Genomic Computing. Explanatory Analysis of Plant Expression Profiling Data Using Machine Learning[1]

Douglas B. Kell*, Robert M. Darby, and John Draper

Institute of Biological Sciences, University of Wales, Aberystwyth SY23 3DD, United Kingdom

"Actually, the orgy of fact extraction in which everybody is currently engaged has, like most consumer economies, accumulated a vast debt. This is a debt of theory, and some of us are soon going to have an exciting time paying it back—with interest, I hope."
—Sydney Brenner, *In Theory*, 1997

As with every other organism whose genome has been sequenced (Hinton, 1997; Bork et al., 1998), a chief finding in plants (Bevan et al., 1999; Somerville and Somerville, 1999) is the presence of a vast number of genes (many with no relatives in the databases) whose existence, let alone function, had previously gone unrecorded. The importance of finding the function of these genes has led to what amounts to a complete reversal of conventional scientific strategies (Brent, 1999, 2000; Kell and Mendes, 2000), in which one would start with a phenotype (e.g. flower color) and devise experiments that would lead one to the genes whose products were responsible for producing that phenotype. Now, the dawn of the postgenomic era has (consequently) spawned major commercial and academic programs in which plants with more or less defined genotypes (e.g. knockouts; Martienssen, 1998) are being subjected to parallel and high-throughput analyses at the level of the transcriptome (Ruan et al., 1998; Schaffer et al., 2000; Schenk et al., 2000), the proteome (Santoni et al., 1998; Jacobs et al., 2000; Prime et al., 2000; van Wijk, 2000), the metabolome (Oliver et al., 1998; Trethewey et al., 1999; Fiehn et al., 2000; Johnson et al., 2000; Kell and Mendes, 2000; Raamsdonk et al., 2001; Trethewey, 2001), and the phenotype (Rieger et al., 1999), which will provide the wherewithal to assess the contribution of different genes through the activities of their products to the overall functioning of cells and organisms. The problem at hand is then how best to exploit the high-dimensional data floods so generated (e.g. with thousands of gene products or metabolites) for providing the comparatively low-dimensional explanations that we require at higher levels of organization (this gene is or is not important, for example, in cold tolerance).

[1] This work was supported by the UK Biotechnology and Biological Science Research Council.

* Corresponding author; e-mail dbk@aber.ac.uk; fax 44–0-1970–622354.

MULTIVARIATE DATA ANALYSIS AND MACHINE LEARNING

This highly multivariate data analysis problem is most easily thought of in relation to Figure 1A (Kell and King, 2000), which depicts a familiar view in the style of a spreadsheet or database table (Fig. 1A), in which the samples of interest are represented in different rows and a set of their properties in columns. Some of the columns might represent, for example, expression profiling data ("explanatory variables" or "*x*-data" in the jargon) that are going to be the inputs, whereas the functional or other classes of interest, which are still variables associated with the samples ("dependent variables" or "*y*-data"), are thus represented by a subset of the columns. The game then is to use the values of the input (*x*) variables to predict the appropriate classes of interest (the appropriate value of the *y* variable). Thus, Figure 1B equivalently depicts the problem as a set of multivariate inputs which may be transformed, via a set of mathematical transformations, into a series of outputs (possibly just one), such that application of the data vector on the left leads to the correct classification of the object from which the data were generated on the right side of the transformation.

In machine learning, it is normal to distinguish methods that use only the *x*-data (unsupervised methods) from supervised learning methods, which are trained using both the *x*-data and the *y*-data (Duda and Hart, 1973; Jain and Dubes, 1988; Therrien, 1989; Rich and Knight, 1991; Weiss and Kulikowski, 1991; Fukunaga, 1992; Michie et al., 1994; Bishop, 1995; Livingstone, 1995; Ripley, 1996; Mitchell, 1997). A vast pantheon of examples shows that supervised methods are always much more powerful than are unsupervised ones (such as the widely used Principal Components Analysis and clustering methods), because they concentrate on the variance that matters for the question of interest. In an example of our own concerning the exploitation of mass spectrometry in the assessment of the adulteration of extra virgin olive oils (Goodacre et al., 1992, 1993), most of the variance in the spectra was found to be due to the cultivar of olive, and not whether the oils were adulterated, such that unsupervised methods were fine for discriminating cultivars (see also Bianchi et al., 2001) but were useless for detecting adulteration. By contrast, a supervised method (in that case a fully interconnected backpropagation-type

A

		Variables going across in different columns					
		Explanatory (x-) Variables…….			Dependent (y-) Variable(s)		
		XVar1	Xvar2	Xvar3…	Yvar1	Yvar2…	Yvar3…
Objects	Obj1						
(Samples)	Obj2						
Going	Obj3						
Down	Obj4						
In	Obj5						
Different	Obj6						
Rows	Obj7						
:	:						

