Microarray Gene Expression: A Study of Between-Platform Association of Affymetrix and cDNA arrays

Chintanu Kumar Sarmah* and Sandhya Samarasinghe

Centre for Advanced Computational Solutions (C-fACS), Lincoln University, Christchurch, New Zealand

ABSTRACT

Microarrays technology has been expanding remarkably since its launch about 15 years ago. With its advancement along with the increase of popularity, the technology affords the luxury that gene expressions can be measured in any of its multiple platforms. However, the generated results from the microarray platforms remain incomparable. In this direction, we earlier developed and tested an approach to address the incomparability of the expression measures of *Affymetrix*®- and cDNA-platforms. The method was an exploit involving transformation of Affymetrix data, which brought the gene expressions of both cDNA and Affymetrix platforms to a common and comparable level. The encouraging outcome of that investigation has subsequently acted as a motivator to focus attention on examining further in the direction of defining the association between the two platforms. Accordingly, this paper takes on a novel exploration towards determining a precise association using a wide range of statistical and machine learning approaches. Specifically, the various models are elaborately trailed using – regression (linear, cubic-polynomial, loess, bootstrap aggregating) and artificial neural networks (self-organizing maps and feedforward networks). After careful comparison in the end, the existing relationship between the data from the two platforms is found to be nonlinear where *feedforward neural network* captures the best delineation of the association.

Keywords: microarrays; Affymetrix; cDNA; regression; cubic-polynomial; loess; bootstrap aggregating; bagging; artificial neural networks; self-organizing maps; feedforward networks

*Address correspondence to this author at the Centre for Advanced Computational Solutions (C-fACS), Lincoln University, Christchurch, New Zealand; Tel: +64-3-3218377; Fax: +64-3-3253845;
E-mail: sarmahc3@lincoln.ac.nz

1 INTRODUCTION

Microarrays technology reveals an unprecedented view into the biology of DNA. Since the publication of the seminal paper [5] about 15 years ago on quantitative monitoring of gene expression in *Arabidopsis*, microarrays have brought about a paradigm shift about how to conduct research in biologically important entities – a shift from the traditional hypothesis-driven research to information-driven research. With the technological and conceptual advancement of the technology in addition to its increase of popularity, microarrays currently afford the luxury that complex biological questions can be queried using any of its multiple platforms, which include commercial vendors, for example Affymetrix® (Santa Clara, CA, USA) and Agilent® (Palo Alto, CA, USA), and other proprietorial arrays of various laboratories.

As the rapid expansion of microarray technology continues to provide comprehensive biologic characterization of various cellular processes, interests have started to grow for integration of multiple microarray studies that are based on the same technological platform or combining data from different array platforms. Integration of multiple studies carries the potential towards higher accuracy, consistency and robust information mining. Further, the integrated result often allows constructing a more complete and broader picture.

Various comparison studies have been published over the years, and the overall observation on accuracy, reliability and reproducibility of microarray investigations can be summarized as cautious optimism [6]. Sarmah & Samarasinghe [7] provides an useful review, which discusses microarray data integration approaches as well as the underlying issues of integration. In the midst of the relentless chase in finding remedies for the issues of microarray data integration, we recently developed an approach [8] to integrate microarray data tested on two microarray platforms, Affymetrix and cDNA. The study, which had used childhood leukemia data (available in supplementary material), produced encouraging outcomes. The proposed approach delivered simpler applicability in addition to furnishing greater transparency. The overall outcome highlighted that its application is capable of addressing the issue of incomparability of the expression measures of the two microarray platforms. The inspiring cross-platform outcomes led us to focus attention on examining further in the direction of precisely identifying the association between the two platforms. With this motivation, a wide range of statistical as well as machine learning approaches are applied to the microarray data. This paper distinctively presents this novel, elaborate exploration, where the modelling of the data is probed into using – regression models (linear, cubic-

polynomial, loess, bootstrap aggregating) and artificial neural networks (self-organizing maps and feed-forward networks).

2 MODELLING A CROSSOVER

Seven of the children used in the study of Sarmah et al. [8] had been tested on both Affymetrix (*HGU-133A* chip) and cDNA platforms; and similar to cDNA, where the expression level of a gene remains in the form of a tumor-to-healthy ratio, Affymetrix expressions were transformed in that work to *Affymetrix-ratio* (or $Affy_{ratio}$). A set of 822 differentially expressed (DE) genes from either platforms were used in the process; and the conducted transformation caused both cDNA and $\text{Affy}_{\text{ratio}}$ to attain a mutually comparable level. In the work, 10 healthy Affymetrix arrays were used such that $Affy_{ratio}$ for a gene of a patient could be calculated as in equation (1) when expression level of a gene, x from one of the leukemic Affymetrix chips is D, and the average of each gene's expression from the set of 10 healthy Affymetrix chips is H.

$$Affy_{\text{ratio}} = \log_2 \frac{Anti\log\,(D_{x_i})}{\frac{\sum_{x=1}^{10} Anti\log\,(H_{x_i})}{10}} \qquad (1)$$

In this paper, each of the seven leukemic patients' data from either platform is examined here to be modelled and tested for their ability in predicting the outcome for the remaining patients. These entire data are also concatenated in two variables, viz. $\text{Affy}_{\text{ratio}}$ and cDNA, each having 5754 genes (i.e., a patient's 822 DE genes × 7 patients), and are available in suppl. Table 1. Out of 5754 DE expression data, a set of 4504 genes' expressions are randomly selected, which would be applied as a separate training dataset to be used by each of the methods. The remaining 1000 DE (i.e., 5754 – 4504 = 1000) expression data would be used for testing a trained framework, wherever possible.

