Genetic Programming and Evolvable Machines, 5 (3): 251-257, September 2004

# Genetic Programming for Mining DNA Chip data from Cancer Patients

W. B. Langdon and B. F. Buxton

Computer Science, University College, Gower Street, London, WC1E 6BT, UK
{W.Langdon, B.Buxton}@cs.ucl.ac.uk
http://www.cs.ucl.ac.uk/staff/W.Langdon, /staff/B.Buxton
Tel: +44 (0) 20 7679 4436, Fax: +44 (0) 20 7387 1397

**Abstract.** In machine learning terms DNA (gene) chip data is unusual in having thousands of attributes (the gene expression values) but few ($< 100$) records (the patients). A GP based method for both feature selection and generating simple models based on a few genes is demonstrated on cancer data.

## 1 Introduction

The problem of over fitting is a dominant concern with machine learning approaches to DNA chip data. These medical data are characterised by class imbalance, non-linear response, high noise, large numbers of attributes and few examples. Last year [Pomeroy *et al.*, 2002] published DNA chip data for 60 cancer patients. Their attempts to model the data using unsupervised learning techniques (self organising maps) were unsuccessful at predicting patient survival (page 441) however they claim statistically significant success using nearest neighbour and other supervised learning techniques. [Li *et al.*, 2001] obtained good results using three nearest neighbours after selecting genes with a multi-run evolutionary approach on similarly sized DNA expression data. ([Valafar, 2002] surveys both supervised and unsupervised data mining techniques used with microarray data.)

Genetic programming (GP) has been used with DNA chip data previously. For example [Gilbert *et al.*, 2000] used it with time sequences of expression levels in yeast to ascribe functions to genes. However here we consider much smaller static expression data sets to classify, either the patient as a whole or specific tissues (e.g. cancer tissues). Classifications might in future suggest particular treatments. Also identification of predictive genes may aid understanding of causes and hence treatments of diseases. While [Moore *et al.*, 2002] showed GP can be used to fit DNA expression data, we show it can be used to find very simple models, with consequently little danger of over fitting. There follows an experiment which shows linear discrimination using two genes found by GP (from a total of 7129) gives similar performance to that given in [Pomeroy *et al.*, 2002].

## 2 Cancer DNA chip data

[Pomeroy *et al.*, 2002]'s "Gene Expression-Based Classification and Outcome Prediction of Central Nervous System Embryonal Tumors" data was copied from `http://www-genome.wi.mit.edu/mpr/publications/projects/CNS/Pomeroy_et_al_0G04850_11142001_datasets.zip`. Gene descriptors and patient identification were removed from `Dataset_C_MD_outcome.gct` and `Dataset_C_MD_outcome.xls`, which were then merged and transposed. There are 7129 signed integer gene expression values for each of the 60 patients, of whom 39 survived.

## 3 Leave One Out Cross Validation

$n$-fold cross validation allows one to estimate how well a learning technique will perform on unseen data without reserving a sizeable volume of data for testing. The data is divided into (say $n = 5$) equal numbers of records, known as folds. The learning system is trained on all records except one fold. Its performance on the remaining records is measured. Then the same system is trained again but this time leaving out another fold. The performance of the learning system is estimated by taking the mean of the (five) performance measurements. As always, care must be taken that their is no cross contamination which might allow knowledge about patterns in the test data (i.e. the test folds) to leak into the training process.

Since there are only a very few training examples, we go to the extreme of having as many folds as their are records (60). I.e. the GP is trained sixty times using 59 patient records. Leave one out gives an almost unbiased estimate but its variance may be high [Kohavi, 1995]. Each time the performance of the evolved model is measured by seeing if it can predict the survival of the remaining patient. Because genetic programming is a stochastic process, the GP is run ten times (cf. [Carvalho and Freitas, 2003]), making a total of 600 runs for each stage in the experiment.

