Genotype–phenotype mapping: genes as computer programs

Douglas B. Kell

The effects of genes on phenotype are mediated by processes that are typically unknown but whose determination is desirable. The conversion from gene to phenotype is not a simple function of individual genes, but involves the complex interactions of many genes; it is what is known as a nonlinear mapping problem. A computational method called genetic programming allows the representation of candidate nonlinear mappings in several possible trees. To find the best model, the trees are 'evolved' by processes akin to mutation and recombination, and the trees that more closely represent the actual data are preferentially selected. The result is an improved tree of rules that represent the nonlinear mapping directly. In this way, the encoding of cellular and higher-order activities by genes is seen as directly analogous to computer programs. This analogy is of utility in biological genetics and in problems of genotype–phenotype mapping.

Published online: 23 September 2002

One result of genome-sequencing programmes is the discovery of many genes with unknown functions. There is also a recognition that rather than testing specific hypotheses by experiment, the large quantities of expression profiling data now being generated [1–5] can be used to generate explanations that become the new hypothesis [6] in a continuing cycle of hypothesis generation and testing (Fig. 1).

Machine learning of complex networks

Consider genetic or metabolic networks. Given the PARAMETERS (see Glossary) (e.g. the nature of interactions, feedback loops, etc.) and rate equations of a kinetic model of a metabolic or genetic network, it is possible to 'run' the model inside a computer (by solving the appropriate differential equations) and determine the time evolution of the metabolic VARIABLES, which include the fluxes and concentrations of metabolites and other catalytic and signalling molecules (e.g. [7–11]). However, the variables are determined by the parameters, not vice versa, and what we often wish to do is to solve the 'inverse problem', in which we measure the variables (such as the levels of metabolites or gene products) and derive the parameters from them [12,13]. This problem is known as predictive modelling or structural equation modelling [14–16], and it shares the goals of machine learning [17,18], in which the aim is to find rules that effect a NONLINEAR MAPPING between inputs and outputs in complex systems (Fig. 2). The majority of current analyses are of course focused on microarray data.

Initially, only clustering methods were used to analyse these complex data sets [19–21]. These and related methods [22–24] are referred to as 'unsupervised' learning methods, and they use only knowledge of the 'input' (microarray) data to represent the 'closeness' (e.g. patterns of relative increase or decrease or temporal co-expression) of a high-dimensional array vector (i.e. a list of numbers representing the expression level of a great many genes) to another such list in some low-dimensional space (i.e. a smaller number) which may be visualized (e.g. by the extraction of principal components [25]). This assumes that things that are nearer to each other in this low-dimensional space are more 'like' each other, a strategy often known as 'guilt by association' [26,27].

However, problems based on input–output mapping of the type shown in Fig. 2 are best analysed using 'supervised' methods. With these – in contrast to the clustering methods – we include knowledge of the output or 'class membership' in the analysis and 'train' the system when presented with the input vector to

Glossary

Genetic programming (GP): A technique in which we evolve computer programs to solve specific problems, typically those cast as a nonlinear mapping problem as in Fig. 2. Although the principles go back further [a], it was popularised by John Koza of Stanford University in a series of books [b–d]. A web-accessible introduction can be found at http://www.geneticprogramming.com/. Our current work uses the software gmax-bio™ (http://www.abergc.com).
Nonlinear mapping problem: This describes a problem which can be cast, as in Fig. 2, in terms of the transformation of a set of input data to an output such that varying the input varies the output. Initially the nature of the mapping is unknown (and nonlinear), and the aim is to discover it.
Operator (or function): Something that takes one or more inputs and transforms them into something else. In genetic programming trees of the type shown in Fig. 3, the operators are represented by the purple rectangles and each has only one output. Operators can be arithmetic (e.g. plus, minus, divide, multiply, log, square root, etc.), logical (e.g. IF A<B output = 1, ELSE output = 0; IF A is true AND B is true output 1, ELSE output 0; etc.), trigonometric (e.g. input = A, output = sin(A)) and so on.
Parameters and variables: In the mathematical modelling of biological networks we discriminate between parameters and variables. The parameters of a system are those things that are either controlled at known values by the experimenter or are inherent to the system and do not change during the experiment. As well as all the initial conditions and the structure of the network in terms of its interactions, examples of the parameters might include pH (if buffered) and the Michaelis and catalytic rate constants of enzymes, or the concentration of an added effector, such as IPTG in a gene expression induction study. The variables of such a system are those things that change during an experiment, and in metabolic networks would especially be the concentrations of metabolites and metabolic fluxes. In genetic networks they might also be the concentrations of proteins and of signalling molecules.

