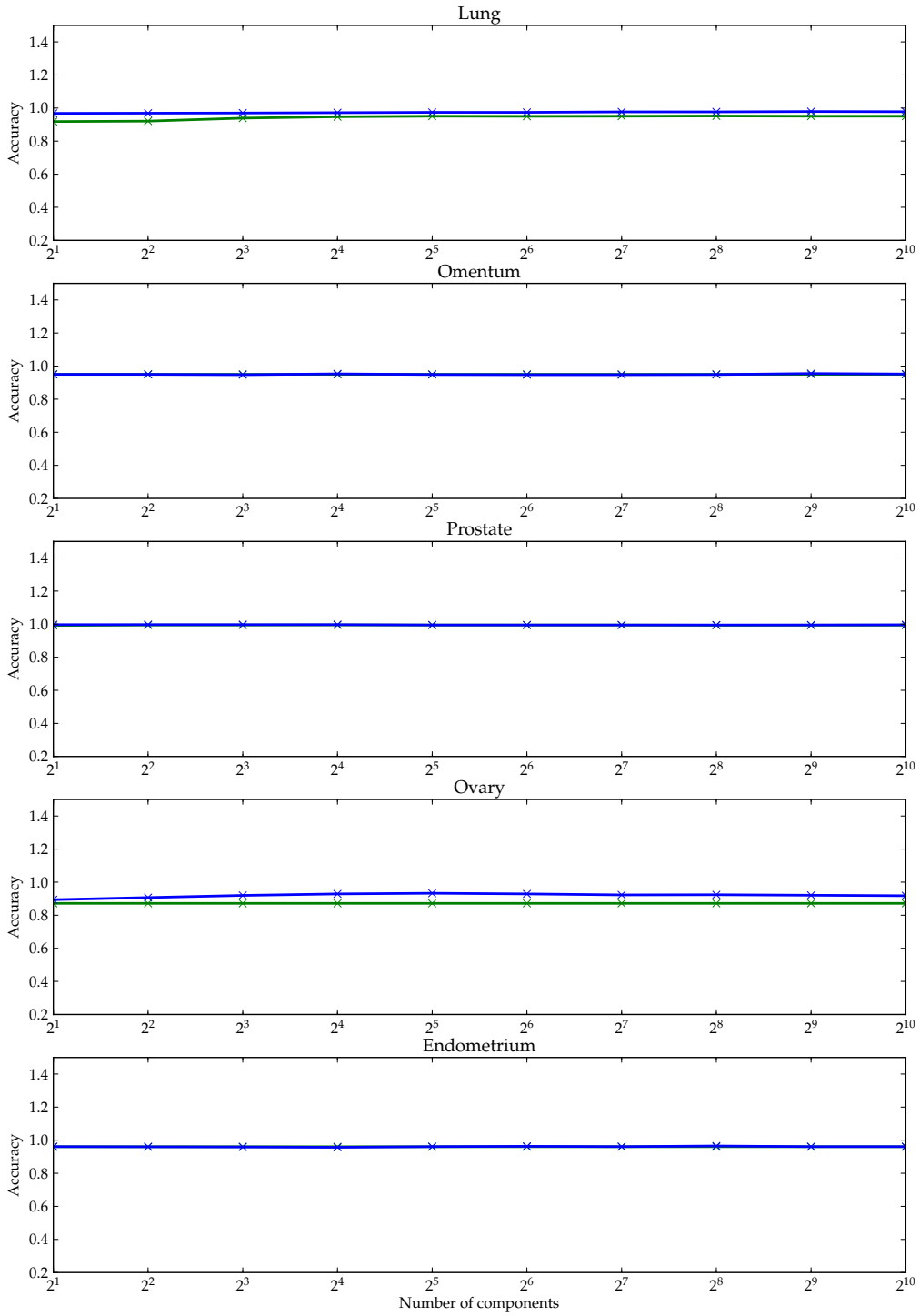


Figure 25: The accuracy on the GEMLeR dataset against the change in dimensionality of the SVM-RFE (blue) against ISOMAP (k = 5) (green) dimensionality reduction methods.