The expression levels of the individual patients are considered for modelling only to represent each patient's ability to predict for others have there been no other patient's data available to form either the large 'global' set or the random set. It is expected that this, in a way, would help judge the impact of each patient's contribution towards the model building from the larger set. As performance indicators of the retrieved models, *mean square error* (MSE) and *Pearson product-moment correlation coefficient* (*corr. coef.*), symbolised by r, would be used. It would be desirable to have lower MSE-values whereas

higher (positive or negative) *r*-values. *Coefficient of Variation* (CV) is also computed as this useful statistic for comparison reveals the degree of variation from one data series to another.

2.1 Linear model

To begin with, bivariate linear regression is applied to test the strength and predictability of the linear model(s). The results, given in suppl. Table 2, presents the regressional output, along with MSE and r-values, of – (i) the whole dataset, (ii) individual patient tested against the remaining patients, and (iii) the random data. The tilde sign (~) between two variables indicates that the variable succeeding this sign is independent, and is a function of the first variable. All the linear fits are overlaid in **Fig. 1**, which shows that the fits do not vary much from each other.

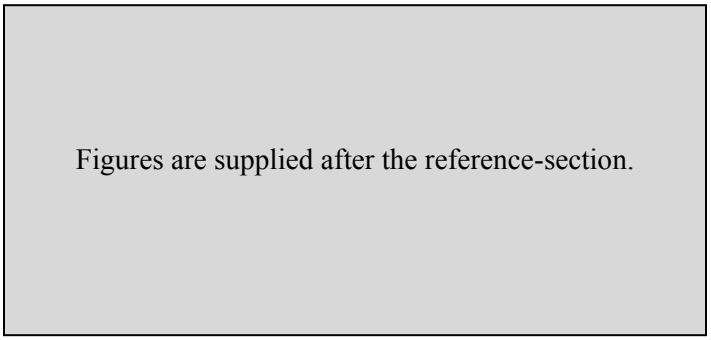


Fig. 1. Overlaying of linear fits

It is not possible to comment on the obtained models as available literature neither has any prescribed benchmark values for such type of linear investigation nor has any results to compare with. Also bivariate linear regression cannot address potential non-linear hidden patterns in seemingly linear data. Therefore, adequate consideration for applying non-linear models is required.

2.2 Consideration for non-linear models

Attempting to use non-linear models on the microarray dataset is futile if just a linear model can adequately represent the data. Therefore, it is necessary to check the need for non-linear methods. Again, it is often difficult to determine such necessity just based on simple visualization as even an apparently linear-looking data can contain underlying non-linear patterns undetectable to the eyes. As a way out, two statistical tests are conducted in which a linear and a cubic polynomial model are used where the latter would query based on the non-linearity of the data.

2.2.1 Extra sum-of-squares test

This test, also known as *F-test using ANOVA*, is based on statistical *hypothesis testing* and ANOVA (analysis of variance); and is computed for the 5754 DE microarray genes. The idea here is that once the data are fit to the two models, goodness-of-fit is calculated as the sum of squares of deviations of the data points from the model. Then, the complexity of the models is measured with the degrees of freedom (*df*), which equal the number of data points minus the number of parameters fit by regression. If the simpler model (the null hypothesis) is correct, the relative increase in the sum of squares approximately equals the relative increase in degrees of freedom. If the more complicated (alternative hypothesis) model is correct, then the relative increase in sum-of-squares (going from complicated to simple model) becomes greater than the relative increase in degrees of freedom. The F-ratio equals the relative difference in sum-of-squares divided by the relative difference in degrees of freedom. This is shown in **equation (2)**.

$$F = \frac{\frac{SS_{null} - SS_{alt}}{SS_{alt}}}{\frac{df_{null} - df_{alt}}{df_{alt}}} = \frac{\frac{SS_{null} - SS_{alt}}{df_{null} - df_{alt}}}{\frac{SS_{alt}}{df_{alt}}} \qquad (2)$$

F-ratios are always associated with degrees of freedom for the numerator and that for the denominator. The F-ratio in the equation has df_{alt} degrees of freedom for the denominator, and $df_{null} - df_{alt}$ degrees of freedom for the numerator. ANOVA computes an F-ratio from which it calculates a probability (*p*)-value. If the obtained p-value is less than the set statistical significance level, usually $\alpha = 0.05$, the alternative (complicated) model fits the data better than the null hypothesis (simpler) model. Otherwise, there is no compelling evidence supporting the alternative model, and so the simpler null model can be accepted.

With the 5754 DE genes, the output rendered a probability less than $2.2e^{-16}$. This suggests that the probability of obtaining a calculated F-value of 84.258 by chance is $2.2e^{-16}$ or smaller. This is highly unlikely; and hence, there is enough possibility that nonlinear model would provide improvement over linear model.