References

a Cramer, N.L. (1985) A representation for the adaptive generation of simple sequential programs. In *Int. Conf. Genetic Algorithms and their Applications*, pp. 183–187
b Koza, J.R. (1992) *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press
c Koza, J.R. (1994) *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press
d Koza, J.R. *et al.* (1999) *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann

Douglas B. Kell
Dept of Chemistry, UMIST, PO Box 88, Sackville St, Manchester, UK M60 1QD.
e-mail: dbk@umist.ac.uk

  PII: S0168-9525(02)02765-8

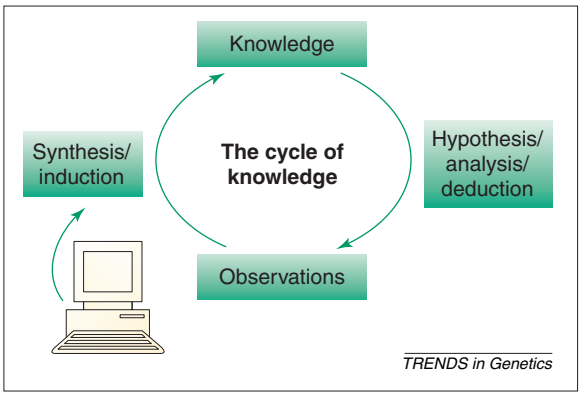

Fig. 1. Scientific advance can be seen as an iterative cycle linking knowledge and observations. The hypothetico-deductive mode of reasoning [60] uses background knowledge to construct a hypothesis that is tested experimentally to produce observations. This is only half the story, however, as the inductive mode of reasoning is based purely on generalizing rules (or hypotheses) from examples; that is, it is purely data driven (and the hypothesis is the end, not the beginning). Because of the high dimensionality of typical data, computer-intensive methods are required to turn the data into knowledge.

effect a mathematical transformation that will produce the desired output [18]. Bootstrapping or cross-validation methods (e.g. [28]), using data for which the answer is known but which were not seen during the training phase, ensure that the mappings produced are robust and generalize appropriately. So, for instance, microarray data were recently collected from breast tumour biopsies where the patient outcome was known [29]. By a series of iterative trials – training – a set of transcripts could be identified that were good predictors of outcome. A similar strategy was used to advantage with serum proteins and ovarian cancer [30].

There are a great many possible machine learning methods that can be applied to mappings of this type. They include neural [31,32], statistical [24], logic-based [33], rule-based [34,35] and evolutionary computing [36,37] strategies. Our focus is on the last, and in particular on a subset of methods popularized as 'genetic programming' (GP) [5,38–43]. (Note that our use of 'GP' here is strictly in its computer science sense.) Here we shall concentrate on the more traditional tree-based encoding [44] (Fig. 3). The attraction of the tree structure is that it is possible to 'mutate' any program by removing a subtree (along with everything below it) and replacing it with anything else that has a single output, thereby preserving the syntax (i.e. the number and nature of inputs that the node above it expects). Similarly, one can effect 'recombination' between two programs by removing a subtree from each of two (or more) programs and swapping them around. Evolutionary computing strategies of this type then allow one to evolve simple rules to solve complex data mining problems, according to the general algorithm given in Fig. 4.

The result of this process is a rule that not only effects the desired nonlinear mapping, but that also has explanatory power: we learn not only which input variables are important to the output of interest, but also the functional form of the relationship between them. This is obviously generic in character, and has the attraction that the combination of a small number of input variables and nonlinear OPERATORS allows a simple and robust mapping that simultaneously identifies both the important input variables and how they interact. Most significantly, however, if the inputs are genetic markers or transcript levels from, for example, a microarray measurement and the output is a disease presence or absence (coded 1 or 0, respectively) or other high-level trait, the rule evolved would in fact be an exact genotype–phenotype mapping. In this sense, we see that the gene encodes the program.

