Targeted Projection Pursuit Tool for Gene Expression Visualisation

Joe Faith, Michael Brockway

Northumbria University, Newcastle, UK

Summary

A tool is introduced that uses a novel technique to enable users to explore two-dimensional views of high dimensional gene expression data sets. Unlike other such tools, the interface is intuitive and efficient, allowing the user to easily select views that meet their requirements. The tool is tested on publicly available gene expression data sets and demonstrated to find views that show the seperation of gene expression data sets into classes more effectively than standard dimension-reduction methods.

1 Introduction

This paper considers the problem of exploring and visulising high dimensional gene expression data. There are many powerful automatic techniques for analysing such data, but visualisation represents an essential part of the analysis as it facilitates the discovery of structures, features, patterns and relationships, enables human exploration and communication of the data and enhances the generation of hypotheses, diagnoses, and decision making.

Visualising gene expression data requires representing the data in two (or occasionally one or three) dimensions. Therefore techniques are required to accurately and informatively show these very high-dimensional data structures in low dimensional representations.

There are many established techniques for reducing the dimensionality of data for visualisation purposes. Among these, multi-dimensional scaling (MDS), including Sammon mapping, finds a view of the data that best preserves the distances between points [6]; VizStruct is a technique based on radial coordinates [18]; dendrograms may be used to linearly arrange and display clustered gene expression data [5]; and projection pursuit [13] finds linear projections that optimise some measure of their quality.

Each of these techniques has limitations and advantages. MDS is able to scale to very high dimensional data spaces – though Figure 5 below illustrates one limitation on this – but is a map-based, rather than projection-based, technique in which adding single datum requires creating a new view of the entire set; thus it is not possible to visualise the relationships of new or unclassified samples to existing ones. VizStruct is not optimized for viewing classifications of the data, and is also only able to accurately visualize data across relatively small numbers of genes (e.g. 12) – hence is reliant on reducing the dimensionality of the original data through some form of feature selection. Dendrograms use linear arrangements of the data and so are restricted to a single dimension for display.

A fundamental advantage of using linear projections for visualisation compared to, for example, MDS, is that they define a transform that can be applied to any point in gene-space. In particular,

the projection contains information about the respective significance of each gene, and how they can be best combined to perform functions such as classification and genetic feature selection, or to identify gene expression signatures [14].

Nonetheless, each of these techniques produces a single static view of the data, possibly optimised for a particular application. Here we present a tool that allows the user to truly 'explore' high dimensional data sets using an interface that is intuitive, efficient, and powerful. The interface presents the user with a two dimensional view of the data which they can then manipulate using mouse actions. The tool then dynamically finds linear projections of the data that best match the user's requirements.

The tool has been implemented in prototype and compared quantitatively and qualitatively with other dimension reduction techniques at the task of visually seperating classified gene expression data sets. It is found to outperform all alternatives in this task.

The prototype tool, along with example data sets, is publicly available from the associated web-site[1].

2 Targeted Projection Pursuit Tool

Friedman and Tukey introduced the term *projection pursuit* to describe the process of finding interesting linear projections by optimizing some function (the *projection pursuit index*)[8]. The definition of what makes a projection 'interesting' depends on the projection pursuit index and on the application or purpose. For example, Lee *et al* [13] discuss a projection pursuit index that measures how well each projection shows the seperation of classes in the data.

The problem here is that all such techniques rely on a pre-determined notion of 'interest'. Different projections may be available but, at best, they will be presented to the user as a fixed menu of possibilities. One alternative to the static views produced by projection pursuit is Asimovs Grand Tour [1] – described as an attempt to look at the data 'from all possible angles. A Grand Tour is a video sequence in which each frame shows the result of a single projection of the data, with the sequence as a whole including all possible projection planes. However, the Grand Tour replaces the quality of projection pursuit with quantity: a grand tour in high dimensional space may be long and mostly uninformative.