2.2.2 Akaike's Information Criterion

As an alternative approach to F-test and choosing a model with the use of statistical hypothesis testing, *Akaike's information criterion* or AIC [9] is used for comparing the two models. AIC combines maximum likelihood theory, information theory, and the concept of the entropy of information [10]. It is different as well as a distinctly independent approach than the F-test, and does not rely on p-values or the concept of statistical significance. Moreover, F test can only be used to compare nested[1] models, where Akaike's method can be used to compare both nested and non-nested models. It is known in statistics as a penalized log-likelihood, and can be written as shown in **equation 3**.

$$AIC = -2l + 2(p+1)$$
$$l(\mu,\sigma) = -\frac{n}{2}\log(2\pi) - n\log(\sigma) - \frac{\sum(y_i-\mu)^2}{2\sigma^2} \qquad (3)$$

In the equation, p is the estimated coefficients in the model, and 1 is added for the estimated variance. *Log-likelihood*, a measure of comparing the fit of two models, is denoted by l, and the value of it gets higher with better model. A somewhat similar and often used structure of equation is also given in the equation. However, in simple terms, AIC can be defined as a method of comparing alternative specifications by adjusting the error sum of squares for the sample size and the number of coefficients in the model (p), i.e., AIC = log(SSE) + 2(p). While using for comparison, the model with the lowest AIC score is most likely to be a better fit. In case of the 5754 DE genes, polynomial was found to have the lower AIC.

Both the statistical tests above present an indication that non-linear methods may potentially bring improved outcomes. This confers a trust upon exploring the non-linear methods further.

2.3 Non-linear models

2.3.1 Polynomial regression

Polynomial models are useful to investigate the presence of possible curvilinear effects in the response function. Such regression fits a nonlinear relationship to the data where the dependent variable is modelled as an n^{th} order function of the dependent variable. Every polynomial corresponds to a polynomial

[1] When a model is a simpler case of the other, the models are said to be *nested*.

function, and can be represented as shown in **equation 4**, where n is a non-negative integer and a_0, a_1, a_2, ... a_n are constant coefficients.

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_2 x^2 + a_1 x + a_0 \qquad (4)$$

The results of polynomial regression applied to the microarray datasets are given in suppl. Table 3. For comparison of these results with the linear regression of suppl. Table 2, the MSE values can be used such that relative decrease of MSE along with no change or an increase of correlation (positive or negative), r is an indication of better representation of the relationship by a model. The polynomial results are found to be relatively improved compared to the values of linear regression, which is already confirmed while carrying out the statistical tests (i.e., extra sum-of-squares test and AIC) above.

2.3.2 Locally weighted regression

In the methodology of time series, there is an old idea deeply buried where the data measured at equally spaced points in time were smoothed by local fitting of polynomials [11]. Then, the era of contributions came where chronologically Watson [12], Stone [13], Cleveland [14], Hastie & Tibshirani [15] and Cleveland & Devlin [16] introduced as well as streamlined the local fitting methods into the more general case of regression analysis. The curve fitting regression technique introduced by William S. Cleveland [14] and further developed by him and Susan Devlin [16] is called LOWESS, which is *locally weighted regression scatter plot smoothing*. Its derivative, LOESS stands more generally for a local regression, and differs from LOWESS based on the model used in the regression: LOWESS uses a linear polynomial whereas LOESS uses a quadratic polynomial [17]. Many researchers consider LOWESS and LOESS as synonyms.

More descriptively, the method of *locally weighted regression* or *Loess* (aka *Lowess*) can be considered as locally weighted polynomial regression. It combines much of the simplicity of linear least square regression with the flexibility of nonlinear regression. To achieve this, it uses a nearest neighbour algorithm and determines localized subsets of data. Local polynomials of usually first or second degree are fit to these subsets of data using weighted least squares. A user specified *smoothing parameter* (*f*) gives the flexibility to the Loess function, and it is approximately the fraction of points to be used in the computation of each fitted values. There is no single correct value of *f*, and the values can range from 0 to 1. However, different *f* values give different summaries. As Chambers et al. explains [18], a small value of

f gives a very local summary of the middle of the distribution of *y* in the neighbourhood of x. Such value tends to force the function to excessively conform to the data, and only points whose abscissas are relatively close to x_i determine y_i. This produces high resolution, but a lot of noise. For large values of *f*, the summary is much less local. In this case, there is low resolution with less noise. With respect to the smoother-line in the scatter plot, the larger the *f*-value gets, the lesser becomes the wiggle in response to the fluctuations in the data, or vice versa.

The subset of data used in each weighted least squares fit is comprised of the data whose explanatory variables are closest to the point at which the response is being estimated. Based on the weight function, closer a data remains to the point of estimation, higher the weight it attains. Therefore, a local model can be considered to have the most influence by the nearby data than the points that are further apart. Any weight function can be used in this purpose as long as it satisfies the properties listed in Cleveland [14].

Application of loess method to the DE genes of the microarray data is quite possible. However, on the basis of the principles involved in loess, it is found that any attempt of finding goodness of its fit through measures such as r and MSE is rather practically meaningless. The reason lies in the explanation of the loess method given above. In loess, a locally weighted estimate of a specified degree over a given fraction of the data is computed, where the region over which the fit is performed slides to the right in each iteration. The combination of all these individual results produces the final fit. Again, this makes little practical sense to determine the form of the loess model; and because of that, measures such as r and MSE becomes pointless for loess models. It may be possible to estimate some r- like measures for the loess model by carefully deriving from its definition, and MSE-like estimate by extension, but it may not actually be meaningful as unlike regression, which produces pre-specified, parametric model for which the parameters are calculated from the data, loess lacks any such analogue, and the entire loess fit is estimated solely from the data without producing a single coherent model: with the change of either the span of the data or the degree of the local fit or both, there would be change in the r- and MSE-like estimates.

After giving careful consideration, application of loess is subsequently avoided because of its data-driven attribute - as none of the outcomes can be considered to be in line with the results of the investigations using the other methods. Nevertheless, to visualize how the method would contribute varying from the linear and polynomial distribution, it is applied for the 5504 DE genes using 0.75 as the smoothing parameter and 2 as the degree of the local polynomial. And, the output is graphically present-

ed in **Fig. 2** with the help of *ggplot2* [19], an implementation based on the *Grammar of Graphics* [20]. The comparative figure shows that the loess and the polynomial fits are close to each other and are relatively better fits than the linear model.