B

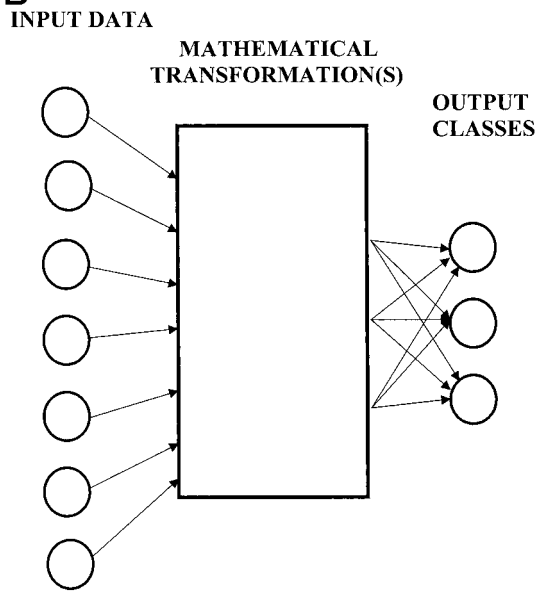

Figure 1. Supervised learning of a classical propositional system. A, The samples are set out to form the rows of a table, while relevant values (or category membership such as male/female) of their associated variables form the columns. Some of the variables are used to predict the values of other variables. B, In a different but equivalent representation, the explanatory variables appear on the left and the dependent variables on the right, and the aim is to produce a mathematical transformation that uses some or all of the inputs and classifies the object into the correct class on the right. Such a classification can also have a numerical value, such as the concentration of a metabolite that cannot be measured directly, the severity of a disease, and so on.

neural network (Wasserman, 1989; Hertz et al., 1991; Bishop, 1995; Ripley, 1996) trained on the same data succeeded in classifying all the oils in an unseen (double blind) test set of data (Goodacre et al., 1992; Goodacre et al., 1993).

The generalized problem so described parallels rather clearly the numerous DNA microarray experiments that have been performed and which are normally analyzed purely in terms of co-expression or clustering (Eisen et al., 1998; Tamayo et al., 1999; Burke, 2000; Getz et al., 2000), a strictly unsupervised method. Supervised methods are again much more appropriate here, though each has strengths and weaknesses (Brown et al., 1999; Alizadeh et al., 2000; Gilbert et al., 2000; Hastie et al., 2000), and using supervised learning methods to classify individual samples from microarray or other data in terms of a gene function or other type of class does require that one has a sensible class structure in the first place (Kell and King, 2000).

It should be noted that many general approaches and methods exist for supervised learning (e.g. neural, statistical, rule-based, symbolic, and so on; Rich and Knight, 1991; Weiss and Kulikowski, 1991; Hutchinson, 1994; Michie et al., 1994; Michalewicz and Fogel, 2000). However, all machine learning methods have their strengths and weaknesses, and there is provably no universal method that works best for all datasets (no free lunch; Radcliffe and Surry, 1995; Wolpert and Macready, 1997). Consequently, the method one may choose is governed by one's individual preferences regarding features such as speed, accuracy, mathematical rigor, and robustness, and by the comprehensibility of the model formed. Any claims for the superiority of one method over another consequently should be seen in this light. In our view (Kell and Sonnleitner, 1995; Davey and Kell, 1996; Alsberg et al., 1997; Goodacre et al., 2000), the best methods not only give one the correct answer, but give an explanation of what, in biological terms, is the basis for that answer. This depends on identifying a subset of the variables with high explanatory power.

Support vector machines (SVMs; Cortes and Vapnik, 1995; Scholkopf et al., 1997; Burges, 1998; Cristianini and Shawe-Taylor, 2000; Vapnik and Chapelle, 2000) are an approach that has generated much recent interest, most pertinently here regarding the analysis of expression profiling data (Brown et al., 1999, 2000). Like all approaches, SVMs too have their strengths and weaknesses. The strengths include the existence of formal theory (see above citations) and a rapid speed of convergence. Against this, they normally give equal weighting to every variable, so unless one has reasons to remove some of the variables, those that contribute only noise will tend to dominate if they are present in large numbers. This is certainly the case in microarray experiments (Wittes and Friedman, 1999), where, for example, an SVM failed to learn the helix-turn-helix class (Brown et al., 1999, 2000), whereas other methods that select only subsets of the variables succeeded on the same data (Gilbert et al., 2000; Delneri et al., 2001). Second, like neural nets (Mozer and Smolensky, 1989; Andrews et al., 1995; Tickle et al., 1998; Alexander and Mozer, 1999), such methods can have poor explanatory power (i.e. when you have trained the system and asked it to classify the test set it will do so,but it will not straightforwardly tell you which variables are used). However, as pointed out by a referee, hybrid methods could prove an interesting approach in which something like an evolutionary algorithm (see below) is used to select candidate subsets of variables

for processing through an SVM (compare with Guyon et al., 2001; Weston et al., 2001) or other learning method, as has been done using genetic algorithms for neural network (Broadhurst et al., 1997) and other statistical data processing analyses (Horchner and Kalivas, 1995).

A DATA FLOOD DEFENSE SYSTEM

Next, it should be noted that the problem of finding an adequate model (i.e. a set of mathematical transformations in the sense of Fig. 1B) scales only linearly with the number of objects but combinatorially with the number of variables. Thus, the number of models that either do or do not use a particular variable (let alone seek to parametrize it) when faced with a choice of M variables is obviously 2^M (a variable is used or not used, binary 1 or 0). Even for $M = 100$, which is much smaller than the numbers typically used in microarrays, $2^{100} \cong 10^{30}$, and the lifetime of the universe is only approximately 10^{17} s. Exhaustive search of such models to find the best model is clearly computationally intractable. By contrast, if we state that a model should use just two, three, four, or five variables out of the 100, the numbers of combinations are, respectively, just 4,950, 161,700, 3,921,225, and 75,287,520. These kinds of numbers are both (a) much more tractable and (b) likely to lead to much more intelligible explanations of which variables (genes/proteins/metabolites) are most important to a particular process (stress, flowering time, and so on) of interest. The rather Zen conclusion is clearly that we are wise to start by seeking simplicity in our explanations.