## 4 Genetic Programming

The individuals in the GP population consist of five trees. At crossover one of the five is chosen and size fair crossover [Langdon, 2000] occurs only between that tree in first and second parents. The remaining four trees are copied unchanged from the first parent [Langdon, 1998]. The GP's prediction is positive if the sum of the floating point values returned by the five trees is greater than or equal to zero. While, in this work, having multiple trees, does not directly increase the power of the representation, it may help by making it easier for different parts of the individuals to evolve to specialise in solving different parts of a problem. [Soule, 1999], [Rodriguez-Vazquez and Oliver-Morales, 2003] and ourselves [Langdon and Buxton, 2001a; Langdon and Buxton, 2001b; Langdon *et al.*, 2001] report some success with it. Gene expression values are represented as floating point numbers. We retained our use of area under the (convex hull of) the receiver operating characteristics (ROC) curve as the fitness

measure [Langdon and Buxton, 2001c]. But for speed and simplicity only a single threshold point on the curve was used. This collapses fitness to $\frac{1}{2}(TP+(1-FP))$, where $TP$=true positive rate and $FP$=false positive rate. I.e. the mean of the accuracy on the positive examples and the accuracy on the negative. See Table 1 for other GP parameters.

**Table 1.** GP DNA Chip gene mining

| | |
|---|---|
| Objective: | Find a simple rule predicting patient survival from gene expression data |
| Function set: | Max Min MaxA MinA MUL ADD DIV SUB IFLTE |
| Terminal set: | DNA chip expression values. |
| Fitness: | $\frac{1}{2}$ fraction of survivors predicted + $\frac{1}{2}$ fraction treatment failures correctly predicted |
| Selection: | generational (non elitist), tournament size 7 |
| Wrapper: | $< 0 \Rightarrow$ patient survival is predicted |
| Pop Size: | 500 |
| Program size: | up to 1000. In two gene runs, all programs were size 5 or 9. |
| Initial pop: | Each individual comprises five trees each created by ramped half-and-half (2:6). Each initial tree limited to 300 |
| Parameters: | 50% size fair crossover, crossover fragments $\leq$ 30 [Langdon, 2000] 50% mutation (point 22.5%, constants 22.5%, shrink 2.5% subtree 2.5%) |
| Termination: | generation 50 |

## 5 Mining Genes from Genetic Programming

The first group of 600 GP runs produced 600 best of run models. Together they contained 6970 of the 7129 DNA chip attributes. Some attributes were used much more than others. We selected the 404 genes which occurred in ten or more best of run individuals ([Moore *et al.*, 2002] used nine as a cut point in similar data). We re-ran the GP a second 600 times.

All the 404 genes were used by at least one of the 600 best of run individuals. However again the distribution of genes across models was highly non uniform with only two genes occurring in more than 100 models, see Figure 1. Genes U08998_at and U41737_at occurred in 182 and 193 best of run models. The first gene (TAR RNA binding protein (TRBP) mRNA, U08998_at) is known to promote the formation or development of cancerous tumours [Benkirane *et al.*, 1997] but [Pomeroy *et al.*, 2002, supplemental] says it is not amongst the top marker genes selected by signal-to-noise (mean) ratio (it is in the middle of the top 200). The second gene (Pancreatic beta cell growth factor (INGAP) mRNA, U41737_at) is not amongst [Pomeroy *et al.*, 2002, supplemental]'s top 200 but may be involved in the onset of diabetes [Rafaeloff *et al.*, 1997]. [Pomeroy *et al.*, 2002] heavily thresholded the data for these two genes (cf. Figure 2) which may explain why they were not identified. However their consistently low expression values do raise questions about data pre-processing and the practicality of using either gene in diagnostics tests.