Ideally we would have some way of allowing user to guide the tour, to use their perception of the data to find projections of interest. Cook and Buja ([2][3]) proposed and implemented an interface that allows the user to not only pause and rewind a given Grand Tour, but also to amend the resulting view by controlling the input from each dimension independently. The problem is that projection component manipulation is an *opaque* interface in the sense that it is rarely possible for the user to anticipate the effect of their actions. Where the user has strong intuitions about the nature of the structure of interest in data, and its relationship with the underlying coordinate system, then it may be possible for them to determine how best to use component-based controls to reveal structure in the data more clearly. In other words, once the user knows what they are looking for, then such an interface will help them find it. But it is unsuited to true exploration of the data. The user has n controls to manipulate (one for each dimension of the original data set), the effect of each will be unknown and which will have

[1] http://computing.unn.ac.uk/staff/CGJF1/tpp/tool.html

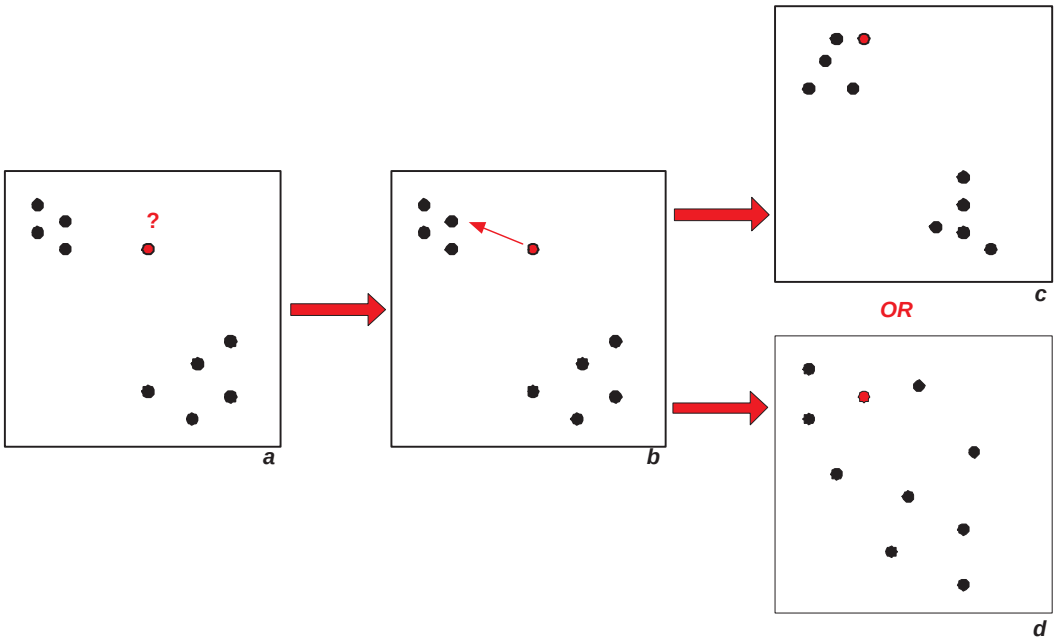

Figure 1: The use of targeted projection pursuit for interactive data exploration. *a* An initial view of the data with two partial clusters and an outlier. *b* The user hypothesises that the outlier is part of the upper cluster and drags it into place. *c* If the data supports such a clustering then the tool finds a view of the data that matches the hypothesis. *d* The data does not support the hypothesis and moving the point disrupts the partial clusters.

unpredictable effects in combination. The user can do little more than random search – which has its place, but is of little use when faced with a truly large dimensionality set.

