VISUALISING UNCERTAIN DATA

D. D'Alimonte, D. Lowe, I. T. Nabney, M. Sivaraksa

Aston University, UK

ABSTRACT

Recent emerging aspects of clinical data analysis, especially of high dimensional data such as genomic studies of microarrays, have begun to exploit interactive visualisation tools for exploratory data mining. However, existing tools have not been designed to accommodate *uncertainty* in the data measurements, even when estimates are available. In this work we show how it is possible to incorporate knowledge of uncertainty in data measurements in influencing the data representation in two modern visualisation tools: NeuroScale and the Generative Topographic Mapping (GTM). The approaches are illustrated on selected synthetic data problems and a microarray dataset of importance to pharmacogenetics research.

INTRODUCTION

Techniques for data visualization and exploration are becoming even more significant as our ability to produce high dimensional biomedical data outstrips our ability to analyse the results. As cited in a recent review on *"Statistical Challenges in Functional Genomics"*, Sebastiani et. al. (1), *"The newly born functional genomic community is in great need of tools for data analysis and visual display of the results"*. Such tools in the clinical/biomedical domains have tended to rely upon clustering and the dendrogram, Khan et. al. (2), or projective methods such as Principal Component Analysis, Raychaudhuri (3), and Sammon map, Apostol and Szpankowski (4), and Independent Component Analysis, Draghici (5). More recent interesting methods for data mining and knowledge discovery in databases have developed using minimal spanning trees for complex data visualisation, Laskaris(6), capable of patient-specific and test-specific analysis.

Tools for data exploration, especially through low dimensional data visualisation are especially important for users such as research scientists or clinicians who are not specialists in data modelling. Visualisation is an effective way for domain experts to detect trends, structures, clusters, outliers, and other important data characteristics. In addition, it can be used to guide the data analysis process by giving feedback on the results of models.

The use of automated tools for assisting the clinician or user to help data-mine complex high-dimensional measurements provided by biomedical systems such as microarrays, MEG and EEG, and lab-on-a-chip medical testing is a major emerging aspect of clinical data analysis. The difficulty stems from the very high dimensional nature of data measurements, coupled with the nonlinearity, lack of strong prior models, nonstationary nature of dynamics in biology, and the often very noisy nature of the obtained data.

Recent innovations to produce algorithms for the low-dimensional representation of high-dimensional data have tried to focus on retaining *structural integrity* of the original data. This implies that the visualisation space should be *topographic* in some sense. Amongst very recent developments to try and achieve such visualisations are methods based around extending the Sammon Map to be generalisable (ie NeuroScale, Lowe and Tipping (7), and later similar approaches but dividing the space into local regions (Local Linear Embeddings, Saul and Roweis (8)), and more generative density modelling approaches such as the Generative Topographic Map (GTM), Bishop et.al. (9), and Stochastic Neighbor Embeddings, Hinton and Roweis (10). These techniques have been extended to provide more information about the embedding of data in its high-dimensional space with magnification factors and curvatures, hierarchical models that support drilling down into data, Tiňo and Nabney (11), and visualisation of time series dynamics, D'Alimonte et.al. (12).

Despite these developments, there are still practical issues that arise when using visualisation tools on medical data. One of the most important of these is the *uncertainty* of measurements as a consequence of the noise processes involved in the experimental design and techniques for gathering data. For example, high-throughput measurement techniques (such as those used for genomics and proteomics) have many potential sources of error. Where these can be quantified, a *certainty* value can be attached to each measurement. In many physiological measurements (such as EEG and ECG) the signal-to-noise ratio varies considerably (particularly in ambulatory recordings) and careful analysis can estimate the quality of the signal and how it changes over time. All analysis should take account of the data certainty so that less certain information has less influence on the results.

Current visualisation methods have not been designed to take into account the influence of measurement uncertainty on the output visualisations. The primary purpose of this paper is to show how some modern visualisation techniques can be modified in a relatively simple way to account for data certainty, both during the training (parameter estimation) phase, and when visualising data. In Section 2, we introduce the models we will be considering and the modifications needed to the training algorithms. Section 3 shows how well the techniques work on synthetic datasets, while Section 4 describes a case study on genomic data of pharmacogenetic relevance where quantitative measures of uncertainty are provided by the measuring process. The final section draws together the main conclusions from our work.

VISUALISATION MODELS AND UNCERTAINTY

We choose to show how a projective and a generative model for data visualisation can be adapted such that their visualisations are influenced by uncertainty in data. We choose *NeuroScale* as the prototype projective model and GTM as the generative model to be modified according to a prescribed 'certainty' C_n of data point $\vec{x}_n$. Hereafter, we assume that the certainty is a non-negative value falling in the range [0, 1]. Often, it can be interpreted in a Bayesian framework as our 'degree of belief' in the quality of the measurement. However, it is not possible to give a uniform definition valid across all applications (for example, when the source of the certainty measure is outside the control of the data analyst: see Section 5).