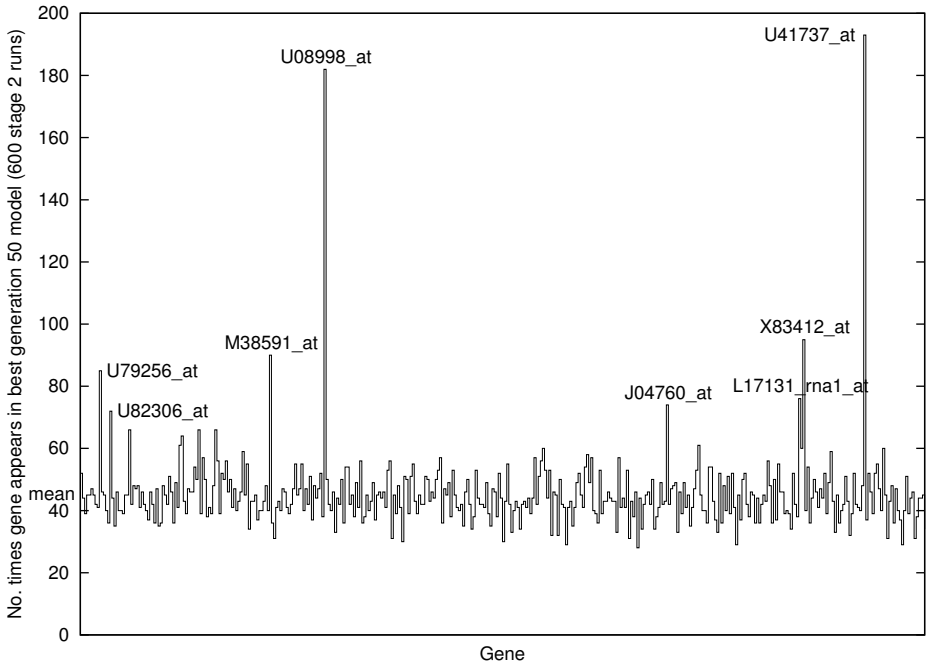


**Fig. 1.** Number of best of run GP models in 600 second stage GP runs which included each of the 404 genes selected by the first stage from the total of 7129 genes. Genes more than $4\sigma$ from mean are labelled.

We felt that applying the full power of genetic programming to noisy data was liable to generate overly complex models. Instead in a final 600 runs we severely limited the GP's expression and learning abilities. The model size and function set were both limited so models could only contain either no functions at all or a single IF statement and only the initial random generation was used.

The leave one out estimate of the accuracy of the random two gene models is 68%. [Pomeroy *et al.*, 2002] claims 47/60 (78% accuracy) with a k=5 nearest neighbour classifier using 8 genes. While their classifiers may be better, they are more complex. Also, with only 60 cases, we cannot show the difference between 68% and 78% is significant ($\sigma = 6\%$).

Three quarters of all cases (147 of 192) where the random models made incorrect predictions can be traced to 15 cases which are significantly ($p < 0.01$) harder than the overall leave one out estimate of accuracy (68%) would suggest. These are shown with solid markings in Figure 2. In 39 of the remaining 45 cases the random two gene models made the correct prediction more often than not.

Figure 2 shows the patient data. It is clear that the cases which prove hard on the leave one out two gene trials are hard because they lie within clusters of patients with the opposite outcome. Figure 2 also shows the decision boundaries of two typical random two gene models. Over the range of the data, one is linear