The basis of our alternative projection pursuit tool is that the user should manipulate their view of the data directly, rather than manipulate the projection that produces that view. This uses a novel method we call *Targeted Projection Pursuit* in which a linear projection is found that produces a view of the data that best matches an ideal target view [7]. The proposed tool works as follows (see Figure 1). When the tool is started, the user sees an initial two-dimensional view of the data. Suppose the user can discern some kind of pattern, for example if there appears to be some clustering in the data, albeit with many outliers (Figure 1a). If this is the case the user would hypothesise that the clustering is due to a genuine regularity in the data and that the outliers are simply a product of the particular projection – for example, due to the inclusion of a component of the data comprised mostly of noise. In this case the user would select an outlier and attempt to drag it into the nearest cluster (Figure 1b). The tool then attempts to find a projection of the data that best matches this revised view, and redisplays the data. If such a view can be found then it will be displayed and the clusters will 'fall into place' (Figure 1c), for example by removing the contribution of the noisy component. Otherwise the partial clusters will be revealed to have been solely been an artefact of the initial projection (Figure 1d).

As well as manipulating the projections to find clusters, the user can also try dragging and dropping points into curvilinear relationships, or linearly separable regions. And, rather than single points, the user can select a region including a cluster of points, to fix or move. Alternatively, if the data is already classified into known classes then the tool may be used to find low-dimensional views in which those classes are most clearly shown, and to identify outliers. (It is this latter application that is used to test the tool below.)

3 Tool Implementation

The tool described in Section 2 was implemented using Java, incorporating data-handling functionality from Weka [16] and neural net functionality from Gurel [11]. Linear projections that best match the target views are found using the method discussed in [7]. Although it is currently at the prototype stage, the tool can be used as follows.

First, the user loads a classified data sets from a standard .arff format file [16], and an initial view of the data is presented using the first two principal components ([17], however the particular choice of initial view is unimportant). Each class in the data set is represented by points of a particular colour. Subsets of data points can be selected using a rectangular 'rubber band' and then dragged in the X and Y directions; the closest possible projection is then found dynamically and the resulting data positions redisplayed. Mouse drags can also be used to expand or contract the area of the selected points.

An example interaction is shown in Figure 3. When the tool is started, the data is displayed using the first two principal components. In this particular case the majority of points are bunched into one corner. The user would like a projection that more clearly shows the 'bunched' points, so selects a region enclosing those points and drags the corner of the bounding box to expand it. The tool then finds a view of the data that produces a better 'spread' of the selected data. The user would then like to see whether a subset of points can separated to form a distinct cluster, so selects those points and drags the region away from other points. The tool again manages to find a view that better approximates the users requirements.

On a standard desktop PC[2], the tool is able to calculate projections of 100 data points of 1000 dimensions and display them at approximately 5 frames per second.

4 Method

The tool is designed to be a general purpose method for exploring and visualising high dimensional data sets, but in order to compare it objectively and quantitatively with standard dimension-reduction techniques it was tested for its effectiveness in one particular task: finding views of classified gene expression data sets,

In this task the user was presented with views of classified gene expression data sets, in which each class of sample is represented by points of differing colours. They were then invited to use the tool to find views that best show the separation between classes. The resulting two-dimensional view of the data was then tested using a standard statistical measure of class separation (I_{LDA}, a version of Wilks' Lamda [13]); and by seeing how effectively a classification algorithm can use that view alone – rather than the original high-dimensional data – to classify the samples (K Nearest Neighbours with $k = 5$).

The following dimension reduction techniques were compared

- **TPP**: The view generated by the prototype Targeted Projection Pursuit described above, using non-orthogonal projections.

[2] 1.8GHz Pentium 4 CPU, 800MB RAM, Windows XP OS.