Neuroscale

In a Neuroscale topographic map the distribution and relative positions of the points in the data space are determined to reflect the relative dissimilarity between data measurements in the high-dimensional space, and hence generalises the established Sammon map. N measurement vectors $\vec{x}_i$ in $\mathbb{R}^p$ are transformed using a Radial Basis Function (RBF) network to a corresponding set of feature (visualisation) vectors $\vec{y}_i$ in $\mathbb{R}^q$. Generally, $q < p$ as dimension reduction is desired, and typically $q = 2$ for visualisation. The quality of the projection is measured by the *Sammon STRESS metric* (n.b. we are using a reduced form here, neglecting a denominator often employed):

$$E = \sum_i^N \sum_j^N (d^*_{ij} - d_{ij})^2, \tag{1}$$

where

$$\begin{aligned} d_{ij} &= \|\vec{y}_i - \vec{y}_j\|, \\ d^*_{ij} &= \|\vec{x}_i - \vec{x}_j\|, \end{aligned} \tag{2}$$

represent the inter-point distances in projection space and data space respectively. The aim of training process is to set the parameters of the RBF to minimise the STRESS metric.

Rather than use a standard non-linear optimisation algorithm to train the weights of the RBF network, Tipping and Lowe (13) showed that there is a more efficient approach, which they called *shadow targets*. This algorithm makes use of the general linear structure of the RBF model, so that hidden unit parameters are set using the input data only, and the output layer weights are found with a model trust region approach, Nabney (14).

One approach to account for confidence values on measurements is to modify equation (1) by adding a weighting term to it.

$$E = \sum_i^N \sum_j^N K_{ij} (d^*_{ij} - d_{ij})^2, \tag{3}$$

where $K_{ij} = \min(C_i, C_j)$; C_i is a confidence value on data point i.

More generally, K_{ij} is a function of points $\vec{x}_i$ and $\vec{x}_j$ that reflects the mutual uncertainty between the data points. The role of K_{ij} is to modify the influence that these points contribute to the STRESS metric depending on our relative confidence in these data points. Equation (3) can be interpreted as saying that relative dissimilarities obtained from data with low inter-point confidence are less important in determining the mapping parameters. This will make the visualisation plot less affected by outlying low-confidence data.

Since K_{ij} are pre-specified, the training algorithm for *NeuroScale* can be used with just the modified STRESS function.

Note that this is a heuristic approach, since a fully probabilistic formulation of *NeuroScale* does not yet exist. A partial address to this problem can be considered where the relative dissimilarities in the STRESS measure are derived from probabilistic distances between distributions generating the measurements. However this will not be considered in this paper.

GTM

The Generative Topographic Mapping (GTM), (9), was introduced as a probabilistic alternative to the well-known Self-Organizing Map (SOM) of Kohonen, Kohonen (15), and overcomes most the significant limitations of the SOM. They include the absence of a cost function, the lack of a theoretical basis for choosing learning rate parameter schedules and neighbourhood parameters, no general proof of convergence, and the lack of a density model, (9). In the GTM, the data are modelled by a constrained mixture of Gaussians whose parameters can be optimised by using the EM (expectation-maximization) algorithm.

The data $\vec{x} = (x_1, \ldots, x_d)$ lie in a d-dimensional space and is modelled using a q-dimensional latent variable space $\vec{z} = (z_1, \ldots, z_q)$. The two spaces are linked by a function $\vec{y}(\vec{z}; \vec{W})$ which maps $\vec{z}$ to $\vec{y}(\vec{z}; \vec{W})$ and is parameterised with the matrix $\vec{W}$. This maps the latent space to a q-dimensional manifold $\mathcal{S}$ embedded in $\mathbb{R}^d$. We shall use an RBF network for this mapping, and $\vec{W}$ represents the adjustable network weights. For this model to be useful, we will usually need $q < d$: in fact, the GTM is most practical when $q = 1$ or 2.