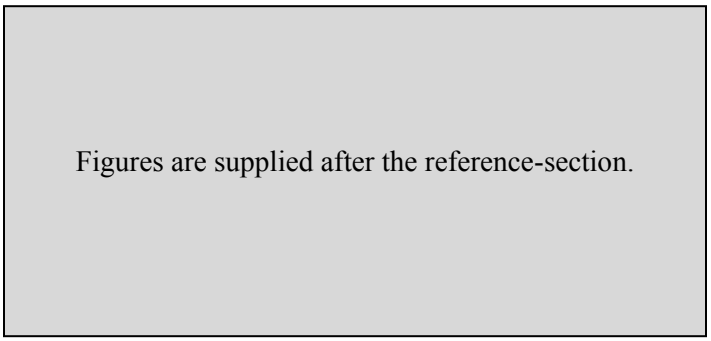


Fig. 2. Microarray data with linear, polynomial and loess fit

2.3.3 Bootstrap Aggregating

Bradley Efron [21] invented the concept, *bootstrapping*, which refers to a group of metaphors that generally mean: a self-sustaining process that proceeds unaided. Bootstrap is the most recently developed, computer-intensive approach to retrieve statistical inference, and the process lets the dataset reveal its true self. This is achieved by sampling from the empirical distribution of the data without replacing or adding to the data. From the concept of bootstrapping, *bootstrap aggregating* or *bagging* originates. Bagging is an ensemble method, i.e., a method of combining multiple predictors. It is useful for avoiding model overfitting to data with variance reduction, and has been in use for a varied range of microarray studies [22-24]. This machine learning meta-algorithm, introduced by Leo Breiman [25], is used here to investigate the microarray data. To apply bagging to the microarray data, a computational algorithm is constructed, and is given in the following box.

Bootstrap algorithm:

- Let the original sample be L = (x_1, x_2, ... , x_n), where x_i is drawn from an empirical population distribution, $\hat{F}$.
- Repeat *B* times :
  - Generate a sample L_k of size *n* from *L* by sampling with replacement.
  - Compute $\hat{\theta}^*$ for x^*
- The corresponding bootstrap values are : $\hat{\theta}^* = \left(\hat{\theta}_1^*, \hat{\theta}_2^*, ..., \hat{\theta}_B^*\right)$
- Use the values of $\hat{\theta}^*$ to calculate the parameters of interest.

Notations:

θ = Parameter; * = Data generated from bootstrapping; ^ = An estimate

The results of implementing the algorithm are provided in suppl. Table 4. The table shows that although the obtained r-values are relatively comparable to the earlier results of linear and polynomial regression, the values of MSE are found to be much higher, with low standard deviation. This is considered to be an indication that even though the method of bootstrap aggregating is a useful method in a number of published microarray studies, it may not be suitable for applying in the current context.

2.3.4 Self-Organizing Maps

Machine learning approaches, such as *Artificial Neural Networks* (ANN) are considered to be effective computational methods that enable efficient capture of the trends potentially available in the data. Pioneered by Rosenblatt [26], Widrow & Hoff [27] and Widrow & Stearns [28], ANN represent a computational tool, based on the properties of biological neural systems. *Self-organizing map* (SOM) is the most widely used unsupervised neural networks. Introduced by Teuvo Kohonen [1-3], it uses only the input data and projects it onto one- or two-dimensional grid for meaningful interpretation of its inherent structure and patterns as well as for visual validation [4].

SOMs are considered highly efficient techniques for exploratory data analysis. Based on the principles and the inherent properties, this exploratory technique is extended to broaden its use towards employing it as a prediction tool.

Each neuron of a trained SOM includes a specific set of datapoints. In a 2D space, such a neuron holds a final weight and the weight bears two components, one in *x*- and the other in *y*-direction. With this as a preface, a computational algorithm was constructed for implementation, and is given below in the box.

Algorithm used for SOM:

- Let the training dataset for microarray be L = (x_1y_1, x_2y_2, ... , x_ny_n) where x_i and y_i is $\text{Affy}_{\text{ratio}}$ and cDNA respectively.
- Train the data using the regular SOM algorithm [1-4].
- Use the test dataset, T = (a_1b_1, a_2b_2, ... , a_nb_n) where a_i and b_i is test data from $\text{Affy}_{\text{ratio}}$ and cDNA respectively.
- For each a_i :
  - Advance in *x*-direction by the value, a_i
  - in *y*-space, search for the closest neuron, N_c
  - Average the cluster of y_i -values that come under N_c. This $\ddot{y}_i$ represents the corresponding SOM-output of the a_i value.

The algorithm is implemented using *Matlab*®, 2010a (The MathWorks Inc., Massachusetts, USA), and an instance of its implementation is given in **Fig. 3** where the final positions of the neurons are shown when SOM-training is completed. The training and test data used belong to the random drawn 4504 and 1000 datasets respectively. In the figure, the neuron positions are demarcated by rectangles while the positions of the training and test data are shown as dots (.) and crosses (×), respectively. The obtained outputs are given in suppl. Table 5. It is evident from the table that the results are better than that of bootstrap aggregating. However, they are not as good as those of either polynomial or linear regression.