6

CONCLUSIONS & FUTURE WORK

6.1 CONCLUSIONS

In this research we have presented an empirical comparison of several manifold learning techniques on the analysis of microarray data. Classifying microarray data can be a very challenging task due to the "curse of dimensionality" as we identified in Section 2.4, but manifold learning techniques proved to aid in the process for a lot types of cancer. Although the ISOMAP algorithm has been previously been used by Dawson et al. (2005) on microarray data in the analysis of temporal, spatial and functional processes we have investigated and presented new ways in which it can be applied on cancer related data.

In Chapter 3 we managed to reduce the dimensionality of microarray datasets down to only a few dimensions, which allowed us to better grasp the available information visually. By visualising the microarray data we not only can easily detect the clusters of types of cancer tumours, but can also aid our approach to choose good clustering/classification algorithms. Additionally it allowed existing supervised techniques like k-NN and SVM to perform better on the manifold embedding for classification of tumour samples, than the original data sets.

To improve the results of the process, we also proposed a novel technique to incorporate *a priori* information into manifold learning, using the well-known ISOMAP algorithm (Chapter 4). Our results indicate that the semi-supervised manifold learning technique leads to improved biological significance, although the choice of the data to be used as prior knowledge is an open and difficult task across genetics researchers.

One of the shortcomings of this research is that gene regulation is a very condition specific task and thus the expression values of each individual gene must be an outcome of changes happening at that particular time when the microarray measurement was made. However our gene expression measurements did not happen at the same time, but rather could be spaced by a period of over a year and maybe more. Moreover, in Chapter 4 we made the assumption that genes that participate in the same biological pathways will possibly show co-expression in unrelated biological process as well.

6.2 FUTURE WORK

Of course the research described in this project is not in any way complete as it can be extended in numerous ways. Ideally we would like to evaluate our manifold learning algorithms on a vast amount of microarray data sets, to validate our algorithms performance by gaining more realistic figures. We highlight several a number of areas for further research such as:

6.2.1 *Complexity*

The system's performance is also is an extremely important factor that needs to need improving upon. The complexity of the proposed *a priori* algorithm as illustrated in Section 2.5.1, can approach $\mathcal{O}[N^3]$ where N is the number of probes we want to analyse; whereas for the standard application of ISOMAP, N would be the number of samples. Additionally we run into space-complexity problems as well as we need $\mathcal{O}[N^2 \cdot d]$ storage for the operation of the algorithm. For these reasons it will be beneficial to investigate iterative forms of ISOMAP such the ones presented in Law and Jain (2006), that require much less memory space but also allow the construction of concurrent implementations of the algorithm.

6.2.2 *Prior knowledge incorporation*

An important aspect will be to also research more about the information that can be used as prior knowledge into the manifold learning techniques, as our approach can be regarded preliminary. There is a lot of ongoing research going in the area of gene co-expression networks that would be interesting to apply in manifold learning techniques. Zhang and Horvath (2005) first demonstrated that gene co-expression can be expressed as a weighted connection network which can predict the biological significance of a gene, which we could use as an extension of our approach of incorporating prior knowledge to manifold learning.

6.2.3 *Regression analysis*

So far we have investigated the use of one-way classification methods to reveal information in the data. It would be interesting to apply regression techniques (on the lower-dimensional embedding of the manifold algorithms), such as survival models, to be able to answer questions like: *What is the expected lifetime of a patient? If one survives the treatment, at what rate will he/she die?* If one would measure the length of time between diagnosis and death or record the vital status of the patient when last observed for every patient in a group, one could potentially describe the survival of the group as the proportion of those who are alive at the end of the period under investigation. A popular method in analysis of gene expression data for predicting cancer recurrence or death at time t, is the Cox proportional hazards model (Cox, 1972).

BIBLIOGRAPHY

Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T., Yu, X., Powell, J., Yang, L., Marti, G., Moore, T., Hudson, J., Lu, L., Lewis, D., Tibshirani, R., Sherlock, G., Chan, W., Greiner, T., Weisenburger, D., Armitage, J., Warnke, R., Levy, R., Wilson, W., Grever, M., Byrd, J., Botstein, D., Brown, P., and Staudt, L. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511.

Baltimore, D. (1970). Rna-dependent dna polymerase in virions of rna tumour viruses. *Nature*, 226(5252):1209–1211.