By defining a probability density $p(\vec{z})$ on the latent space, we induce a density $p(\vec{y}|\vec{W})$ in the data space. Since $q < d$, this density will be zero away from the manifold $\mathcal{S}$. This is an unrealistic constraint, since we cannot reasonably expect the data to lie *exactly* on a q-dimensional manifold. Hence we add a noise model for $\vec{x}$. For real-valued data, it is convenient and appropriate to use a spherical Gaussian with variance σ^2, so that the data density conditional on the latent variables is given by

$$p(\vec{x}|\vec{z}, \vec{W}, \sigma) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left\{ -\frac{\|y(\vec{z}; \vec{W}) - \vec{x}\|^2}{2\sigma^2} \right\}. \tag{4}$$

The density in data space is then obtained by integrating out the latent variables; however, for a general model $\vec{y}(\vec{z}; \vec{W})$, this integral is analytically intractable. Let the density $p(\vec{z})$

be given by a sum of delta functions centred on *nodes* $\vec{z}_1, \ldots, \vec{z}_M$ in latent space:

$$p(\vec{z}) = \frac{1}{M} \sum_{j=1}^{M} \delta(\vec{z} - \vec{z}_j). \tag{5}$$

If the nodes are uniformly spread in latent space, this is an approximation to a uniform distribution. The density in data space becomes a simple sum of M Gaussians:

$$p(\vec{x}|\vec{W}, \sigma) = \frac{1}{M} \sum_{j=1}^{M} p(\vec{x}|\vec{z}_j, \vec{W}, \sigma). \tag{6}$$

This is a mixture model where all the kernels have the same mixing coefficient $1/M$ and variance σ^2, and the jth centre is given by $\vec{y}(\vec{z}_j; \vec{W})$. It is a *constrained* mixture model because the centres are not independent but are related by the mapping $\vec{y}$.

The log likelihood for a dataset $\vec{x}_n$, $n = 1, \ldots, N$ is given by

$$\mathcal{L}(\vec{W}, \sigma) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{M} \sum_{j=1}^{M} p(\vec{x}_n|\vec{z}_j, \vec{W}, \sigma) \right\}. \tag{7}$$

This opens the way to determining the parameters $\vec{W}$ and σ using maximum likelihood.

In our modified GTM model, the variance estimates depend on the training data points $\vec{x}$, so the log likelihood becomes:

$$\mathcal{L}(\vec{W}) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{M} \sum_{j=1}^{M} p(\vec{x}_n|\vec{z}_j, \vec{W}, \sigma_n(\vec{x})) \right\}. \tag{8}$$

In the EM algorithm, the basic formula for the E-step remains the same as the standard GTM (though the computation of $p(\vec{x}_n|\vec{z}_j, \vec{W}^{(m)})$ differs in detail). In fact, the posterior probability of the n–th data point being generated by the j–th latent point is:

$$\begin{aligned} R_{jn}^{(m)}(\vec{W}^{(m)}) &= P^{(m)}(\vec{z}_j|\vec{x}_n, \vec{W}^{(m)}) \\ &= \frac{p(\vec{x}_n|\vec{z}_j, \vec{W}^{(m)})}{\sum_{j'=1}^{M} p(\vec{x}_n|\vec{z}_{j'}, \vec{W}^{(m)})}. \end{aligned} \tag{9}$$

However, the M-step is different; from the complete-data log likelihood:

$$\langle \mathcal{L}_{comp}(\vec{W}) \rangle = \sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(\vec{W}^{(m)}) \ln\{p(\vec{x}_n|\vec{z}_j, \vec{W})\}. \tag{10}$$

Maximizing the expectation of the complete-data log likelihood (10) with respect to $\vec{W}$ gives:

$$\sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(\vec{W}^{(m)}) \frac{\|\vec{W}^{(m+1)}\phi(\vec{z}_j) - \vec{x}_n\|}{\sigma_n^2} \phi^T(\vec{z}_j) = 0. \tag{11}$$

Solving the above equation, we get

$$\mathbf{\Phi}^T \vec{G}^{(m)} \mathbf{\Phi} (\vec{W}^{(m+1)})^T = \mathbf{\Phi}^T \vec{R}^{(m)} \vec{T}, \tag{12}$$

where $\mathbf{\Phi}$ is the $M \times K$ RBF design matrix with elements $\mathbf{\Phi}_{ji} = \phi_i(x_j)$, $\vec{T}$ is the $N \times d$ data matrix, $\vec{R}$ is an $M \times N$ responsibility matrix with elements $\frac{R_{jn}}{\sigma_n^2}$, and $\vec{G}$ is an $M \times M$ diagonal matrix with elements $G_{jj} = \sum_{n=1}^{N} \frac{R_{jn}(W)}{\sigma_n^2}$.

The result in (12) remains the same as the standard GTM. Since in our new model σ_n^2 is dependent on n, it cannot be eliminated from the equation as if it was a constant as in the standard GTM model.