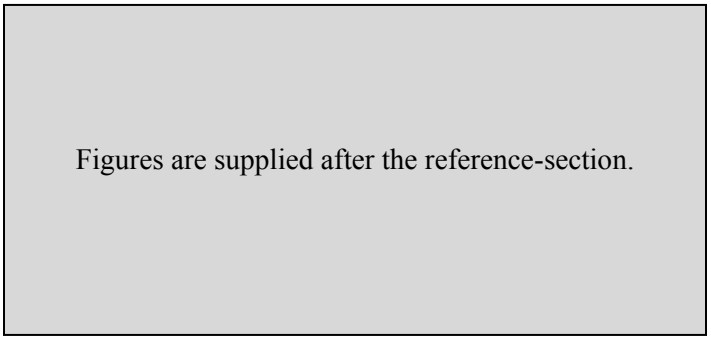


Fig. 3. Final neuron-positions along with training- and test-data

2.3.5 Feedforward Neural Network

Supervised neural networks are the mainstream of neural network development, and the *feedforward neural networks* fall in the category of supervised networks. Invented by Frank Rosenblatt [26], these networks are considered to be effective computational methods that enable efficient capture of the trends potentially available in the data.

In multi-layer feedforward networks, it is said that high number of neurons with multiple layers often tends to create undesirable complexity. The same is empirically experienced while working with our microarray data. Subsequently, a simple feedforward network is finally preferred, and it consists of one neuron in the middle layer, in addition to the input- and the output-layer. Again, Matlab® was used to do the required computations. Various parameters used in these calculations are given below:

i) Training function: *Levenberg-Marquardt* method [29] is used as learning method. It is a second-order method, and relies on both first and second derivative of error (slope and curvature) while searching for the optimum weights. The method is considered as a hybrid algorithm as it combines the advantages of *steepest descent* and *Gauss-Newton* methods. Levenberg-Marquardt algorithm is a fast method, and it primarily makes use of the Gauss-Newton method; but encountering situations where the 2nd derivative is negative, it reverts to the steepest descent method, and uses only the first derivative.

ii) Transfer function: Transfer functions calculate a neural layer's output from its net input. Here, hyperbolic tangent sigmoid function [30] is used. It can be mathematically represented as given in **equation 5** where *la* stands for *linear activation* of a neuron.

$$\tanh(la) = \frac{2}{1+e^{-2\times la}} - 1 \qquad (5)$$

The final outputs obtained from the feedforward network are given in suppl. Table 6. These results are indeed better than all the other approaches examined so far.

3 SUMMARY OF RESULTS

Various statistical and machine learning approaches rigorously applied in this work; and broadly, the available results provided in suppl. Table 2 - 6 can be studied by comparing the model outputs concerning the whole and the random dataset. The gist of these results is summarised below in Table 1. With the simple neural architecture, the feedforward network is able to present the best results, while cubic-polynomial delivers the next best set of results. The summarised results also indicate that despite its enormous potential, bootstrap aggregating method fails to deliver a comparable outcome than the rest of the methods. The self-organizing maps (SOM) are used by various researchers to constitute a very powerful and unsupervised data visualization technique. This technique is probed into and redesigned to make it operational to address our task, which otherwise falls outside of its usual application environment. This redesigning of SOM's application makes it capable of bringing better outcomes than the bagging method, but comes out to be relatively less effective than the remaining approaches.

Table 1 Summary of results

Model	**Whole dataset**		**Random dataset**			
			Training set		Test set	
	MSE	*r*	*MSE*	*r*	*MSE*	*r*
Linear	0.6013	0.5886	0.6172	0.5892	0.5299	0.5771
Polynomial (cubic)	0.5842	0.6042	0.5979	0.6064	0.5140	0.5835
Bootstrap aggregating (bagging)	1.4870	0.5938	1.4806	0.5956	1.5588	0.5872
Self Organizing Maps (SOMs)	0.8994	0.4650	0.9316	0.4643	0.8586	0.4405
Feedforward network	0.5187	0.6253	0.5400	0.6267	0.4962	0.6042

A look at the suppl. Table 2 - 6 also suggests that at the level of individual patients, the predictive gene expression of atleast one patient, viz., patient number 78, produces at times results that tend to exceed the range of the outputs obtained by the others. However, it is difficult to reason this as in addition to observing the best housekeeping standards, the elaborate data quality analysis prior to the investigation indicated presence of no faults in any of these arrays. The predicted results of the remaining patients are, however, found to be more or less similar.

4 CONCLUSION

This research is a novel exploration in precisely determining the relationship between two common microarray platforms, Affymetrix and cDNA. The major highlight of this work is its distinctively extensive search implementing a wide range of statistical as well as machine learning approaches towards drawing the closest association between the two platforms. The rigorously applied approaches probe into whether and how these methods would be useful when applied to microarray data in a situation when they come from two separate platforms. The resultant output of the study suggests that the relation between the two microarray platforms is non-linear; and given a gene's expression level in one platform, there is a possibility that a feed-forward neural network would provide more accurate expression value of the gene in the other platform compared to the rest of the approaches trialled. The conducted work maintains the highest housekeeping standards, besides carrying out a series of trials and testings with the applications, methods and algorithms. Finally, this process of interrogation is believed to have offered its own contribution towards bridging microarray platforms, and would possibly serve useful pointers in other microarray studies.

ACKNOWLEDGEMENTS

The authors thank Dr. Daniel Catchpoole, Head of Tumour Bank, The Children's Hospital at Westmead, Australia for providing with the microarray data used in this work.

REFERENCES

[1] T. Kohonen, Clustering, taxonomy and topological maps of patterns, in: Proceedings of the 6th International Conference on Pattern Recognition, Munich, Germany, 1982, pp. 114-128.