EVOLVING SIMPLE ANSWERS TO COMPLEX QUESTIONS OF FUNCTIONAL GENOMICS

A particularly useful approach to attacking combinatorial optimization problems of this type lies in the use of the methods of evolutionary computing. In evolutionary computing (Goldberg, 1989; Michalewicz, 1994; Mitchell, 1995; Bäck et al., 1997, 2000a, 2000b; Corne et al., 1999; Zitzler, 1999; Michalewicz and Fogel, 2000), we have a population of individual computer programs or algorithms whose output is a potential solution to a problem (typically a combinatorial optimization problem). These outputs are ranked according to their "fitness" (usually their closeness to the true solution in the dataset they are given, though other criteria such as simplicity of explanation may be used as well), and the better-performing individuals retained. Some of these individuals/programs/algorithms are then modified, typically by mutating (changing) them "asexually" or by recombining parts of them from more than one parent "sexually," and the process of generating an output, evaluating the fitness function, and mutating and selecting at each generation continued until a specified stopping criterion is met. (This is usually the number of generations or the achievement of an adequately small difference between desired and true values.)

This may be set out in pseudocode as follows:

1. START by generating a population of individuals (computer programs/algorithms)
2. At the end of each generation, EVALUATE the "fitness" of each individual
3. RANK these individuals and RETAIN a certain fraction of them with a probability related to their fitness for one or more purposes
4. CREATE NEW INDIVIDUALS from these parents so as to replenish the population by mutation (changing an individual parent) or recombination (between two or more parents)
5. Return to step 2 UNTIL. . .
6. . . .when a suitable criterion (elapsed time, evolved fitness, generation number) is met, then STOP.

A particularly interesting subset of evolutionary computing methods, popularized by John Koza as genetic programming (GP; Koza, 1992, 1994; Banzhaf et al., 1998; Langdon, 1998; Koza et al., 1999, 2000), involves an arrangement in which the rules are arrayed in a tree-like structure that is read from the bottom and a subset of variables passed through appropriate operators or functions to provide the output (i.e. fitness). Such so-called parse trees—unlike conventional computer programs—can be mutated and recombined to provide variants that remain syntactically correct (Fig. 2). Thus, one can evolve solutions to a complex problem yet produce equations that are simple and intelligible. These equations are essentially in the form of rules, in that the best ones give entirely different outputs for the different inputs characteristic of examples from particular classes, and if the classes are encoded as the outputs (say in binary, 1 or 0 for contribution of a gene to a particular function), the equations are the rules. (An equivalent procedure can be used, for example, in spectroscopy for high-throughput screening, where the output is continuous and is the concentration of a substance of interest [Gilbert et al., 1997; Taylor et al., 1998b; Woodward et al., 1999].) The special power of GP, which we have found particularly valuable (Gilbert et al., 1997, 1998, 1999; Jones et al., 1998; Taylor et al., 1998a, 1998b; Goodacre and Gilbert, 1999; Woodward et al., 1999; Goodacre et al., 2000; Johnson et al., 2000), stems from the fact that both the (potentially small number of) explanatory variables and the functional form of the relationship between them are evolved together. As suggested by a referee, and to avoid confusion, we note that genetic programming differs significantly from methods such as genetic algorithms. In genetic algorithms, the length of the string is normally fixed, and what evolves is a parameterization of a given model. In GP, the model itself evolves too (and while this is normally effected in a

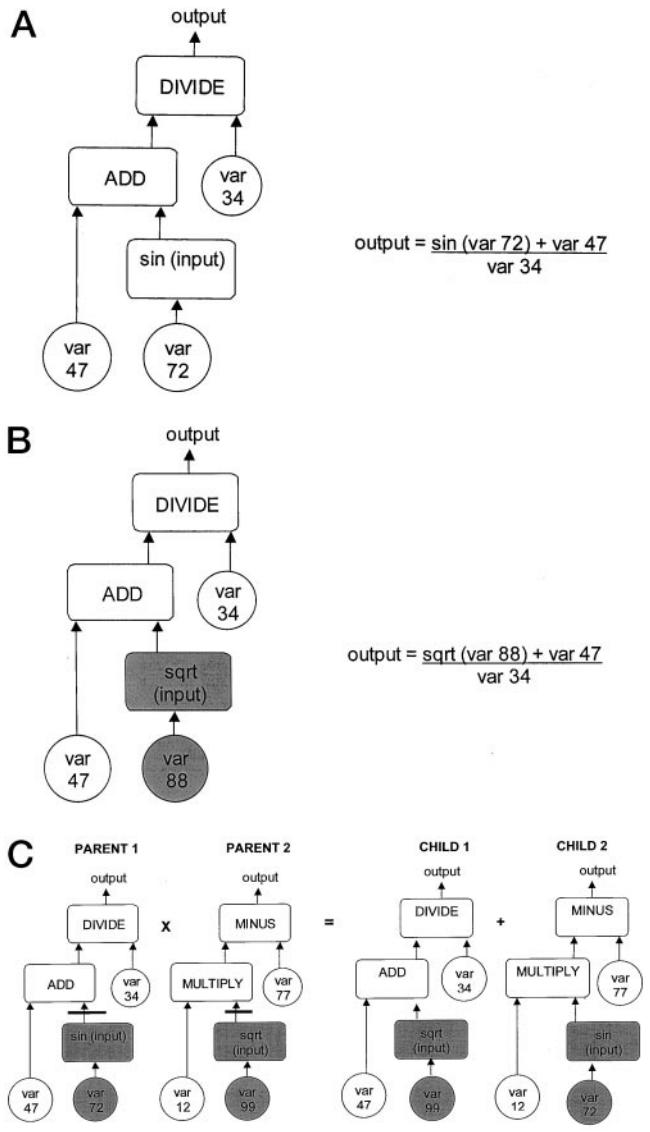

Figure 2. A, Parse-tree representation of an equation which takes some of the variables as the input and, by reading from the lowest leaves of the tree, produces an output at the top, which is clearly equivalent to the equation given on the right. Also shown are cartoons of how mutation (B) and recombination (C) can be effected to produce equations (rules) with different properties while preserving a syntactically logical structure.

tree structure, as shown in Fig. 2, linear versions of GP are also possible; Banzhaf et al., 1998; Gilbert et al., 2000).