Genes as computer programs

We are of course used to considering developmental processes in terms of an ordered programme of genetic expression events, where genes are turned on temporally. What are the consequences of accepting the direct analogy of genes as computer programs? I think the most important will lie in several main areas (Table 1). First, these methods provide an effective approach to biomarker or 'surrogate marker' detection

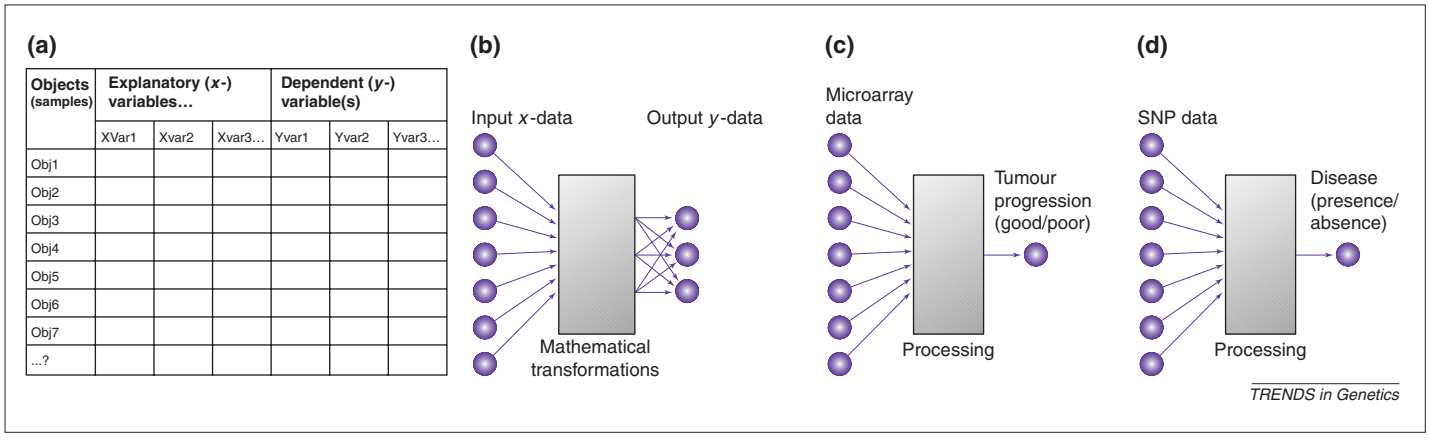

Fig. 2. The genotype–phenotype (nonlinear) mapping problem. (a) All data sets can be viewed in terms of a spreadsheet table, in which different samples (individuals, objects) appear in different rows, and the values or classes of variables (properties) associated with them appear in different columns. (Of course data for some variables could be missing in some samples, whether because they were not collected, are considered unreliable, do not apply to the particular individual, or whatever.) It is frequently the case that we wish to account for some of the properties in terms of appropriate combinations of some of the other variables. The ones we want to account for are usually termed the 'dependent variables', '*y*-variables' or '*y*-data', and the ones contributing to the explanation are usually called the 'explanatory variables', '*x*-variables' or '*x*-data'. In the present case the explanatory variables will be genotypic or expression profiling data, and the dependent variables are phenotypic traits such as disease susceptibility. (b) Machine learning in its commonest forms uses data set out as in (a) but represents them as an association or nonlinear mapping between coupled inputs (*x*-data) and outputs (*y*-data). The aim of machine learning is then to use some or all of the inputs and determine a mathematical transformation that produces the correct output(s) when presented with the relevant input. Here, the inputs might typically be expression profiling data from a series of microarray (c), proteome or metabolome experiments, or allelic or polymorphism markers (d), in which one also has knowledge of the phenotypes of interest of the same individuals (which are then the outputs).