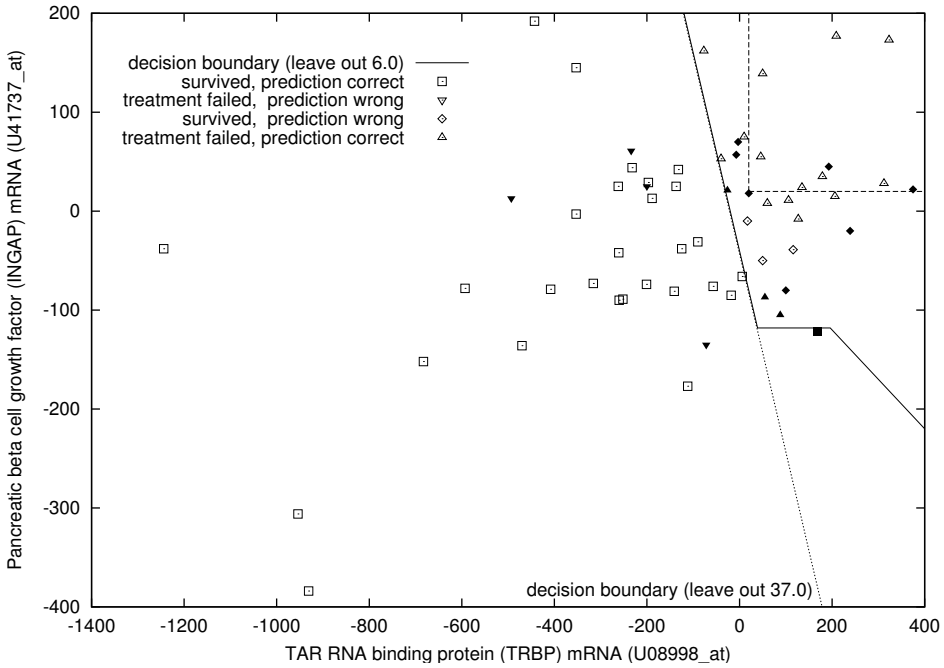


**Fig. 2.** Expression of two genes from 7129 used to predict outcome of cancer treatment [Pomeroy *et al.*, 2002]. The piecewise linear decision boundary is the best of 500 random programs using just these two genes (from the first of ten runs leaving out patient record 6). Solid shapes indicate cases which are significantly harder to predict. The linear boundary produced when leaving out a different record (37) is shown dotted. It predicts survival if $2 \times \text{U08998\_at} + \text{U41737\_at} < -43$ and makes only one more error. Dashed rectangle indicates threshold (20) used by [Pomeroy *et al.*, 2002], only nine patient records are not affected by such thresholding.

and the other is piecewise linear. While there many possible views of the data, the intermixing of survivors (squares) and non-survivors (triangles) in Figure 2 suggests that gene expression data cannot be 100% predictive and that there are other important factors.

## 6 Discussion

In general optimal selection of features (genes in our case) requires exponential effort, even for simple fixed interaction between features. Since optimal selection is not feasible, heuristics are used. Traditionally either forward or backward feature selection are used. They use a fixed interaction rule between features (typically linear) and sequentially process features one at a time. Either all features are included and the set is trimmed one feature at a time or no features are initialy included, then the most informative is added, followed by the next

(given those already selected) and so on. There is no scope for going back and re-considering features that have already been discarded (or selected).

Genetic programming is an alternative heuristic. Instead of a fixed combination rule, it allows almost any means of combining a number of features. (GP is also free to select the number of features.) At any one time, there isn't a single set of selected features, but instead a population of individuals using features. Evolution is free to add/remove multiple features (rather than one) and can re-consider previous selection/removal decisions as new combinations are tried.

We have rerun the experiments with different settings. Naturally in detail each run is very different. While, with ten replications of each validation fold, no significant difference in the two main genes was found, a different GP study might give different models (of similar performance).

It is clear from this, and similar, data sets that there are many ways to make predictions from non-linear combinations of subsets of genes whose accuracies are not significantly different. GP is a general powerful, noise tolerant, way of finding them, which can yield easily interpretable functions, rather than black boxes.

## 7 Conclusions

The enormous width of DNA gene chip data makes over fitting an ever present danger, particularly with powerful machine learning approaches. Genetic programming, in combination with leave one out cross validation and a principled objective function (which takes into account the class imbalance often found in Biological data sources) has been used to evolve many non-linear functions of gene expression values. The goal has been to whittle down the thousands of data attributes (gene expression measurements) into a few predictive ones.

We were surprised to find only one or two genes are needed to make predictions and by the simplicity of the models found by genetic programming. These results strongly suggest deterministic algorithms which only allow linear interaction but which are able to deal effectively with thousands of data attributes will also do well on this dataset. However the low data values given for the expression of the two genes raise questions about the pre-processing required when performing gene chip experiments. Other experiments (also on treatment outcome in animal and Human studies) confirm GP as a potentially valuable gene selection technique when extracting knowledge from DNA chip data.

### Acknowledgements

I would like to thank Matthew Trotter and David Corney.

### Source Code

C++ code can be obtained from `ftp://cs.ucl.ac.uk/genetic/gp-code/`

## References


Benkirane *et al.*, 1997. Monsef Benkirane, Christine Neuveut, Rene F. Chun, Stephen M. Smith, Charles E. Samuel, Anne Gatignol, and Kuan-Teh Jeang. Oncogenic potential of TAR RNA binding protein TRBP and its regulatory interaction with RNA-dependent protein kinase PKR. *EMBO Journal*, 16(3):611–624, 1997.

Carvalho and Freitas, 2003. Deborah R. Carvalho and Alex A. Freitas. A hybrid decision tree/genetic algorithm method for data mining. *Information Sciences Journal*, 2004. In press.

Gilbert *et al.*, 2000. Richard J. Gilbert, Jem J. Rowland, and Douglas B. Kell. Genomic computing: explanatory modelling for functional genomics. In Darrell Whitley *et al.,* editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 551–557, Las Vegas, Nevada, USA, 10-12 July 2000. Morgan Kaufmann.