[2] T. Kohonen, Self-organized formation of topologically correct feature maps, Biological Cybernetics, 43 (1982) 59-69.

[3] T. Kohonen, The self-organizing maps, Neurocomputing, 21 (1998) 1-6.

[4] T. Kohonen, Self-Organizing Maps, 3rd ed., Springer, NY, 2001.

[5] M. Schena, D. Shalon, R.W. Davis, P.O. Brown, Quantitative monitoring of gene expression patterns with a complementary DNA microarray, Science, 270 (1995) 467–470.

[6] P. Cahan, F. Rovegno, D. Mooney, J.C. Newman, G.S.L. III, T.A. McCaffrey, Meta-analysis of microarray results: challenges, opportunities, and recommendations for standardization, Gene, 401 (2007) 12-18.

[7] C.K. Sarmah, S. Samarasinghe, Microarray data integration: frameworks and a list of underlying issues, Current Bioinformatics, 5 (2010) 280-289.

[8] C.K. Sarmah, S. Samarasinghe, D. Kulasiri, D. Catchpoole, A simple Affymetrix ratio-transformation method yields comparable expression level quantifications with cDNA data, in: C. Ardil (Ed.) International Conference on Bioinformatics and Bioengineering, World Academy of Science, Engineering and Technology, Cape Town, South Africa, 2010, pp. 78-83.

[9] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control, 19 (1974) 716-723.

[10] K.P. Burnham, D.R. Anderson, Model selection and multi-model inference: a practical information-theoretic approach, 2nd ed., Springer-Verlag, New York, 2002.

[11] F.R. Macaulay, The smoothing of time series, National Bureau of Economic Research, Inc., California, 1931.

[12] G.S. Watson, Smooth regression analysis, Sankhya: The Indian Journal of Statistics, Series A, 26 (1964) 359-372.

[13] C.J. Stone, Consistent nonparametric regression, The Annals of Statistics, 5 (1977) 595-620.

[14] W.S. Cleveland, Robust locally weighted regression and smoothing scatterplots, Journal of the American Statistical Association, 74 (1979) 829-836.

[15] T. Hastie, R. Tibshirani, Generalized additive models, Statistical Science, Vol. 1, No. 3 (Aug., 1986), pp. 297-310 1(1986) 297-310.

[16] W.S. Cleveland, S.J. Devlin, Locally weighted regression: an approach to regression analysis by local fitting, Journal of the American Statistical Association, 83 (1988) 596-610.

[17] A.I. Saeed, N.K. Bhagabati, J.C. Braisted, W. Liang, V. Sharov, E.A. Howe, J. Li, M. Thiagarajan, J.A. White, J. Quackenbush, TM4 microarray software suite, Methods Enzymol, 411 (2006) 134-193.

[18] J.M. Chambers, W.S. Cleveland, B. Kleiner, P.A. Tukey, Studying two-dimensional data, in: Graphical methods for data analysis, Wadsworth & Brooks/Cole, California, 1983, pp. 75-127.

[19] H. Wickham, ggplot2: elegant graphics for data analysis, 2nd ed., Springer, New York, 2009.

[20] L. Wilkinson, The grammar of graphics, 2nd ed., Springer-Verlag, New York, 2005.

[21] B. Efron, Bootstrap methods: another look at the jackknife, The Annals of Statistics, 7 (1979) 1 – 26.

[22] S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure, Bioinformatics, 19 (2003) 1090-1099.

[23] D.N. Politis, Bagging multiple comparisons from microarray data, in: I. Mandoiu, R. Sunderraman, A. Zelikovsky (Eds.) Bioinformatics Research and Applications, Springer, Berlin, Heidelberg, 2008, pp. 492-503.

[24] C. Lu, A. Devos, J.A.K. Suykens, C. Arus, S.V. Huffel, Bagging linear sparse Bayesian learning models for variable selection in cancer diagnosis, IEEE Trans Inf Technol Biomed, 11 (2007) 338-347.

[25] L. Breiman, Bagging predictors, Machine Learning, 24 (1996) 123 - 140.

[26] F. Rosenblatt, Principles of neurodynamics: perceptrons and the theory of brain mechanism, Spartan Books, Washington, DC, 1962.

[27] B. Widrow, M.E. Hoff, Adaptive switching circuits, IRE WESCON Convention Record, 4 (1960) 96-104.

[28] B. Widrow, S. Stearns, Adaptive signal processing, Prentice Hall, Englewood Cliffs, NJ, 1985.

[29] J.J. More, The Levenberg-Marquardt algorithm: implementation and theory, in: G.A. Watson (Ed.) Numerical Analysis, Springer-Verlag, Berlin, Heidelberg, New York, 1977, pp. 105-116.

[30] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Alkon, Accelerating the convergence of the back-propagation method, Biological Cybernetics, 59 (1988) 257-263.