We also mention here that a significant problem, especially with some of the original flavors of GP, can be their tendency to bloat, i.e. to continue to grow branches onto trees even when the contribution of these branches to fitness is small (Langdon and Poli, 1998), a phenomenon mirroring what may be observed in nature where it is referred to as "Muller's ratchet" (Muller, 1964). There are ways around this however, e.g. by incorporating the desirability for small trees into the fitness function itself (Goodacre et al., 2000), and we do not nowadays find this a problem. Another feature of GPs is that because of the stochastic way in which they are initiated and evolve, they are not deterministic. This said, it is possible to turn such properties to advantage by running the GP several times, as there is plenty of evidence that combining or voting among several independent solutions to a problem can give improved learning (Drucker et al., 1994; Bauer and Kohavi, 1999; Dietterich, 2000a, 2000b; Friedman et al., 2000; King et al., 2000). Finally, all computer-intensive methods of this type, including those based on purely multivariate statistical strategies (Martens and Næs, 1989), have a great many degrees of freedom, which require that we provide a careful evaluation with respect to the reality of the solutions found (Chatfield, 1995).

In our own approach, which we refer to as genomic computing, we choose to equate the fitness of an individual, not by a regression-based analysis such as the root mean square error of prediction or the percentage of samples classified correctly, but by using a ranking scheme in which the metric encoding the quality of the model is analyzed in terms of the ordering of the samples with respect to their ease of prediction via the model. We find that this is particularly good at drawing out the variables that are most important for the particular problem.

AN EXAMPLE: ANALYSIS OF THE TOBACCO METABOLOME IN TRANSGENIC PLANTS

As an illustrative example, we set up a transgene discovery problem in which we measured a series of metabolites via HPLC and used these as the inputs to a genetic program designed to find a rule that would tell from the metabolome data whether the transgene of interest was present or absent (of course, we could have sought to encode its activity). The experiment was also aimed at investigating the biosynthesis and function of salicylic acid in plant defense by the expression of a salicylate hydroxylase enzyme to block accumulation (Bi et al., 1995).

Salicylic acid has been known for more than a decade to play a key role in defense mechanisms in many plants and is associated specifically with the hypersensitive response and the phenomenon of systemic acquired resistance (Mur et al., 1996, 2000; Draper, 1997; Mur et al., 1997). Tobacco (*Nicotiana tabaccum*) has provided a model organism for the study of salicylate biology in plant defense, but despite a considerable amount of research, little is known regarding its synthesis, catabolism, and mode of action. A bacterial gene encoding the enzyme salicylate hydroxylase (SH-L) expressed from the cauliflower mosaic virus 35S promoter has provided a useful tool to block salicylic acid accumulation in transgenic tobacco (Bi et al., 1995; Mur et al., 1996, 1997; Darby et al., 2000). Six-week-old transgenic

tobacco plants (35S-SH-L) and control plants (Samsum NN) were inoculated with tobacco mosaic virus (TMV) at a temperature (32°C) non-permissive for the hypersensitive response (Mur et al., 1997; Roberts et al., 1999). Under these conditions, the TMV can replicate and spread without inducing lesion formation. Following a shift to a permissive temperature (24°C), hypersensitive response is induced synchronously with cell death visible after 8 h. Leaf tissue from TMV-inoculated, temperature-shifted plants was sampled at different time points (0–24 h), flash frozen in liquid N_2, extracted in 90% (v/v) methanol, dried, partitioned with dichloromethane, and then analyzed by HPLC using standard procedures (Bi et al., 1995). A total of 48 peaks from the HPLC traces were digitized and integrated using standard software provided with the instrument, and a total of 36 samples were studied.

The metabolite peak values were used as inputs to the genomic computing software Gmax-bio (Aber Genomic Computing, Aberystwyth, UK), with the presence or absence of SH-L in the genotype being encoded 1 or 0.

One of many rules which evolved could be written as follows:

```
x1 = V24
     x2 = V37
   If x1 <> 0 Then x1 = x2/x1 Else x1 = 1
   x1 = Sqr(Abs(x1))
     x2 = V24
        x3 = V42
     x2 = x2 + x3
        x3 = V30
     If x2 <> 0 Then x2 = x3/x2 Else x2 = 1
     x2 = Sqr(Abs(x2))
   x1 = x1 + x2
   SCORE = x1
PRBBLITY = 1/(1 + Exp[−{−8.046777 + SCORE *
1.872833}])
```

This rule has an accuracy of more than 95% and is shown in tree form in Figure 3A. A power of genomic computing is that it ranks variables in order of their utility in successful rules. The top three variables are peaks 24, 30, and 42, and peak 24 is indeed salicylate. The low intrinsic dimensionality of this problem thus allows us to visualize the salient features of the experiment in a straightforward way, as illustrated in Figure 3B.