In the standard GTM, the E-step also estimates σ, but in our new model the variance estimates are derived from confidence values. The simplest assumption that we can make is that variances are the inverse of the confidences, so that $\sigma_n^2 = \frac{1}{C_n}$ where C_n is a confidence value of the $\vec{x}_n$ data point. With this definition, σ_n^2 ranges from 1 to ∞. It is helpful to introduce a constant, K, to control the scaling of the variance:

$$\sigma_n^2 = K\sigma_n^{*2} = \frac{K}{C_n}. \tag{13}$$

Substituting (13) into (4) gives the following equation:

$$p(\vec{x}|\vec{z}, \vec{W}, \sigma^*) = \frac{1}{(2\pi K\sigma^{*2})^{d/2}} \exp\{-\frac{||y(\vec{z}_i; \vec{W}) - \vec{x}||^2}{2K\sigma^{*2}}\}. \tag{14}$$

The value of K can be estimated by using maximum likelihood: maximizing (10) with respect to K gives the following equation to solve:

$$\sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(\vec{W}^m)(-\frac{d}{2K} + \frac{1}{2}\frac{\|\vec{W}\phi(\vec{z}_j) - \vec{x}_n\|^2}{K^2\sigma_n^{*2}}) = 0 \tag{15}$$

Therefore, the re-estimation equation for K becomes

$$\begin{aligned} K^{(m+1)} &= \frac{1}{Nd} \\ &\sum_{n=1}^{N} \sum_{j=1}^{M} R_{jn}^{(m)}(\vec{W}^{(m)}) \frac{\|\vec{W}^{(m+1)}\phi(\vec{z}_j) - \vec{x}_n\|^2}{(\sigma_n^{*(m)})^2}, \end{aligned} \tag{16}$$

which is similar to the re-estimation of σ in a standard GTM.

SYNTHETIC DATASETS

In this section we discuss the application of the two modified visualisation techniques to some synthetic data.

Neuroscale

The synthetic dataset is a 2-dimensional flat sheet in a 3-dimensional space, with random noise disturbing the sheet in the third dimension (see Figures 1, 2 and 3). The first two dimensions (x and y axes) represent the data manifold spread evenly in space and the z-axis is for the noise. There are 121 data points in total.

A low level of noise variance is used for most of the data, but some points are selected for larger additions of noise variance; they may be considered as *outliers*. The number of outliers vary between 10 and 20 percent of the dataset. The outlier's variance σ^2 is varied between 1 to 10. The confidence values are scaled so that the minimum is 0. Two NeuroScale models, one based on the standard learning function and the other exploiting the modified STRESS definition (see Section 2.1) are then trained 10 times on the data. Visualisation results are presented in Figures 2 and 3, respectively.

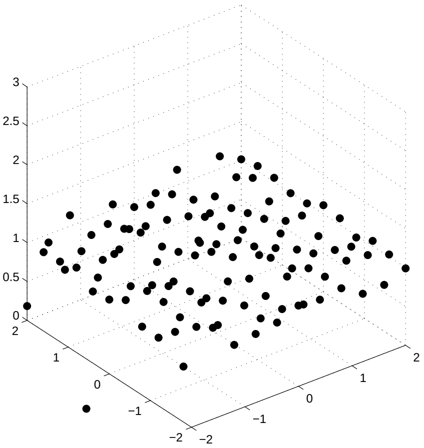

Figure 1: Synthetic data with 10% outliers and $\sigma^2 = 4$.

Table 1 shows the result of the experiment where the STRESS value does not include those of the outliers. It can be observed that the structure of non-outliers has been better preserved with the modified cost function. Table 1 confirms this result in a quantitative way.

GTM

The synthetic data used for the GTM model consists of three Gaussian mixture components in a four-dimensional space. The centres of the components are aligned in a triangle and the number of points in each is 100. When fitting a standard GTM, the noise variance σ^2 is 0.0557. To explore the effect of data confidences, the value C_n is set randomly in the range $[0, 1]$. The modified GTM was trained both with K fixed and learned from the data: the results are shown in Figures 4, 5, and 6.

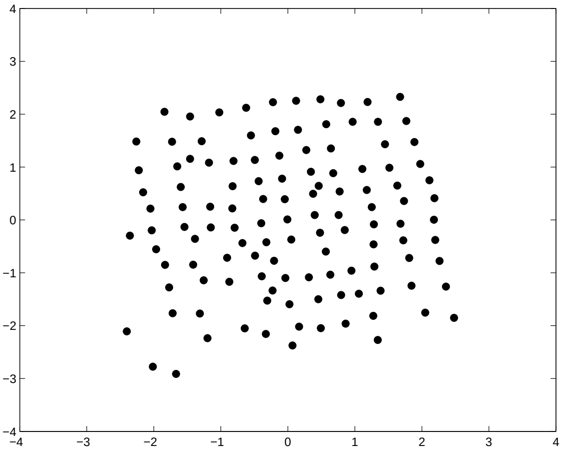

Figure 2: Standard Neuroscale visualisation of synthetic data in Figure 1. Note the distortion of the 2D grid and position of the outlier points with high variance.