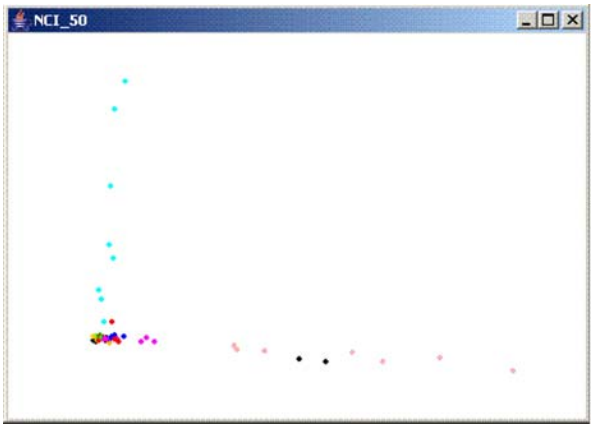

Figure 3.1: Prototype Tool Interface: the initial view of the chosen data set showing the first two principal components.

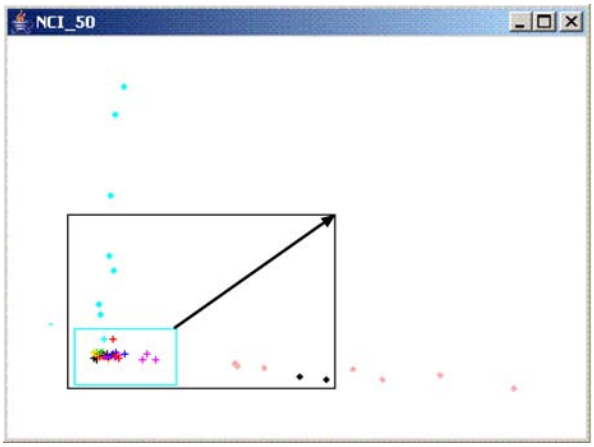

Figure 3.2: Prototype Tool Interface: the user expands a region to distinguish bunched points.

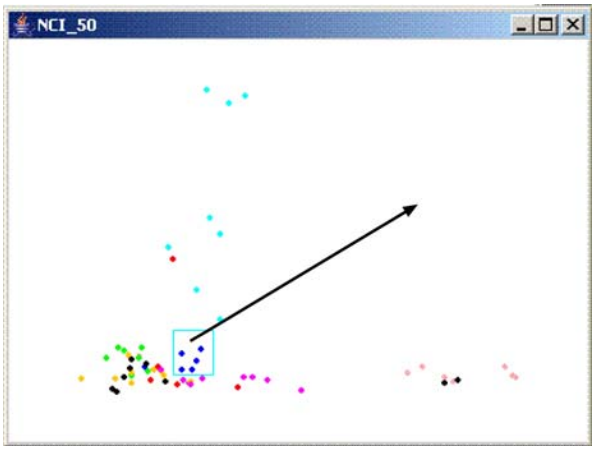

Figure 3.3: Prototype Tool Interface: the user selects points that may form a cluster.

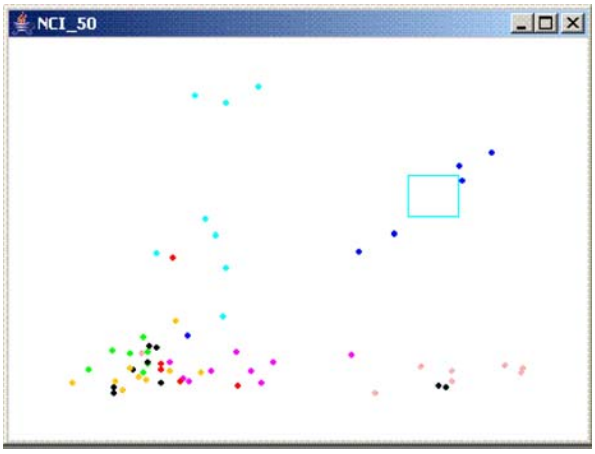

Figure 3.4: Prototype Tool Interface: the user successfully separates the chosen cluster of points.

- **PP**: The linear projection produced by search-based projection pursuit [13]. This algorithm uses simulated annealing to search the space of all possible linear projections, maximising a 'projection pursuit index' that measures how effectively each projection seperates each class of sample.
- **SAM**: The map of the data produced by a Sammon Mapping – a multidimensional scaling techniques that finds a 2D arrangement of points that best preserves the distances between samples in the original data [6].
- **VS**: The result of a VizStruct non-linear projection based on radial coordinates [18].