Kohavi, 1995. Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of IJCAI*, pages 1137–1143. Morgan Kaufmann, 1995.

Langdon and Buxton, 2001a. William B. Langdon and Bernard F. Buxton. Genetic programming for combining classifiers. In Lee Spector *et al.,* editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 66–73, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.

Langdon and Buxton, 2001b. William B. Langdon and Bernard F. Buxton. Genetic programming for improved receiver operating characteristics. In Josef Kittler and Fabio Roli, editors, *Second International Conference on Multiple Classifier System*, volume 2096 of *LNCS*, pages 68–77, Cambridge, 2-4 July 2001. Springer Verlag.

Langdon and Buxton, 2001c. William B. Langdon and Bernard F. Buxton. Evolving receiver operating characteristics for data fusion. In Julian F. Miller *et al.,* editors, *Genetic Programming, Proceedings of EuroGP'2001*, volume 2038 of *LNCS*, pages 87–96, Lake Como, Italy, 18-20 April 2001. Springer-Verlag.

Langdon *et al.*, 2001. William B. Langdon, Steven J. Barrett, and Bernard F. Buxton. Genetic programming for combining neural networks for drug discovery. In Rajkumar Roy *et al.,* editors, *Soft Computing and Industry Recent Applications*, pages 597–608. Springer-Verlag, 10–24 September 2001. Published 2002.

Langdon, 1998. William B. Langdon. *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!*, volume 1 of *Genetic Programming*. Kluwer, Boston 1998.

Langdon, 2000. William B. Langdon. Size fair and homologous tree genetic programming crossovers. *Genetic Programming and Evolvable Machines*, 1(1/2):95–119, April 2000.

Li *et al.*, 2001. Leping Li, Clarice R. Weinberg, Thomas A. Darden, and Lee G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, 17(12):1131–1142, 2001.

Moore *et al.*, 2002. Jason H. Moore, Joel S. Parker, Nancy J. Olsen, and Thomas M. Aune. Symbolic discriminant analysis of microarray data in automimmune disease. *Genetic Epidemiology*, 23:57–69, 2002.

Pomeroy *et al.*, 2002. Scott L. Pomeroy, Pablo Tamayo, Michelle Gaasenbeek, Lisa M. Sturla, Michael Angelo, Margaret E. McLaughlin, John Y. H. Kim, Liliana C. Goumnerovak, Peter M. Black, Ching Lau, Jeffrey C. Allen, David Zagzag, James M. Olson,

Tom Curran, Cynthia Wetmore, Jaclyn A. Biegel, Tomaso Poggio, Shayan Mukherjee, Ryan Rifkin, Andrea Califanok, Gustavo Stolovitzkyk, David N. Louis, Jill P. Mesirov, Eric S. Lander, and Todd R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415:436–442, 24 January 2002.

Rafaeloff *et al.*, 1997. Ronit Rafaeloff, Gary L. Pittenger, Scott W. Barlow, Xiao F. Qin, Bing Yan, Lawrence Rosenberg, William P. Duguid, and Aaron I. Vinik. Cloning and sequencing of the pancreatic islet neogenesis associated protein (INGAP) gene and its expression in islet neogenesis in hamsters. *Journal of Clinical Investigations*, 99(N):2100–2109, May 1 1997.

Rodriguez-Vazquez and Oliver-Morales, 2003. Katya Rodriguez-Vazquez and Carlos Oliver-Morales. Divide and conquer: Genetic programming based on multiple branches encoding. In Conor Ryan *et al.,* editors, *Genetic Programming, Proceedings of EuroGP'2003*, volume 2610 of *LNCS*, pages 224–234, Essex, 14-16 April 2003. Springer-Verlag.

Soule, 1999. Terence Soule. Voting teams: A cooperative approach to non-typical problems using genetic programming. In Wolfgang Banzhaf *et al.,* editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 916–922, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.

Valafar, 2002. Faramarz Valafar. Pattern recognition techniques in microarray data analysis: A survey. *Annals of New York Academy of Sciences*, 980:41–64, December 2002. Special issue Techniques in Bioinformatics and Medical Informatics.