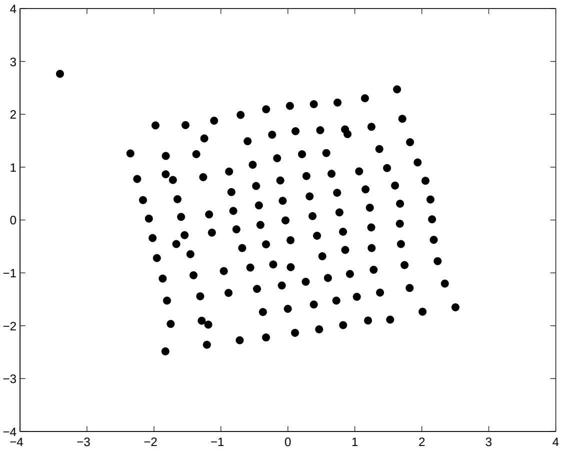

Figure 3: Modified Neuroscale visualisation of synthetic data in Figure 1. Note the positions of the high variance outlier points.

Bigger circles identify larger uncertainty, i.e. smaller confidence values and the smaller circles represent small uncertainty, i.e. larger confidence values.

The dependence of the data visualisation on the K parameter (see Equation 3) is demonstrated testing two different values:for Figure 4 K=0.05, while for Figure 5 K=0.01. It can been see how in the latter case data are more dispersed. The results show that when K=0.01, they have a very clear separation. Comparing the results with Figure 6 where K is optimised by maximum likelihood from equation(16). The estimated K value from maximum likelihood is 0.0167. Figures 5 and 6 give very similar visualisations, which shows that the learning of K from the data is effective.

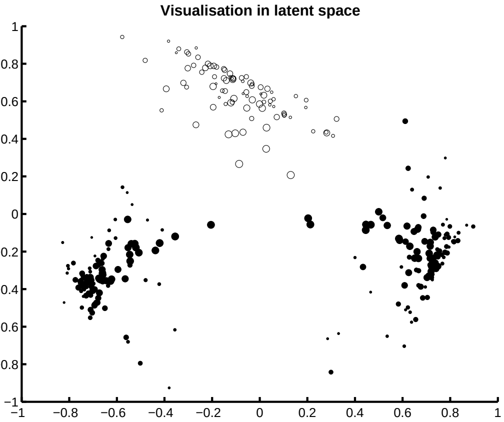

Figure 4: Visualisation with modified GTM and fixed $K = 0.05$.

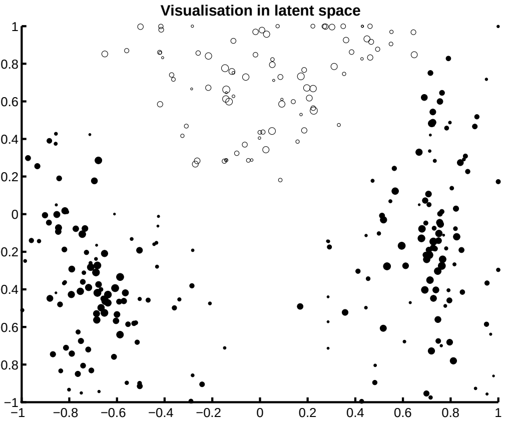

Figure 5: Visualisation with modified GTM and fixed $K = 0.01$.

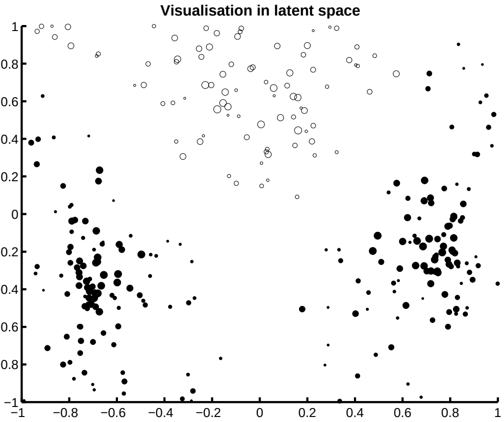

Figure 6: Visualisation with modified GTM and estimated K, final value 0.0167.