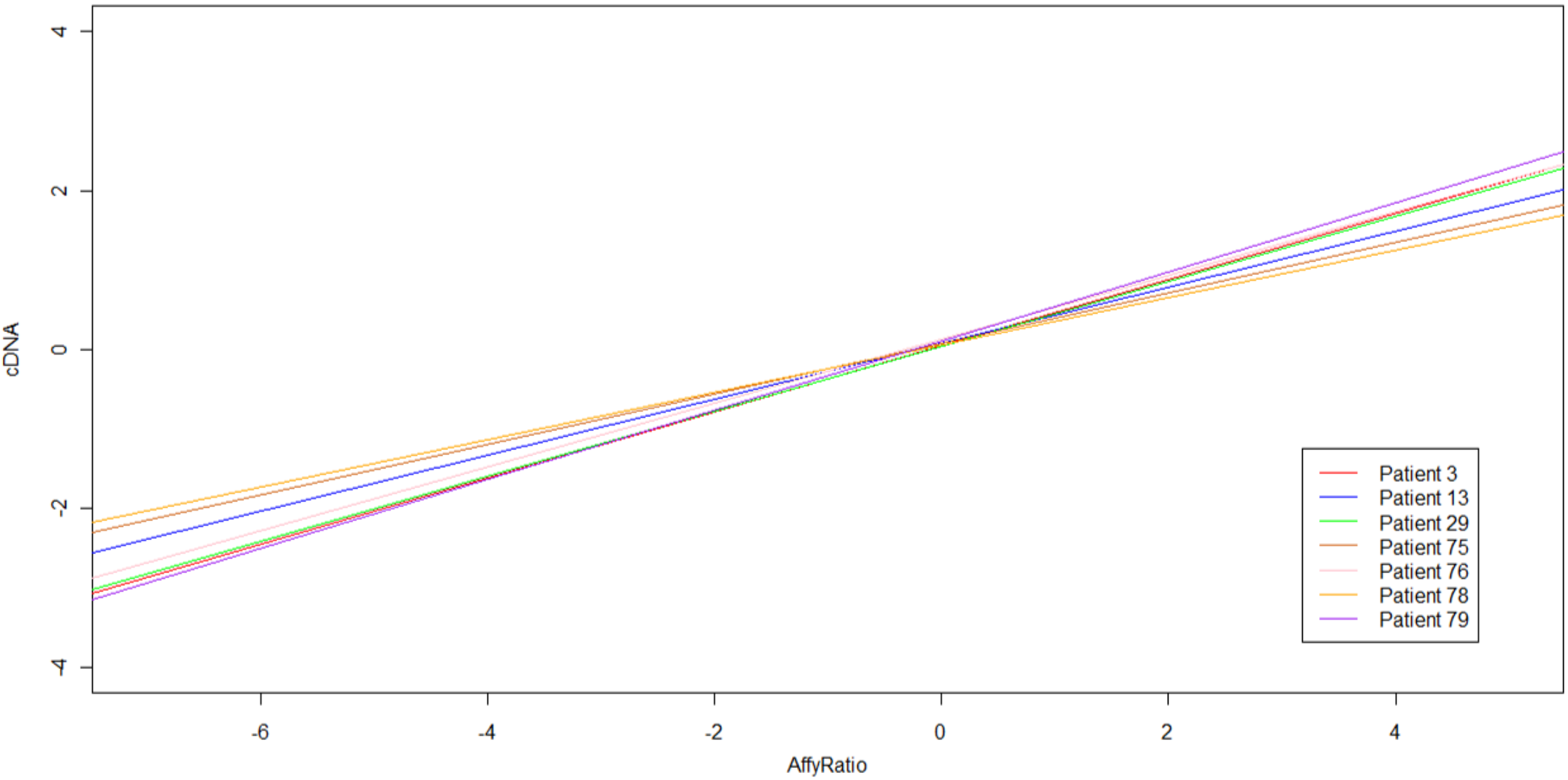
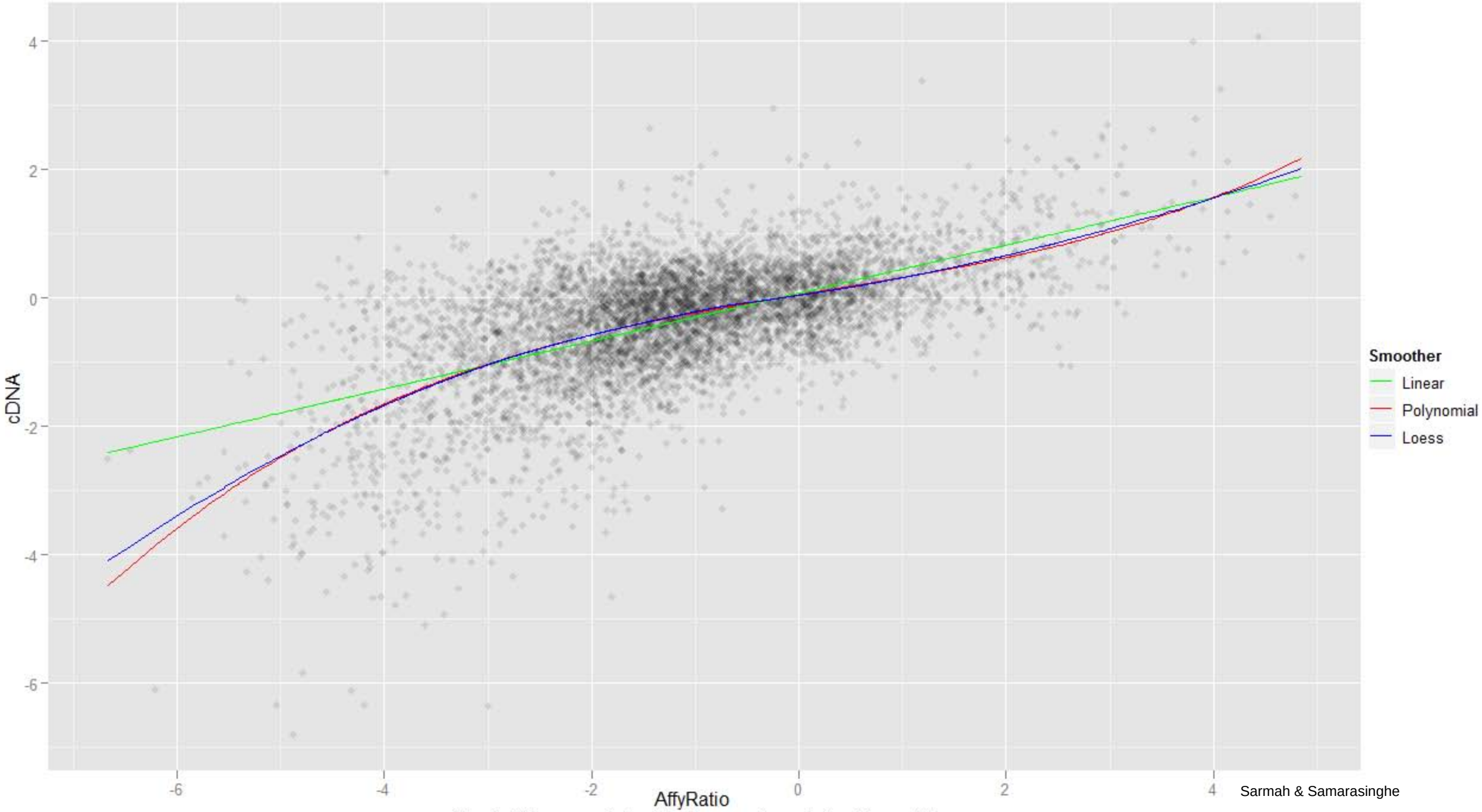
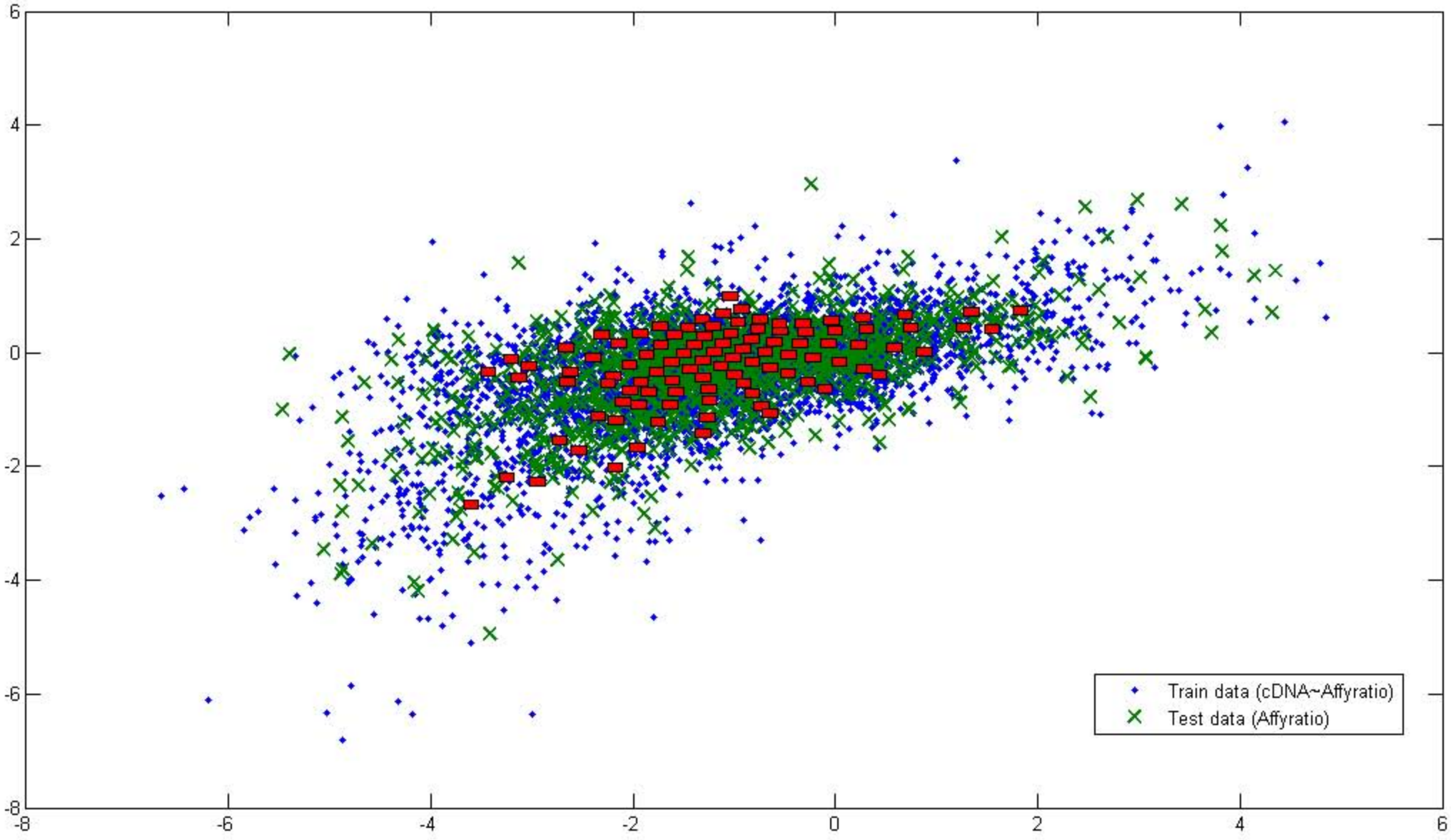


Fig. 1. Overlaying of linear fits



Sarmah & Samarasinghe

Fig. 2. Microarray data with linear, polynomial and loess fit



Sarmah & Samarasinghe

Fig. 3. Final neuron-positions along with training- and test-data