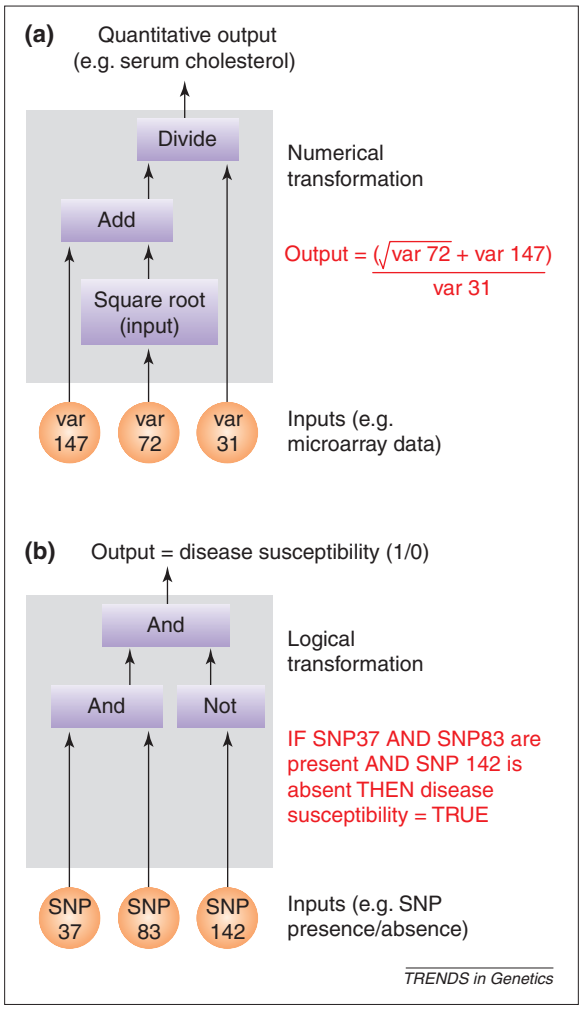

Fig. 3. Encoding or representing an input–output relationship as a tree. In a tree-based structure, the input variables that are selected are the 'terminals' (orange circles) and are acted on by operators or functions (purple boxes) that form nodes. The tree is read from the bottom upwards. The 'tree' representation on the left is exactly equivalent to the red 'equation' representation on the right. In genetic programming (GP), any transformation function can be used as an operator (including arithmetic, trigonometric, logical and so on). The GP is synonymously a program and a rule. (a) A numerical example. (b) An example using SNPs to predict disease susceptibility.

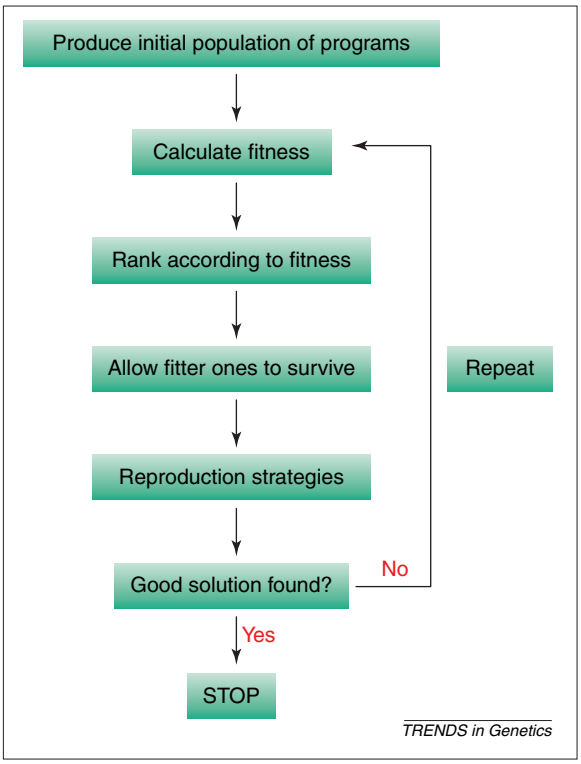

Fig. 4. The basic principle of evolutionary computing. We have an initial population of programs of the type shown in Fig. 3, commonly created by randomly combining subsets of the input variables and functions. We evaluate them to establish their 'fitness', which normally means their ability to give the correct input–output mapping (although we can also 'penalize' the fitness of bulky programs in favour of simpler ones). We then select – in part on the basis of fitness – some of these to act as parents, and use them to breed another generation (by mutation and recombination of those preferentially selected). This essentially darwinian process continues through the evaluation cycle repeatedly until a stopping criterion is met. This could be several generations, a continuing failure to improve, or a complete solution of the problem.

in phenotypic mapping, where both the inputs and outputs are phenotypic, and where by transcriptomic [45], proteomic [46] or metabolomic [5,47–50] methods we can map the underlying causes to observable phenotypes. Second, we can effect genotype–phenotype mapping directly by using the presence or absence of particular polymorphisms as the input, and the trait of interest as the output, for example in 'disease association' studies. In both of these cases, the power of the GENETIC PROGRAMMING is that it provides a straightforward and natural approach to the analysis of 'synthetic phenotypes' or multigenic traits in which the phenotype depends on the presence of multiple alleles (for a notable example in which each of six loci needed to be present to see a 'blockbuster' phenotype, see [51]).