Barmparas, G., Branco, B., Schnüriger, B., Lam, L., Inaba, K., and Demetriades, D. (2010). The incidence and risk factors of post-laparotomy adhesive small bowel obstruction. *Journal of Gastrointestinal Surgery*, 14:1619–1628.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396.

Bellman, R. E. (1957). *Dynamic programming*. Number ISBN 978-0-691-07951-6. Princeton University Press.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1999). When is "nearest neighbor" meaningful? In Beeri, C. and Buneman, P., editors, *Database Theory — ICDT'99*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer Berlin Heidelberg.

Birzele, F., Fauti, T., Stahl, H., Lenter, M., Simon, E., Knebel, D., Weith, A., Hildebrandt, T., and Mennerich, D. (2011). Next-generation insights into regulatory t cells: expression profiling and foxp3 occupancy in human. *Nucleic Acids Research*, 39(18):7946–7960.

Breslow, N. E. (1975). Analysis of survival data under the proportional hazards model. *International Statistical Review / Revue Internationale de Statistique*, 43(1):pp. 45–57.

Brunet, J.-P., Tamayo, P., Golub, T., and Mesirov, J. (2004). Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169.

Cayton, L. (2005). Algorithms for manifold learning. Technical Report CS2008-0923, UCSD.

Cheng, Y. and Church, G. (2000). Biclustering of expression data. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 8:93–103.

Cobaleda, C. and Sanchez-Garcia, I. (2009). B-cell acute lymphoblastic leukaemia: towards understanding its cellular origin. *BioEssays*, 31(6):600–609.

Cox, D. R. (1972). Regression models and life-tables. *J. Roy. Statist. Soc. Ser. B*, 34:187–220. With discussion by F. Downton, Richard Peto, D. J. Bartholomew, D. V. Lindley, P. W. Glassborow, D. E. Barton, Susannah Howard, B. Benjamin, John J. Gart, L. D. Meshalkin, A. R. Kagan, M. Zelen, R. E. Barlow, Jack Kalbfleisch, R. L. Prentice and Norman Breslow, and a reply by D. R. Cox.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2):224–227.

Dawson, K., Rodriguez, R. L., and Malyj, W. (2005). Sample phenotype clusters in high-density oligonucleotide microarray data sets are revealed using isomap, a nonlinear algorithm. *Bmc Bioinformatics*, 6(1):195.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets.

DeRisi, J., Iyer, V., and Brown, P. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.

Dong, J. and Horvath, S. (2007). Understanding network concepts in modules. *BMC systems biology*, 1(1).

Donoho, D. and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5591–5596.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868.

Elegans (1998). Genome sequence of the nematode c. elegans: a platform for investigating biology. *Science (New York, N.Y.)*, 282(5396):2012–2018.

Floyd, R. W. (1962). Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science (New York, N.Y.)*, 286(5439):531–537.

Gurney, H. (2002). How to calculate the dose of chemotherapy. *British Journal of Cancer*, 86(8):1297–1302.

Hartigan, J. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129.

He, X., Yan, S., Hu, Y., Niyogi, P., and Zhang, H. (2005). Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340.

Irizarry, R., Hobbs, B., Collin, F., Beazer-Barclay, Y., Antonellis, K., Scherf, U., and Speed, T. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics (Oxford, England)*, 4(2):249–264.

Jafari, A. and Almasganj, F. (2010). Using laplacian eigenmaps latent variable model and manifold learning to improve speech recognition accuracy. *Speech Communication*.

Jemal, A., Bray, F., Center, M. M., Ferlay, J., Ward, E., and Forman, D. (2011). Global cancer statistics. *CA: A Cancer Journal for Clinicians*, 61(2):69–90.

Jemal, A., Siegel, R., Ward, E., Hao, Y., Xu, J., and Thun, M. (2009). Cancer statistics, 2009. *CA: A Cancer Journal for Clinicians*, 59(4):225–249.

Klein, D., Kamvar, S. D., and Manning, C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 307–314, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Knuth, D. (1974). Computer programming as an art. *Commun. ACM*, 17(12):667–673.