The following data sets were used:

- **LEUK**: This dataset is the result of a study of gene expression in two types of acute leukemia: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) [10]. The samples consist of 38 cases of B-cell ALL, 9 cases of T-cell ALL, and 25 cases of AML with the expression levels of 7219 genes measured. Note that, following [13], the B-cell and T-cell ALL samples are considered as separate classes.
- **SRBCT**: This dataset comprises cDNA microarray analysis of small, round blue cell childhood tumors (SRBCT), including neuroblastoma (NB), rhabdomyosarcoma (RMS), Burkitt Lymphoma (BL; a subset of non-Hodgkin lymphoma) and members of Ewing family of tumors (EWS). Expression levels from 6567 genes for 83 samples were taken [12].
- **NCI**: This dataset records the variation in gene expression among the 60 cell lines from the National Cancer Institute's anticancer drug screen [15]. It consists of 8 different tissue types where cancer was found: 9 breast, 5 central nervous system (CNS), 7 colon, 6 leukemia, 8 melanoma, 9 non-small-cell lung carcinoma (NSCLC), 6 ovarian, 2 prostate, 8 renal. 9703 cDNA sequences were used.

No additional normalisation was applied to the data. Initial feature selection reduced the dimensionality of the data set by finding the 50 genes with the highest Between-Group to Within-Group Sum of Squares [4].

5 Results

The quantitative comparison of the four projections on the three data sets is shown in Table 1, and a sample of the resulting views are given in Figure 5. (A complete set of views are available on the associated web-site.)

The first aspect of the results to note is that the choice of dimension-reduction technique can alter radically the resulting view of the data, judged both quantitatively and qualitatively. The structure and relationship between clusters appears very differently in each view, resulting in very different performances of classification algorithms. The choice of dimension reduction technique clearly matters in visualising high dimensional data such as gene expression data.

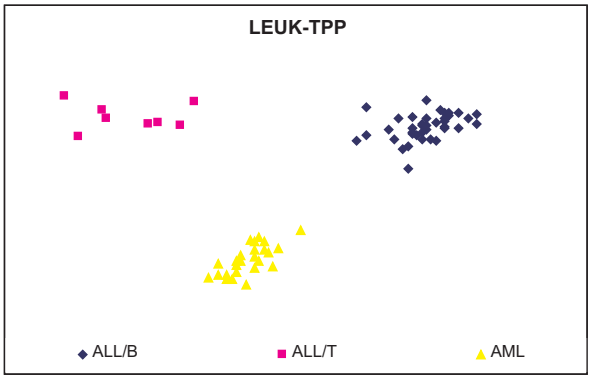

Figure 5.1: Two-Dimensional View of LEUK data set produced by TPP method showing a clear separation between all classes.

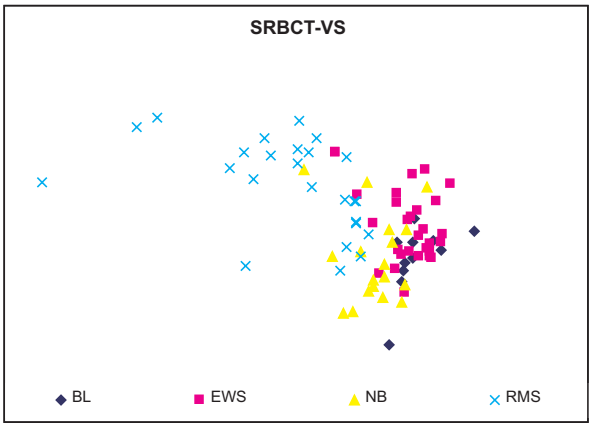

Figure 5.4: Two-Dimensional View of SRBCT data set produced by VS method with very little class separation.