Third, we can use such associations in quantitative trait loci (QTL) (and other genetic) mapping where the inputs are any complex phenotypic data (e.g. from expression profiling) and the outputs (encoded 1 or 0) are the presence or absence of a suitable genetic marker in the relevant organism. The phenotypic rules that are evolved will only 'fire' (return a value close to 1) when the genetic markers are present, but can themselves be considered to be an expression of the genotype; thus, the rules themselves are pseudo-genetic markers, which we refer to as 'phenogenes'. Such rules might be of the form: IF (understated environmental conditions) the ratio of proteome spot 472: proteome spot 1511 >3.7,

Table 1. Some important nonlinear input–output mappings that could benefit from the analogy of 'genes as programs', and from the use of genetic programming in learning rules that represent the nonlinear mapping

Domain	Input and its encoding	Output and its encoding	Refs
Genetic basis of disease or other traits	Quantitative transcriptome data	Disease presence or severity	[61,62]
Genetic basis of disease or other traits	Polymorphism presence/absence, SNP and/or haplotype data	Disease presence or severity	[63–65]
Genetic mapping of quantitative traits	Expression profiling data (transcriptome, proteome, metabolome)	Presence/absence of a molecular genetic marker (e.g. RFLP, SNP, etc.)	[53,54]
Strain improvement in biotechnology	Expression profiling data (transcriptome, proteome, metabolome)	Productivity or titre	[66]

THEN {gene or allele name, but not the genes encoding proteins 472 or 1511} is present. It is clear that the evaluation of such a rule will return either 1 (true) or 0 (false). An attraction of this particular idea, which is related to the concepts of 'genetical genomics' [52] and 'expression level polymorphism' (D. St Clair and R. Michelmore, unpublished results, cited in [53]), is that it is possible to evolve any number of these phenogenes from expression profiling data obtained from suitable mapping populations, and thus have very high density markers indeed. This is because for any n expression profiling markers that one has, the number of combinations that use any m of them scales as n^m [43]. The presence or absence (1 or 0) data of the phenogenes are then read into standard software for QTL mapping [53–56]. Finally, these methods are of interest in any area where a large number of potential loci, but a much smaller number of actual loci, contribute significantly to a continuous output of interest, such as the productivity of a particular strain in biotechnology. Here we know that changing the concentration of individual enzymes is unlikely to change the flux to desirable end-products significantly, due to the organization of enzymes into metabolic pathways obeying nonlinear kinetics [57–59]. The availability of transcriptome and/or proteomic profiles with associated yield values at suitable times will enable an efficient nonlinear mapping to determine which combinations of genes should best be altered (and how) so as to effect the desired yield improvement.

Conclusions

A genotype–phenotype mapping can be encoded in the form of a tree (or indeed a directed acyclic graph [18]). The methods of genetic programming allow us to evolve such trees by mutation and recombination, to produce good representations that permit an efficient, robust and parsimonious mapping. In this sense, the rule evolved by the GP *is* the nonlinear mapping, relating events at the genetic level to the higher-order processes that are typically of medical, agricultural or biotechnological interest. In other words, a genetic locus can fairly be represented as a kind of computer program.

Acknowledgements
I thank Ian King for useful discussions and the BBSRC for financial support.