Kohler, G. and Milstein, C. (1975). Continuous cultures of fused cells secreting antibody of predefined specificity. *Nature*, 256(5517):495–497.

Kumaravel Somasundaram, Sathish Kumar Mungamuri, N. W. (2002). Dna microarray technology and its applications in cancer biology. *Applied Genomics and Proteomics*.

Kung, S. and Mak., M. (2009). *Machine Learning in Bioinformatics*, volume Chapter 1: Feature Selection for Genomic and Proteomic Data Mining. New Jersey : John Wiley & Sons.

Law, M. and Jain, A. (2006). Incremental nonlinear dimensionality reduction by manifold learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(3):377–391.

Lenz, G., Wright, G., Emre, T., Kohlhammer, H., Dave, S., Davis, E., Carty, S., Lam, L., Shaffer, A., Xiao, W., Powell, J., Rosenwald, A., Ott, G., Muller-Hermelink, H., Gascoyne, R., Connors, J., Campo, E., Jaffe, E., Delabie, J., Smeland, E., Rimsza, L., Fisher, R., Weisenburger, D., Chan, W., and Staudt, L. (2008). Molecular subtypes of diffuse large b-cell lymphoma arise by distinct genetic pathways. *Proceedings of the National Academy of Sciences*, 105(36):13520–13525.

Mirkin, B. (1996). *Mathematical Classification and Clustering*. Kluwer Academic Publishers.

Morin, R., Bainbridge, M., Fejes, A., Hirst, M., Krzywinski, M., Pugh, T., McDonald, H., Varhol, R., Jones, S., and Marra, M. (2008). Profiling the hela s3 transcriptome using randomly primed cdna and massively parallel short-read sequencing. *BioTechniques*, 45(1):81–94.

NHGRI (2012). Biological pathways. http://www.genome.gov/.

Omohundro, S. M. (1989). Five Balltree Construction Algorithms.

Pal, T., Permuth-Wey, J., Betts, J., Krischer, J., Fiorica, J., Arango, H., LaPolla, J., Hoffman, M., Martino, M., Wakeley, K., Wilbanks, G., Nicosia, S., Cantor, A., and Sutphen, R. (2005). Brca1 and brca2 mutations account for a large proportion of ovarian carcinoma cases. *Cancer*, 104(12):2807–2816.

Parker, W., Broder, M., Chang, E., Feskanich, D., Farquhar, C., Liu, Z., Shoupe, D., Berek, J., Hankinson, S., and Manson, J. (2009). Ovarian conservation at the time of hysterectomy and long-term health outcomes in the nurses' health study. *Obstetrics and gynecology*, 113(5):1027–1037.

Perou, C. M., Sørlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A., et al. (2000). Molecular portraits of human breast tumours. *Nature*, 406(6797):747–752.

Rocca WA, Bower JH, M. D. A. J. G. B. d. A. M. M. L. r. (2007). Increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. increased risk of cognitive impairment or dementia in women who underwent oophorectomy before menopause. *Neurology*.

Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65.

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326.

Schena, M., Shalon, D., Davis, R., and Brown, P. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470.

Shendure, J. (2008). The beginning of the end for microarrays? *Nature Methods*, 5(7):585–587.

Stiglic, G. and Kokol, P. (2010). Stability of ranked gene lists in large microarray analysis studies. *Journal of Biomedicine and Biotechnology*, 2010.

Tang, C., Zhang, L., Zhang, A., and Ramanathan, M. (2001). Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In *Bioinformatics and Bioengineering Conference, 2001. Proceedings of the IEEE 2nd International Symposium on*, pages 41 –48.

TCGA (2011). Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615.

Tenenbaum, J., de Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Tothill, R., Tinker, A., George, J., Brown, R., Fox, S., Lade, S., Johnson, D., Trivett, M., Etemadmoghadam, D., Locandro, B., Traficante, N., Fereday, S., Hung, J., Chiew, Y.-E., Haviv, I., Group, A. O. C. S., Gertig, D., deFazio, A., and Bowtell, D. (2008). Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clinical Cancer Research*, 14(16):5198–5208.

van der Laan, M. and Pollard, K. (2003). A new algorithm for hybrid hierarchical clustering with visualization and the bootstrap. *Journal of Statistical Planning and Inference*, 117(2):275–303.