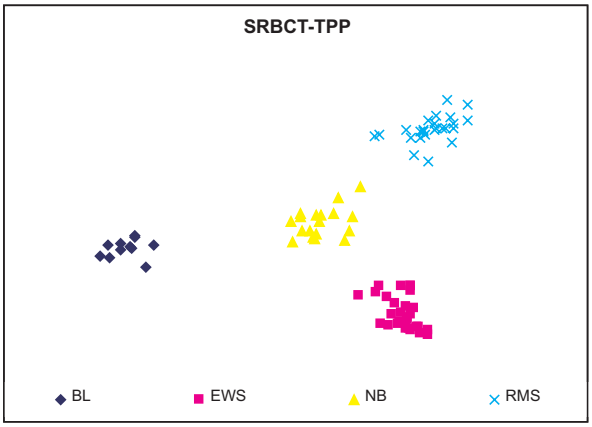

Figure 5.2: Two-Dimensional View of SRBCT data set produced by TPP method, showing clear separation between all clasess.

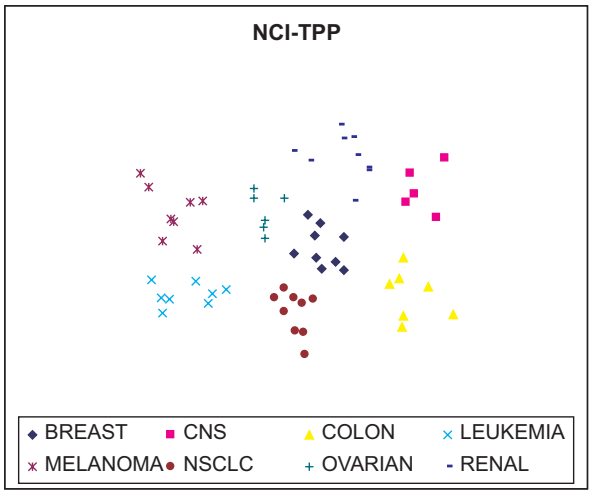

Figure 5.5: Two-Dimensional View of NCI data set produced by TPP method: most classes are clearly seperated, albeit with outliers.

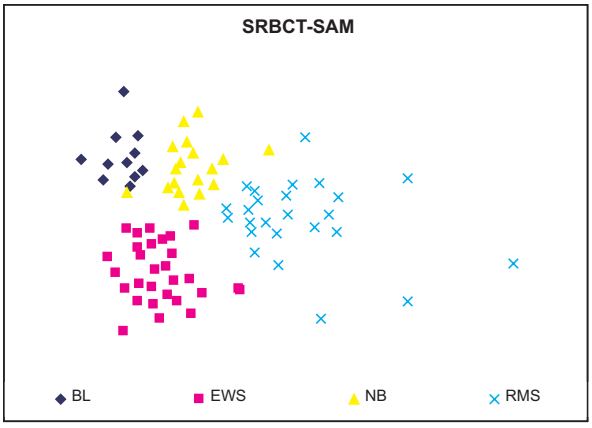

Figure 5.3: Two-Dimensional View of SRBCT data set produced by SAM method showing one aspect of the 'curse of dimensionality': the small variance between points in high dimensional space produces a reduced view with little 'bunching'.