References

1 Oliver, S.G. (1996) From DNA sequence to biological function. *Nature* 379, 597–600
2 Bork, P. *et al.* (1998) Predicting function: From genes to genomes and back. *J. Mol. Biol.* 283, 707–725
3 Brent, R. (1999) Functional genomics: Learning to think about gene expression data. *Curr. Biol.* 9, R338–R341
4 Brent, R. (2000) Genomic biology. *Cell* 100, 169–183
5 Kell, D.B. *et al.* (2001) Genomic computing: explanatory analysis of plant expression profiling data using machine learning. *Plant Physiol.* 126, 943–951
6 Kell, D.B. and Mendes, P. (2000) Snapshots of systems: metabolic control analysis and biotechnology in the post-genomic era. In *Technological and Medical Implications of Metabolic Control Analysis* (Cornish-Bowden, A. and Cárdenas, M.L., eds.), pp. 3–25 (and see http://qbab.aber.ac.uk/dbk/mca99.htm), Kluwer Academic Publishers
7 Mendes, P. (1997) Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends Biochem. Sci.* 22, 361–363
8 Tomita, M. *et al.* (1999) E-CELL: software environment for whole-cell simulation. *Bioinformatics* 15, 72–84
9 Giersch, C. (2000) Mathematical modelling of metabolism. *Curr. Opin. Plant Biol.* 3, 249–253
10 Edwards, J.S. *et al.* (2001) In silico predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. *Nat. Biotechnol.* 19, 125–130
11 von Dassow, G. *et al.* (2000) The segment polarity network is a robust development module. *Nature* 406, 188–192
12 Mendes, P. and Kell, D.B. (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics* 14, 869–883
13 D'haeseleer, P. *et al.* (2000) Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* 16, 707–726
14 Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann
15 Pearl, J. (2000) *Causality: models, reasoning and inference*, Cambridge University Press
16 Shipley, B. (2001) *Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference*, Cambridge University Press
17 Mitchell, T.M. (1997) *Machine learning*, McGraw Hill
18 Kell, D.B. and King, R.D. (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
19 Eisen, M.B. *et al.* (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. U. S. A.* 95, 14863–14868
20 Tamayo, P. *et al.* (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. U. S. A.* 96, 2907–2912
21 Altman, R.B. and Raychaudhuri, S. (2001) Whole-genome expression analysis: challenges beyond clustering. *Curr. Opin. Struct. Biol.* 11, 340–347
22 Everitt, B.S. (1993) *Cluster Analysis*, Edward Arnold
23 Duda, R.O. *et al.* (2001) *Pattern classification, 2nd ed.*, John Wiley
24 Hastie, T. *et al.* (2001) The elements of statistical learning: data mining, inference and prediction, Springer-Verlag
25 Jolliffe, I.T. (1986) *Principal Component Analysis*, Springer-Verlag
26 Oliver, S.G. (2000) Proteomics: guilt-by-association goes global. *Nature* 403, 601–603
27 Altshuler, D. *et al.* (2000) Guilt by association. *Nat. Genet.* 26, 135–137
28 Chatfield, C. (1995) Model uncertainty, data mining and statistical inference. *J. R. Stat. Soc. Ser. A* 158, 419–466
29 van 't Veer, L.J. *et al.* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415, 530–536
30 Petricoin, E.F. *et al.* (2002) Use of proteomic patterns in serum to identify ovarian cancer. *Lancet* 359, 572–577
31 Michie, D. *et al.* eds (1994) Machine learning: neural and statistical classification, Ellis Horwood
32 Lucek, P. *et al.* (1998) Multi-locus nonparametric linkage analysis of complex trait loci with neural networks. *Hum. Hered.* 48, 275–284
33 King, R.D. *et al.* (1996) Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proc. Natl. Acad. Sci. U. S. A.* 93, 438–442
34 Quinlan, J.R. (1993) *C4.5: programs for machine learning*, Morgan Kaufmann
35 Adamo, J-M. (2001) *Data Mining for Association Rules and Sequential Patterns*, Springer-Verlag
36 Bäck, T. *et al.* eds (1997) *Handbook of evolutionary computation.*, IOP Publishing/Oxford University Press
37 Foster, J.A. (2001) Evolutionary computation. *Nat. Rev. Genet.* 2, 428–436
38 Koza, J.R. (1992) *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press
39 Koza, J.R. (1994) *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press
40 Banzhaf, W. *et al.* (1998) *Genetic Programming: An Introduction*, Morgan Kaufmann
41 Koza, J.R. *et al.* (1999) *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann
42 Langdon, W.B. and Poli, R. (2002) *Foundations of genetic programming*, Springer-Verlag
43 Kell, D.B. (2002) Defence against the flood: a solution to the data mining and predictive modelling challenges of today. *Bioinformatics World* 1, 16–18
44 Cramer, N.L. (1985) A representation for the adaptive generation of simple sequential programs. In *Int. Conf. Genetic Algorithms and their Applications*, pp. 183–187
45 Gilbert, R.J. *et al.* (2000) Genomic computing: explanatory modelling for functional genomics. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)* (Whitley, D. *et al.*, eds.), pp. 551–557, Morgan Kaufmann
46 Link, A.J. ed. (1999) *2-D proteome analysis protocols*, Humana Press
47 Schilling, C.H. *et al.* (1999) Toward metabolic phenomics: Analysis of genomic data using flux balances. *Biotechnol. Prog.* 15, 288–295
48 Johnson, H.E. *et al.* (2000) Explanatory analysis of the metabolome using genetic programming of simple, interpretable rules. *Genetic Progr. Evolvable Machines* 1, 243–258