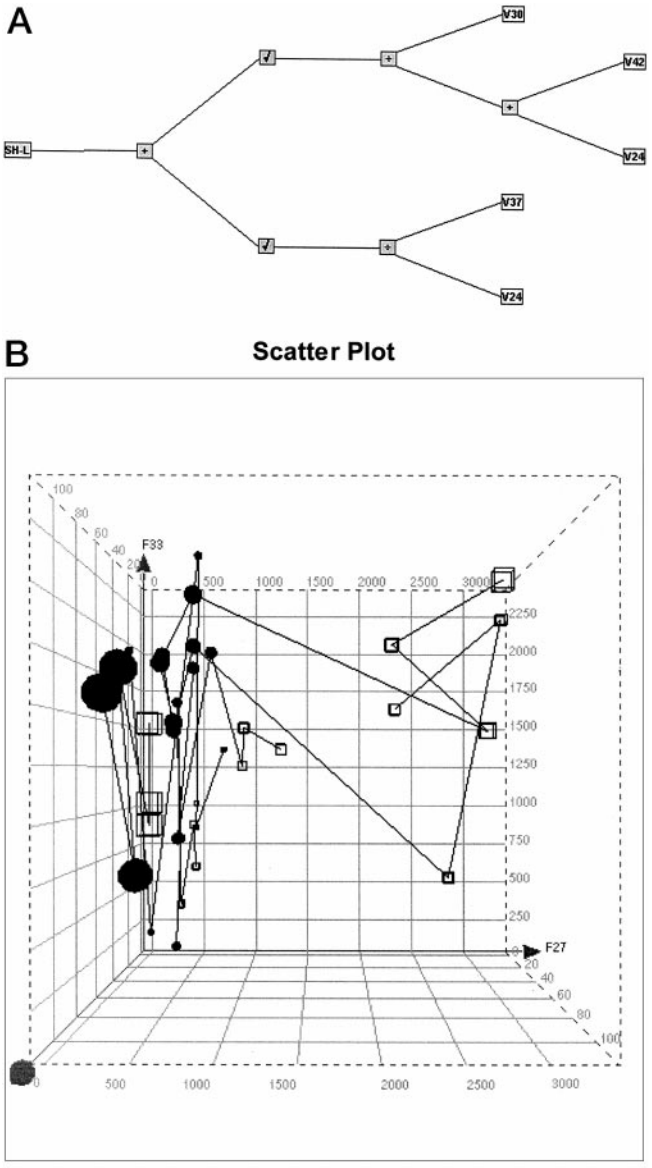

Figure 3. A rule derived from genomic computing for the presence of a specific transgene (salicylate hydroxylase) in tobacco. A, Tree illustrating the rule. B, Plot of the data using the three variables identified as most significant. Presence of SH-L is encoded by the symbol (□, none; ●, present). Time after shift (0, 3, 6, 9, 12, and 24 h) is encoded by size: F27 (abscissa) = variable 24, F33 (ordinate) = variable 30, and F45 (coming out of page) is variable 42. The latest time points have low values for variable 24 but large values for variable 23 (data not shown). Note that the problem is apparently not linearly separable. The Gmax-bio software (Aber Genomic Computing) was used with the following default parameters: population size, 1,000; maximum program length, 44 nodes; fitness, based on tournament selection/Gmax(v); crossover operator used 80% of the time; and of the mutations, terminals were selected 20% of the time. Operators used were the default numeric (0.1, 1, 3, 5, and Rand) and arithmetic (+, −, /, and *) operators plus square root, log, tanh, <, AND, OR, or IfT (this latter operator takes three inputs and if the first ≠ 0 then it returns the second, else the third). The derivation of this rule required 56 generations and approximately 2 min on a standard 750-MHz Pentium III (Intel) personal computer.

CONCLUSION

Many modern "omics" technologies are producing highly multivariate data at unprecedented rates. Only modern machine learning methods can turn these data into knowledge. Genomic computing provides an approach that can effect this desirable transformation and provide simple rules that map back onto the variables measured in the real world and thus have high explanatory power.

ACKNOWLEDGMENT

We thank Mike Sinclair for useful discussions.

Received April 13, 2001; accepted May 1, 2001.

LITERATURE CITED

Alexander JA, Mozer MC (1999) Template-based procedures for neural network interpretation. Neural Netw **12:** 479–498

Alizadeh AA, Eisen MB, Davis RE, Ma, C, Lossos IS, Rosenwald A, Boldrick JG, Sabet H, Tran T, Yu, X et al. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature **403:** 503–511

Alsberg BK, Goodacre R, Rowland JJ, Kell DB (1997) Classification of pyrolysis mass spectra by fuzzy multivariate rule induction-comparison with regression, K-nearest neighbour, neural and decision-tree methods. Anal Chim Acta **348:** 389–407

Andrews R, Diederich J, Tickle AB (1995) Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Syst **8:** 373–389

Bäck T, Fogel DB, Michalewicz Z (1997) Handbook of Evolutionary Computation. IOP Publishing/Oxford University Press, Oxford

Bäck T, Fogel DB, Michalewicz Z (2000a) Evolutionary Computation 1: Basic Algorithms and Operators. IOP Publishing, Bristol, UK

Bäck T, Fogel DB, Michalewicz Z (2000b) Evolutionary Computation 2: Advanced Algorithms and Operators. IOP Publishing, Bristol, UK

Banzhaf W, Nordin P, Keller RE, Francone FD (1998) Genetic Programming: An Introduction. Morgan Kaufmann, San Francisco

Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Mach Learn **36:** 105–139

Bevan M, Bancroft I, Mewes HW, Martienssen R, McCombie R (1999) Clearing a path through the jungle: progress in *Arabidopsis* genomics. Bioessays **21:** 110–120

Bi YM, Kenton P, Mur L, Darby R, Draper J (1995) Hydrogen peroxide does not function downstream of salicylic acid in the induction of PR protein expression. Plant J **8:** 235–245

Bianchi G, Giansante L, Shaw A, Kell DB (2001) Chemometric criteria for the characterization of Italian protected denomination of origin (DOP) olive oils from their metabolic profiles. Eur J Lipid Sci Technol **103:** 141–150

Bishop CM (1995) Neural Networks for Pattern Recognition. Clarendon Press, Oxford

Bork P, Dandekar T, Diaz-Lazcoz Y, Eisenhaber F, Huynen M, Yuan YP (1998) Predicting function: from genes to genomes and back. J Mol Biol **283:** 707–725

Brenner S (1997) In theory. *In* Loose Ends. Current Biology, London, p 37

Brent R (1999) Functional genomics: learning to think about gene expression data. Curr Biol **9:** R338–R341

Brent R (2000) Genomic biology. Cell **100:** 169–183

Broadhurst D, Goodacre R, Jones A, Rowland JJ, Kell DB (1997) Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry. Anal Chim Acta **348:** 71–86

Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet C, Furey TS, Manuel Ares J, Haussler D (1999) Support vector machine classification of microarray gene expression data. Technical Report No. UCSC-CRL-99–09. University of Santa Cruz, Santa Cruz, CA. http://www.cse.ucsc.edu/research/compbio/genex/genex.ps

Brown MPS, Grundy WN, Lin D, Cristianini N, Sugnet C, Furey TS, Manuel Ares J, Haussler D (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc Natl Acad Sci USA **97:** 262–267

Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Mining Knowledge Discovery **2:** 121–167

Burke HB (2000) Discovering patterns in microarray data. Mol Diagn **5:** 349–357

Chatfield C (1995) Model uncertainty, data mining and statistical inference. J R Stat Soc Ser A **158:** 419–466

Corne D, Dorigo M, Glover F (1999) New Ideas in Optimization. McGraw Hill, London

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn **20:** 273–297

Cristianini N, Shawe-Taylor J (2000) An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge, UK

Darby RM, Maddison A, Mur LAJ, Bi YM, Draper J (2000) Cell-specific expression of salicylate dydroxylase in an attempt to separate localised HR and systemic SAR in tobacco. Plant Mol Pathol **1:** 115–124

Davey HM, Kell DB (1996) Flow cytometry and cell sorting of heterogeneous microbial populations: the importance of single-cell analysis. Microbiol Rev **60:** 641–696

Delneri D, Brancia FL, Oliver SG (2001) Towards a truly integrative biology through the functional genomics of yeast. Curr Opin Biotechnol **12:** 87–91

Dietterich TG (2000a) Ensemble methods in machine learning. *In* Multiple Classifier Systems, Lecture Notes in Computer Science Vol 1857. Springer, Berlin, pp 1–15

Dietterich TG (2000b) An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn **40:** 139–157

Draper J (1997) Salicylate, superoxide synthesis and cell suicide in plant defense. Trends Plant Sci **2:** 162–165

Drucker H, Cortes C, Jackel LD, Lecun Y, Vapnik V (1994) Boosting and other ensemble methods. Neural Comput **6:** 1289–1301

Duda RO, Hart PE (1973) Pattern Classification and Scene Analysis. John Wiley, London

Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA **95:** 14863–14868

Fiehn O, Kopka J, Dormann P, Altmann T, Trethewey RN, Willmitzer L (2000) Metabolite profiling for plant functional genomics. Nat Biotechnol **18:** 1157–1161

Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat **28:** 337–374

Fukunaga K (1992) Introduction to Statistical Pattern Recognition, Ed 2. Academic Press, New York

Getz G, Levine E, Domany E (2000) Coupled two-way clustering analysis of gene microarray data. Proc Natl Acad Sci USA **97:** 12079–12084

Gilbert RJ, Goodacre R, Shann B, Taylor J, Rowland JJ, Kell DB (1998) Genetic programing-based variable selection for high-dimensional data. *In* JR Koza, W Banzhaf, K Chellapilla, K Deb, M Dorigo, DB Fogel, MH Garzon, DE Goldberg, H Iba RL Riolo, eds, Genetic Programming 1998: Proceedings of the Third Annual Conference. Morgan Kaufmann, San Francisco, pp 109–115

Gilbert RJ, Goodacre R, Woodward AM, Kell DB (1997) Genetic programming: a novel method for the quantitative analysis of pyrolysis mass spectral data. Anal Chem **69:** 4381–4389

Gilbert RJ, Johnson HE, Winson MK, Rowland JJ, Goodacre R, Smith AR, Hall MA, Kell DB (1999) Genetic programming as an analytical tool for metabolome data. *In* WB Langdon, R Poli, P Nodin, T Fogarty, eds, Late-Breaking Papers of EuroGP-99. CWI, Amsterdam, pp 23–33

Gilbert RJ, Rowland JJ, Kell DB (2000) Genomic computing: explanatory modelling for functional genomics. *In* D Whitley, D Goldberg, E Cantú-Paz, L Spector, I Parmee, H-G Beyer, eds, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000). Morgan Kaufmann, Las Vegas, pp 551–557

Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA

Goodacre R, Gilbert RJ (1999) The detection of caffeine in a variety of beverages using Curie-point pyrolysis mass spectrometry and genetic programming. Analyst **124:** 1069–1074