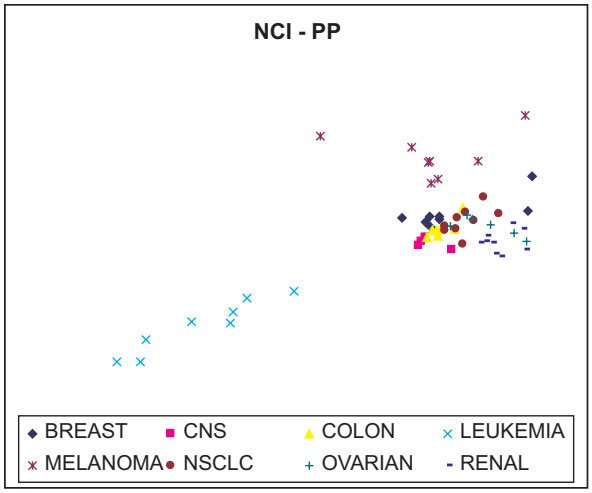

Figure 5.6: Two-Dimensional View of NCI data set produced by PP method: only leukemia and melanoma cases are separated.

Table 1: Comparison of class separability following dimension reduction for visualisation. Each technique (TPP, PP, SAM and VS) is evaluated on each data set (LEUK, SRBCT,NCI), and the separability of the resulting view tested using 5-Nearest Neighbours classification (generalisation error in %) and a version of Wilks Lamda ($0 < I_{LDA} < 1$).

Data Set		**LEUK**		**SRBCT**		**NCI**	
Class Separation Measure		I_{LDA}	$5NN$	I_{LDA}	$5NN$	I_{LDA}	$5NN$
	TPP	.997	100.0	.999	100.0	.994	96.7
Dimension Reduction Technique:	PP	.972	98.6	.988	100.0	.981	62.3
	SAM	.959	97.2	.911	95.2	.927	67.2
	VS	.952	95.8	.637	56.6	.838	32.8

The second aspect to note is that finding views that show distinct classes is harder the more classes there are in the data: the classification performance resulting from each technique drops almost monotonically as the number of classes increases.

Overall, VizStruct performed least well in separating classes. Although the difference between VizStruct and the other techniques was least for the case with fewer classes (LEUK), the difference became more marked as the number of classes increased. This poor performance is unsurprising, since this technique is not explicitly designed to accentuate classifications (though see [18]).

The Sammon mapping performed well in separating classes, but its output was marked by the 'curse of dimensionality': *i.e.* in high dimensional spaces, the variance in distances between randomly distributed points decreases. Sammon mapping attempts to preserve the distance between data points, and hence the resulting views tend to be evenly distributed, with little bunching of points belonging to a single class. Classification algorithms may succeed in ascribing points to classes – and hence the classification scores for SAM are similar to those for the linear mappings produced by TPP and PP – but this may not be an accurate reflection of the perceived class seperation.

Both projection pursuit techniques performed well, which is unsurprising since they are specifically designed for this task; however in each case targeted projection pursuit using the prototype tool outperformed all other techniques, including search-based projection pursuit, in visually seperating sample classes.

Conventional search-based projection pursuit also suffered from unreliability. Since it is partly a stochastic technique, the results could differ. Over a sequence of 100 trials, the values for I_{LDA} for PP applied to the NCI set ranged from 0.935 to 0.992 (mean=0.978, standard deviation=0.00924). The values for I_{LDA} and $5NN$ shown in Table 1, and the view show in Figure 5 are for a projection of near-mean I_{LDA} value.

6 Discussion

The wealth of data produced by high-volume experimental techniques such as DNA microarray analysis brings its own problems. Many computational techniques have been developed, and many existing techniques have been applied, to their analysis; but for the experimenter or analyst there is still no substitute for getting a 'feel' of a data set. In this paper we have introduced

a tool that allows a user to do just that, and have tested its efficacy on a particular task. Initial results are extremely encouraging

Targeted projection pursuit using the prototype tool was able to produce views of gene expression data that separated sample classes better than all other techniques. This is unexpected, since it may be supposed that a computational technique, such as search-based projection pursuit, would outperform a human user in searching a large space of possible projections. However, the tool is so designed to combine the strengths of human pattern recognition with a computer's ability to optimise sets of linear equations.

(It may be supposed that for simpler problems – *ie* where the number of gene/dimensions is lower – then search-based techniques may prove more capable. See [7].)