		Standard NSC		Modified NSC	
Outliers		Stress value		Stress value	
%	σ^2	mean	Variance	mean	Variance
10	1	0.41	0.01	0.37	0.018
	2	0.59	0.08	0.49	0.016
	3	0.84	0.22	0.48	0.021
	4	0.87	0.07	0.53	0.029
	5	1.04	0.11	0.7	0.08
	6	0.87	0.06	0.8	0.149
	7	1.08	0.13	0.7	0.026
	8	1.35	0.55	0.81	0.148
	9	1.4	0.5	0.73	0.042
	10	1.87	0.43	1.11	0.097
20	1	0.65	0.05	0.38	0.014
	2	1	0.24	0.63	0.062
	3	1.47	0.22	0.79	0.035
	4	1.39	0.39	0.94	0.159
	5	1.39	0.41	0.93	0.159
	6	1.7	0.37	0.97	0.199
	7	1.69	0.38	1.02	0.27
	8	1.84	0.53	1.1	0.169
	9	3.02	2.17	1.17	0.288
	10	3.02	2.17	1.42	0.191

Table 1: Means and variances of calculated STRESS without including outliers.

APPLICATION TO GENOMIC DATA

Dataset

S. coelicolor is a complex mycelial Gram-positive bacterium which undergoes developmental changes leading ultimately to sporulation and production of antibiotics and other secondary metabolites. The 7,825 predicted genes in the linear *S. coelicolor* chromosome include more than 20 gene clusters coding for known or predicted secondary metabolites. The genome also contains an unusually large proportion of regulatory genes, Bentley et.al. (16).

The *S. coelicolor* dataset[1] consists of expression data from samples of surface-grown cultures taken at 16, 18, 20, 21, 22, 23, 24, 25, 39 and 67 hours after the inoculation of the growth medium. Two independent sets of cultures were processed and each Cy3-labelled cDNA sample was hybridized against Cy5-labelled *S. coelicolor* genomic DNA (gDNA) as the common reference. The signal in the gDNA channel follows a different distribution from the cDNA channel since the number of copies of the gene in the genome is fixed while the number of RNA copies can vary widely. There were many probes yielding a significant gDNA signal and very little signal in the cDNA channel, resulting in a long (left-hand) tail in the distribution of ratios. Using gDNA as a reference has the advantage of excluding false negatives due to spotting failures and allows results from different experiments to be directly compared.

The preprocessing of the microarray data consisted of: i) correcting the data for spatial effects, Colantuoni et.al. (17), ii) taking the log-ratio of the signal and the reference measurements; iii) applying the quantile method, Bolstad et.al (18), for across-condition normalization (correction implemented using the SMIDA package for R, Wit and McClure(19),; iv) taking the mean of the replicates; v) applying a variance filtering and a low-value filtering to remove genes that are not significantly expressed; and finally vi) normalizing each pattern of gene expression by subtracting the values at the first time step. The last procedure has been applied in order to investigate the gene expression patterns factoring out the absolute expression level.

In addition, Bayesian techniques were used to analyse the 'spots' and provide confidence values for each measurement using the BlueFuse tool[2].

Many of the genes are not significantly expressed in this experiment. We have visualised the 2,489 most expressed genes filtered from the entire dataset. 1,000 randomly selected genes were used to train the models.

To make a comparison, Figures 7 and 8 illustrate the results from both Neuroscale models.

The results using the modified cost function show that the low confidence value genes are spread away from the centre more than the original Neuroscale model. This is because the new model does not take as much account of the low confidence genes in constructing the visualisation space.

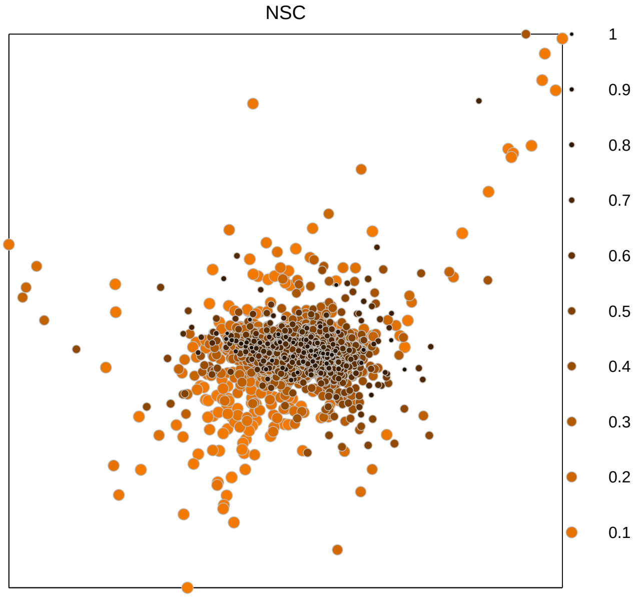

Figure 7: Visualisation of the microarray dataset using modified NeuroScale.