Goodacre R, Kell DB, Bianchi G (1992) Neural networks and olive oil. Nature **359:** 594

Goodacre R, Kell DB, Bianchi G (1993) Rapid assessment of the adulteration of virgin olive oils by other seed oils using pyrolysis mass spectrometry and artificial neural networks. J Sci Food Agric **63:** 297–307

Goodacre R, Shann B, Gilbert RJ, Timmins EM, McGovern AC, Alsberg BK, Kell DB, Logan NA (2000) Detection of the dipicolinic acid biomarker in Bacillus spores using Curie-point pyrolysis mass spectrometry and Fourier transform infrared spectroscopy. Anal Chem **72:** 119–127

Guyon I, Weston J, Barnhill S, Vapnik V (2001) Gene selection for cancer classification using support vector machines. Mach Learn (in press)

Hastie T, Tibshirani R, Eisen MB, Alizadeh A, Levy R, Staudt L, Chan WC, Botstein D, Brown P (2000) "Gene shaving" as a method for identifying distinct sets of genes with similar expression patterns. Genome Biol **1:** research 0003.1–0003.21

Hertz J, Krogh A, Palmer RG (1991) Introduction to the theory of neural computation. Addison-Wesley, Redwood City, CA

Hinton JCD (1997) The *Escherichia coli* genome sequence: the end of an era or the start of the FUN? Mol Microbiol **26:** 417–422

Horchner U, Kalivas JH (1995) Further investigation on a comparative study of simulated annealing and genetic algorithm for wavelength selection. Anal Chim Acta **311:** 1–13

Hutchinson A (1994) Algorithmic Learning. Clarendon Press, Oxford

Jacobs DI, van der Heijden R, Verpoorte R (2000) Proteomics in plant biotechnology and secondary metabolism research. Phytochem Anal **11:** 277–287

Jain AK, Dubes RC (1988) Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, NJ

Johnson HE, Gilbert RJ, Winson MK, Goodacre R, Smith AR, Rowland JJ, Hall MA, Kell DB (2000) Explanatory analysis of the metabolome using genetic programming of simple, interpretable rules. Genet Progr Evolvable Mach **1:** 243–258

Jones A, Young D, Taylor J, Kell DB, Rowland JJ (1998) Quantification of microbial productivity via multi-angle light scattering and supervised learning. Biotechnol Bioeng **59:** 131–143

Kell DB, King RD (2000) On the optimization of classes for the assignment of unidentified reading frames in functional genomics programmes: the need for machine learning. Trends Biotechnol **18:** 93–98

Kell DB, Mendes P (2000) Snapshots of systems: metabolic control analysis and biotechnology in the post-genomic era. *In* A Cornish-Bowden, ML Cárdenas, eds, Technological and Medical Implications of Metabolic Control Analysis. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp 3–25

Kell DB, Sonnleitner B (1995) GMP—good modelling practice: an essential component of good manufacturing practice. Trends Biotechnol **13:** 481–492

King RD, Ouali M, Strong AT, Aly A, Elmaghraby A, Kantardzic M, Page D (2000) Is it better to combine predictions? Protein Eng **13:** 15–19

Koza JR (1992) Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA

Koza JR (1994) Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press, Cambridge, MA

Koza JR, Bennett FH, Keane MA, Andre D (1999) Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann, San Francisco

Koza JR, Keane MA, Yu, J, Bennett FH III, Mydlowec W (2000) Automatic creation of human-competitive programs and controllers by means of genetic programming. Genet Progr Evolvable Mach **1:** 121–164

Langdon WB (1998) Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming! Kluwer, Boston

Langdon WB, Poli R (1998) Fitness causes bloat: mutation. *In* W Banzhaf, R Poli, M Schoenauer, TC Fogarty, eds, Proceedings of the First European Workshop on Genetic Programming, Vol 1391. Springer-Verlag, Berlin, pp 37–48

Livingstone D (1995) Data Analysis for Chemists. Oxford University Press, Oxford

Martens H, Næs T (1989) Multivariate Calibration. John Wiley, Chichester, UK

Martienssen RA (1998) Functional genomics: probing plant gene function and expression with transposons. Proc Natl Acad Sci USA **95:** 2021–2026

Michalewicz Z (1994) Genetic Algorithms + Data Structures = Evolution Programs, Ed 3. Springer-Verlag, Berlin

Michalewicz Z, Fogel DB (2000) How to Solve It: Modern Heuristics. Springer-Verlag, Heidelberg

Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine Learning: Neural and Statistical Classification, Ellis Horwood Series in Artificial Intelligence. Ellis Horwood, Chichester, UK

Mitchell M (1995) An Introduction to Genetic Algorithms. MIT Press, Boston

Mitchell TM (1997) Machine Learning. McGraw Hill, New York

Mozer MC, Smolensky P (1989) Using relevance to reduce network size automatically. Connect Sci **1:** 3–16

Muller HJ (1964) The relation of recombination to mutational advance. Mutat Res **1:** 2–9

Mur LAJ, Bi YM, Darby RM, Firek S, Draper J (1997) Compromising early salicylic acid accumulation delays the hypersensitive response and increases viral dispersal during lesion establishment in TMV-infected tobacco. Plant J **12:** 1113–1126

Mur LAJ, Brown IR, Darby RM, Bestwick CS, Bi YM, Mansfield JW, Draper J (2000) A loss of resistance to avirulent bacterial pathogens in tobacco is associated with the attenuation of a salicylic acid-potentiated oxidative burst. Plant J **23:** 609–621