The tool is currently in prototype, but the authors are using the experience of these initial experiments to produce a more fully featured version; to explore how it can be used in the experimental process; and to explore applications in feature selection and unsupervised classification. Early examples of such applications can be seen on the associated web-site[3].

References

[1] Asimov, D. (1985). The Grand Tour: A Tool for Viewing Multidimensional Data. *SIAM Journal of Scientific and Statistical Computing* 6(1), pp128 – 11.

[2] Cook, D., Buja, A., Cabrera, J., and Hurley, H. (1995), Grand tour and projection pursuit, *Journal of Computational and Graphical Statistics*, 2(3), pp.225–250.

[3] Cook,D. and Buja,A. (1997), Manual Controls for High-Dimensional Data Projections, *Journal of Computational and Graphical Statistics*, 6(4), pp. 464-480.

[4] Dudoit, S., Fridlyand, J., and Speed, T. P. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, Vol. 97, No. 457, p. 77-87.

[5] Eisen,M.B., Spellman,P.T., Brown,P.O., and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns, *PNAS* 95:25, 14863-14868

[6] Ewing,R.M. and Cherry,J.M. (2001) Visualisation of expression clusters using Sammon's non-linear mapping. *Bioinformatics*, 17,658-659.

[7] Faith,J and Mintram,R. and Angelova,M. (2006) Targeted Projection Pursuit for Gene Expression Data Classification and Visualisation, *Bioinformatics*, (submitted)

[8] Friedman, J. H., and Tukey, J. W., (1974),A Projection Pursuit Algorithm for Exploratory Data Analysis,*IEEE Transactions on Computers*, C-23, 881-890.

[9] Golub,J. and Loan,C.F. (1996) *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press.

[3] http://computing.unn.ac.uk/staff/CGJF1/tpp/tool.html

[10] Golub,T.R., Slonim,D.K., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J.P., Coller,H., Loh,M.L., Downing,.J.R., Caligiuri,M.A., Bloomfield,C.D., Lander,E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*,286(5439):531-7.

[11] Gurel,A. (2003) *Feed Forward Neural Networks - A Java Implementation V2.0*, `aydingurel.brinkster.net/neural`

[12] Khan,J., Wei,J.S., Ringnr,M., Saal,L.H., Ladanyi,M., Westermann,F., Berthold,F., Schwab,M., Antonescu,C.R., Peterson,C., and Meltzer,P.S. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6): 673–679.

[13] Lee,E.K, Cook,D., Klinke,S. and Lumley,T. (2005), Projection Pursuit for Exploratory Supervised Classification, *Journal of Computational and Graphical Statistics*, 14(4), 831-846

[14] Misra,J.,Schmitt,W., Hwang,D., Hsiao,LL., Gullans,S., Stephanopoulos,Ge. and Stephanopoulos,Gr. (2002), Interactive Exploration of Microarray Gene Expression Patterns in a Reduced Dimensional Space, *Genome Research*, 12(7), 1112–1120

[15] Scherf,U., Ross,D.T., Waltham,M., Smith,L.H., Lee,J.K., Tanabe,L., Kohn,K.W., Reinhold,W.C., Myers,T.G., Andrews,D.T., Scudiero,D.A., Eisen,M.B., Sausville,E.A., Pommier,Y., Botstein,D., Brown,P.O., and Weinstein,J.N. (2000) A Gene Expression Database for the Molecular Pharmacology of Cancer, *Nature Genetics*, 24(3), 236-244.

[16] Witten,I.H. and Frank,E. (2005) *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[17] Yeung,K.Y. and Ruzzo,W.L. (2001) Principal Components Analysis for clustering gene expression data, *Bioinformatics* 17 (9) 763-774

[18] Zhang,L., Zhang, A. and Ramanathan,M. (2004) VizStruct: exploratory visualisation for gene expression profiling, *Bioinformatics*, 20, 85-92.