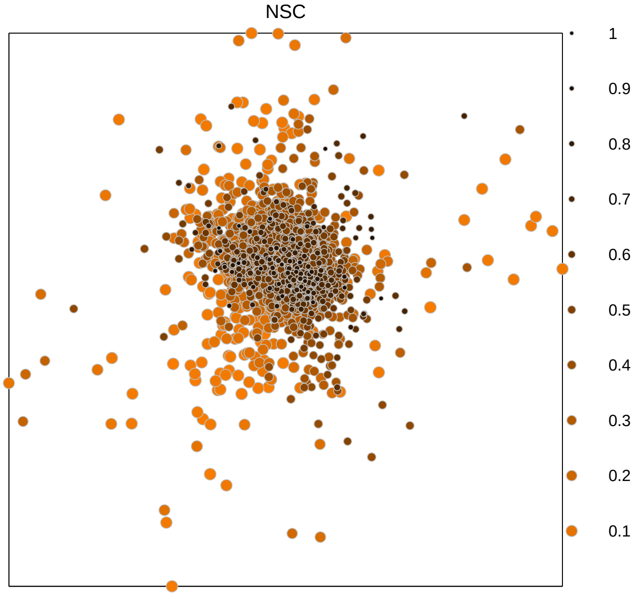

Figure 8: Visualisation of the microarray dataset using standard NeuroScale.

[1] The microarrays used in this study, and associated protocols are described at `http://www.surrey.ac.uk/SBMS/Fgenomics/`.

[2] See `http://www.cambridgebluegnome.com/technology/`

It can also be sen that the neighbourhood relationships between the high and low confidence genes has altered in local regions. This is a crucial point if the relative positions of the visualisation points are going to be used for data mining and inferring relationships amongst genes.

To complete the comparison, Figures 9 and 10 show the use of modified and standard GTM (together with magnification factors). These show a clearer distinction between outliers and confident data points in the modified algorithm. For example, a larger part of the plot in Figure 9 is given over to the bulk of the data, and the regions of high magnification (which are for outliers) contain more points.

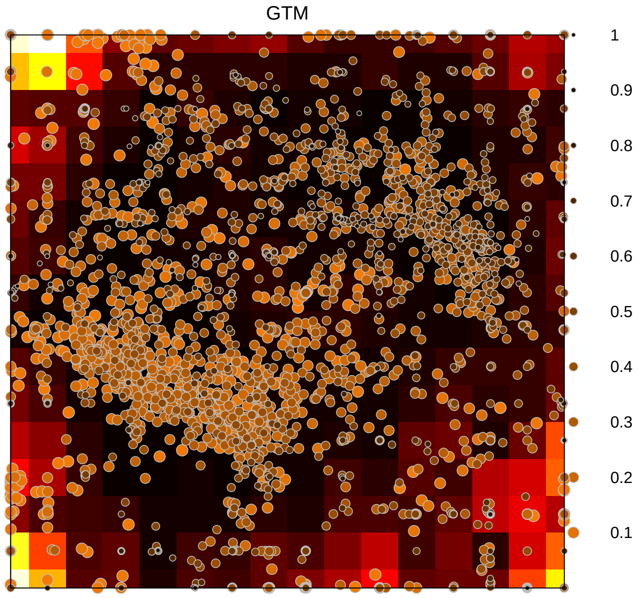

Figure 9: Visualisation of the microarray dataset using modified GTM and estimating K.

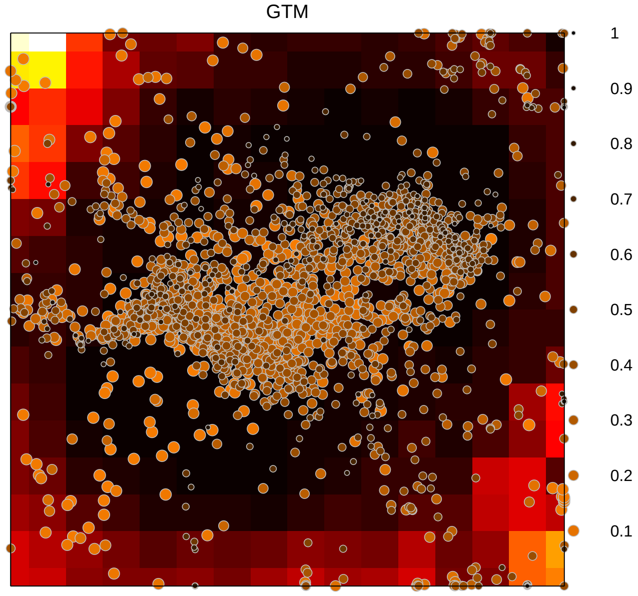

Figure 10: Visualisation the microarray dataset using standard GTM.

CONCLUSIONS

We have proposed a simple approach for incorporating uncertainty of data values into the training of two leading-edge visualisation algorithms for the purposes of data mining. The approach does not add significantly to the computational cost of training and shows benefits in visualisation in allowing the user to view more clearly the more certain data points. These benefits have been shown on synthetic data and a real biological application.