Mur LAJ, Naylor G, Warner SAJ, Sugars JM, White RF, Draper J (1996) Salicylic acid potentiates defense gene expression in tissue exhibiting acquired resistance to pathogen attack. Plant J **9:** 559–571

Oliver SG, Winson MK, Kell DB, Baganz F (1998) Systematic functional analysis of the yeast genome. Trends Biotechnol **16:** 373–378

Prime TA, Sherrier DJ, Mahon P, Packman LC, Dupree P (2000) A proteomic analysis of organelles from *Arabidopsis thaliana*. Electrophoresis **21:** 3488–3499

Raamsdonk LM, Teusink B, Broadhurst D, Zhang N, Hayes A, Walsh M, Berden JA, Brindle KM, Kell DB, Rowland JJ et al. (2001) A functional genomics strategy that uses metabolome data to reveal the phenotype of silent mutations. Nat Biotechnol **19:** 45–50

Radcliffe NJ, Surry PD (1995) Fundamental limitations on search algorithms: evolutionary computing in perspective. Comput Sci Today **1995:** 275–291

Rich E, Knight K (1991) Artificial Intelligence. McGraw Hill, New York

Rieger K-J, Orlowska G, Kaniak A, Coppee J-Y, Aljinovic G, Slonimski PP (1999) Large-scale phenotypic analysis in microtitre plates of mutants with deleted open reading frames from yeast chromosome III: key step between genomic sequencing and protein function. *In* AG Craig, JD Joheisel, eds, Methods in Microbiology. Automation: Genomic and Functional Analysis, Vol 28. Academic Press, London, pp 205–227

Ripley BD (1996) Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge, UK

Roberts MR, Warner SAJ, Darby R, Lim EK, Draper J, Bowles DJ (1999) Differential regulation of a glucosyl transferase gene homologue during defense responses in tobacco. J Exp Bot **50:** 407–410

Ruan Y, Gilmore J, Conner T (1998) Towards *Arabidopsis* genome analysis: monitoring expression profiles of 1400 genes using cDNA microarrays. Plant J **15:** 821–833

Santoni V, Rouquie D, Doumas P, Mansion M, Boutry M, Degand H, Dupree P, Packman L, Sherrier J, Prime T et al. (1998) Use of a proteome strategy for tagging proteins present at the plasma membrane. Plant J **16:** 633–641

Schaffer R, Landgraf J, Perez-Amador M, Wisman E (2000) Monitoring genome-wide expression in plants. Curr Opin Biotechnol **11:** 162–167

Schenk PM, Kazan K, Wilson I, Anderson JP, Richmond T, Somerville SC, Manners JM (2000) Coordinated plant defense responses in Arabidopsis revealed by microarray analysis. Proc Natl Acad Sci **97:** 11655–11660

Scholkopf B, Sung KK, Burges CJC, Girosi F, Niyogi P, Poggio T, Vapnik V (1997) Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Trans Sign Process **45:** 2758–2765

Somerville C, Somerville S (1999) Plant functional genomics. Science **285:** 380–383

Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander ES, Golub TR (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc Natl Acad Sci USA **96:** 2907–2912

Taylor J, Goodacre R, Wade WG, Rowland JJ, Kell DB (1998a) The deconvolution of pyrolysis mass spectra using genetic programming: application to the identification of some Eubacterium species. FEMS Microbiol Lett **160:** 237–246

Taylor J, Rowland JJ, Goodacre R, Gilbert RJ, Winson MK, Kell DB (1998b) Genetic programming in the interpretation of Fourier transform infrared spectra: quantification of metabolites of pharmaceutical importance. *In* JR Koza, W Banzhaf, K Chellapilla, K Deb, M Dorigo, DB Fogel, MH Garzon, DE Goldberg, H Iba, RL Riolo, eds, Genetic Programming 1998: Proceedings of the Third Annual Conference. Morgan Kaufmann, San Francisco, pp 377–380

Therrien CW (1989) Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics. Wiley, New York

Tickle AB, Andrews R, Golea M, Diederich J (1998) The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. IEEE Trans Neural Netw **9:** 1057–1068

Trethewey RN (2001) Gene discovery via metabolic profiling. Curr Opin Biotechnol **12:** 135–138

Trethewey RN, Krotzky AJ, Willmitzer L (1999) Metabolic profiling: a Rosetta Stone for genomics? Curr Opin Plant Biol **2:** 83–85

van Wijk KJ (2000) Proteomics of the chloroplast: experimentation and prediction. Trends Plant Sci **5:** 420–425

Vapnik V, Chapelle O (2000) Bounds on error expectation for support vector machines. Neural Comput **12:** 2013–2036

Wasserman PD (1989) Neural Computing: Theory and Practice. Van Nostrand Reinhold, New York

Weiss SH, Kulikowski CA (1991) Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning, and Expert Systems. Morgan Kaufmann Publishers, San Mateo, CA

Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2001) Feature selection for SVMs. *In* Neural Information Processing Systems, Vol 13. Morgan Kaufmann, San Francisco (in press)

Wittes J, Friedman HP (1999) Searching for evidence of altered gene expression: a comment on statistical analysis of microarray data. J Natl Cancer Inst **91:** 400–401

Wolpert DH, Macready WG (1997) No Free Lunch theorems for optimization. IEEE Trans Evol Comput **1:** 67–82

Woodward AM, Gilbert RJ, Kell DB (1999) Genetic programming as an analytical tool for non-linear dielectric spectroscopy. Bioelectrochem Bioenerg **48:** 389–396

Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: methods and applications. Shaker Verlag, Aachen, Germany