Vinh, N., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*.

Weinstein, J., Myers, T., O'Connor, P., Friend, S., Fornace, A., Kohn, K., Fojo, T., Bates, S., Rubinstein, L., Anderson, N., Buolamwini, J., van Osdol, W., Monks, A., Scudiero, D., Sausville, E., Zaharevitz, D., Bunow, B., Viswanadhan, V., Johnson, G., Wittes, R., and Paull, K. (1997). An information-intensive approach to the molecular pharmacology of cancer. *Science*, 275(5298):343–349.

Whipkey, K. L. (1984). Identifying predictors of programming skill. *SIGCSE Bull.*, 16(4):36–42.

Yang, Y., Dudoit, S., Luu, P., Lin, D., Peng, V., Ngai, J., and Speed, T. (2002). Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4):e15–e15.

Zhang, B. and Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1).

Zhang, Z. and Wang, J. (2006). Mlle: Modified locally linear embedding using multiple weights. *IEEE*.

APPENDIX

A.1 MICROARRAY DATA REPOSITORIES

Table 5: Repositories containing microarray data

#	Database	Description
1	Gene Expression Omnibus ncbi.nlm.nih.gov/geo/	Public data deposition and public queries
2	MSigDB/GSEA (Broad Institute) broadinstitute.org/gsea/msigdb/	Local installation and public queries
3	Oncomine (University Michigan) oncomine.org/	Queries and data installation
4	National Cancer Institute (NCI) madb.nci.nih.gov/	Local installation
5	Array Express ebi.ac.uk/arrayexpress/	Local installation
6	Kent Ridge Bio-medical Data set Repository datam.i2r.a-star.edu.sg/	Local installation
7	Gene Expression Machine Learning Repository gemler.fzv.uni-mb.si/	Local installation

A.2 IMPLEMENTATION DETAILS

This section will give a brief overview of the tools used and the reasons of which were chosen to fulfil the needs of the microarray analysis research.

Python was the main programming tool of choice as, along with the provide libraries, provide an excellent scientific platform. Most importantly the libraries we make use of are all open-source and thus we can easily adapt the algorithms we are going to use to our needs. It is not unusual for us to want to trade-off running-time for space complexity or vice-versa depending on the application.

Main tools used throughout this project:

- http://scikit-learn.org/
  A general machine learning library which contains a variety of state-of-the-art learning algorithms spanning supervised learning, unsupervised learning, model selection and samples datasets.

- http://matplotlib.org/
  matplotlib is a python plotting library which produces publication quality figures in a variety of formats.

- http://www.numpy.org/
  numpy allows for efficient and powerful N-dimensional array manipulation, essential for working with vast amount of data and vectors like microarrays.

- http://www.r-project.org/
  R-lang is a powerful statistical programming language that that we used to statistically evaluate our algorithms. We also made use of RPy2 which provides a robust interface between Python and the R programming language.

- http://www.bioconductor.org/
  In addition Bioconductor uses R-lang which we use to provide us annotations for various manufacturer microarray products, in order to be able to link this information with KEGG.

- http://vis.usal.es/bicoverlapper/
  BicOverlapper is a visual framework that allows for the simultaneous visualisation of one or more sets of biclusters, heatmaps of gene expression matrices and gene annotations.

ACRONYMS

adjuvant	An adjuvant is a pharmacological or immunological agent that modifies the effect of other agents. 9
ALL	Acute lymphoblastic leukaemia. 37, 39, 49, 51
ARI	Adjusted Rand Index. 39, 41, 51
DBI	Davies-Bouldin Index. 40
DNA	Deoxyribonucleic acid (DNA) is the material that encodes the genetic instructions in all known living organisms. 11
GEMLeR	Gene Expression Machine Learning Repository. 51
KEGG	Kyoto Encyclopedia of Genes and Genomes. 45, 58
LOO	Leave-One-Out. 41
NMF	Non-negative matrix factorisation. 37
PCC	Pearson correlation coefficient. 23
RMA	Robust Multi-array Average. 15
RNA	Ribonucleic acid (RNA) is preliminarily responsible for coding, decoding, regulation and expression of genes. 11