Without the corrections demonstrated in this new visualisation approach, where data integrity is included as part of the analysis process, it would be possible for some very uncertain genes to appear alongside very certain genes in an unmodified visualisation map. This would have the potential for falsely leading a biologist using these visualisation tools for data mining to a wrong conclusion (such that some genes may apparently relate to the same process - falsely - since their expression profiles are mapped to similar regions). However, by allowing the measurement uncertainty to influence the visualisation maps, the data mining can proceed with a higher degree of belief in the tools.

Finally, the present study addressed the problem of data certainty on the basis of a currently available set of microarray data. It is however stressed that the issue of data certainty is not limited to gene expression data or data visualisation tools. Instead, this is a general aspect involving any biomedical measurements and experiments, and the subsequent development of models for diagnosis and prognosis. A deeper consideration, understanding and modelling of these aspects is one of key task of our future work.

ACKNOWLEDGEMENTS

We are grateful to Prof. Colin P. Smith of Surrey University for supplying the genomic dataset. Some of this work was funded under the BBSRC's Toolkit for Functional Genomics Initiative (Grant FGT11407 to CPS) and the BBSRC/EPSRC's Exploiting Genomics Initiative (Grant 92/EGM17737). MS was supported by a grant from the EU Network of Excellence, BioPattern, contract 508803, which funded the development work on uncertainty.

REFERENCES

[1] Sebastiani, P., Gussoni, E., Kohane, I. S., and Ramoni, M. F. 2003. Statistical Science. 18(1): 33-70.

[2] Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C., Peterson, C., and Meltzer, P. 2001. Nature Medicine. 7.: 673–679

[3] Raychaudhuri, S., Stuart, J. M., and Altman, R. 2000. In Pacific Symposium on Biocomputing: 455–466

[4] Apostol, I. and Szpankowski, W. 1999. Journal of Computational Chemistry, 20(10): 1049–1059.

[5] Draghici, S., Graziano, F., Kettoola, S., Sethi, I., and Towfic, G. 2003. Bioinformatics. 22;19(8): 981-986.

[6] Laskaris, N., Fotopoulos, S., and Ioannides, A. A. 2004. IEEE Signal Processing Magazine.May. 66–77.

[7] Lowe, D. and Tipping, M. E. 1997. In Mozer, M. C., Jordan, M. I., and Petsche, T., editors, Advances in Neural Information Processing Systems, 543–549.

[8] Saul, L. K. and Roweis, S. T. 2003. Journal of Machine Learning Research. 4.: 119–155.

[9] Bishop, C. M., Svensén, M., and Williams, C. K. I. 1997. Neural Computation 10(1): 215–234.

[10] Hinton, G. and Roweis, S. 2002. In Becker, S., Thrun, S., and Obermayer, K., editors, Advances in Neural Information Processing Systems. 15.: 857-864.

[11] Tiňo, P. and Nabney, I. 2002. IEEE Trans. Pattern Analysis and Machine Intelligence. 24: 639–656.

[12] D'Alimonte, D., Lowe, D., and Nabney, I. T. 2005. In CIMED 2005, 8–89, Lisbon. IEE.

[13] Tipping, M. E. and Lowe, D. 1997. In Proceedings of the International Conference on Artificial Neural Networks. 440.: 7-12. IEE.

[14] Nabney, I. T. 2002. Advances in Pattern Recognition. Springer-Verlag, London.

[15] Kohonen, T. 1982. Biological Cybernetics. 43: 59–69.

[16] Bentley, S. D., Chater, K. F., Cerdeno-Tarraga, A. M., Challis, G. L., Thomson, N. R., James, K. D., Harris, D. E., Quail, M. A., Kieser, H., Harper, D., Bateman, A., Brown, S., Chandra, G., Chen, C. W., Collins, M., Cronin, A., Fraser, A., Goble, A., Hidalgo, J., Hornsby, T., Howarth, S., Huang, C. H., Kieser, T., Larke, L., Murphy, L., Oliver, K., O'Neil, S., Rabbinowitsch, E., Rajandream, M. A., Rutherford, K., Rutter, S., Seeger, K., Saunders, D., Sharp, S., Squares, R., Squares, S., Taylor, K., Warren, T., Wietzorrek, A., Woodward, J., Barrell, B. G., Parkhill, J., and Hopwood, D. 2002. NATURE, 417(3): 141–147.

[17] Colantuoni, C., Henry, G., Zeger, S., and Pevsner, J. 2002. Biotechniques, 32(6): 1316–1320.

[18] Bolstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. 2003. Bioinformatics, 19: 185–193.

[19] Wit, E. and McClure, J. 2004. Statistics for Microarrays: Design, Analysis and Inference. Wiley.