49 Raamsdonk, L.M. *et al.* (2001) A functional genomics strategy that uses metabolome data to reveal the phenotype of silent mutations. *Nat. Biotechnol.* 19, 45–50
50 Fiehn, O. (2002) Metabolomics: the link between genotypes and phenotypes. *Plant Mol. Biol.* 48, 155–171
51 Lippman, Z. and Tanksley, S.D. (2001) Dissecting the genetic pathway to extreme fruit size in tomato using a cross between the small-fruited wild species *Lycopersicon pimpinellifolium* and *L. esculentum* var. giant heirloom. *Genetics* 158, 413–422
52 Jansen, R.C. and Nap, J.P. (2001) Genetical genomics: the added value from segregation. *Trends Genet.* 17, 388–391
53 Doerge, R.W. (2002) Mapping and analysis of quantitative trait loci in experimental populations. *Nat. Rev. Genet.* 3, 43–52
54 Kearsey, M.J. and Pooni, H.S. (1996) *The genetical analysis of quantitative traits*, Nelson Thomas
55 Manly, K.F. and Olson, J.M. (1999) Overview of QTL mapping software and introduction to Map Manager QT. *Mamm. Genome* 10, 327–334
56 Flint, J. and Mott, R. (2001) Finding the molecular basis of quantitative traits: Successes and pitfalls. *Nat. Rev. Genet.* 2, 437–445
57 Kell, D.B. *et al.* (1989) Control analysis of microbial growth and productivity. *Symp. Soc. Gen. Microbiol.* 44, 61–93
58 Cornish-Bowden, A. (1995) Kinetics of multi-enzyme systems. In *Biotechnology* (Vol. 9) (Rehm, H.J. *et al.*, eds.), pp. 121–136, Verlag Chemie
59 Westerhoff, H.V. and Kell, D.B. (1996) What BioTechnologists knew all along? *J. Theor. Biol.* 182, 411–420
60 Oldroyd, D. (1986) *The Arch of Knowledge: An Introduction to the History of the Philosophy and Methodology of Science*, Methuen
61 Golub, T.R. *et al.* (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537
62 Jones, M.B. *et al.* (2002) Proteomic analysis and identification of new biomarkers and therapeutic targets for invasive ovarian cancer. *Proteomics* 2, 76–84
63 Wang, D.G. *et al.* (1998) Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome. *Science* 280, 1077–1082
64 Stephens, J.C. *et al.* (2001) Haplotype variation and linkage disequilibrium in 313 human genes. *Science* 293, 489–493
65 Bader, J.S. (2001) The relative power of SNPs and haplotype as genetic markers for association tests. *Pharmacogenomics* 2, 11–24
66 Epstein, C.B. and Butow, R.A. (2000) Microarray technology – enhanced versatility, persistent challenge. *Curr. Opin. Biotechnol.* 11, 36–41

Evidence for population growth in humans is confounded by fine-scale population structure

Susan E. Ptak and Molly Przeworski

Although many studies have reported human polymorphism data, there has been no analysis of the effect of sampling design on the patterns of variability recovered. Here, we consider which factors affect a summary of the allele-frequency spectrum. The most important variable to emerge from our analysis is the number of ethnicities sampled: studies that sequence individuals from more ethnicities recover more rare alleles. These observations are consistent with fine-scale geographic differentiation as well as population growth. They suggest that the geographic sampling strategy should be considered carefully, especially when the aim is to infer the demographic history of humans.

Published online: 23 September 2002

Genetic variation among extant humans carries information about the evolutionary history of our species. Unlinked regions in the genome represent independent realizations of this evolutionary process and thus, with polymorphism data from enough loci, it should be possible to infer many aspects of our evolution [1–4]. Conversely, a better understanding of the evolutionary history of humans should help us to predict patterns of genetic variability, thereby aiding in the design and interpretation of genome-wide association studies [5,6]. It will also help us to interpret polymorphism data from regions of the genome that have experienced natural selection [7].

With these myriad goals in mind, researchers have collected polymorphism data from more than 400 regions of the human genome in over 40 studies. The loci have been sequenced in different laboratories and distinct strategies have been implemented regarding the number and variety of geographic sampling localities, the number of individuals considered and so forth. The patterns of variability recovered have been extremely varied. To some extent, this variability is expected: patterns of polymorphism will differ greatly from locus to locus by chance, even if they have been generated by exactly the same evolutionary process [8,9]. However this variance might also reflect differences among study designs. If there are aspects of the sampling strategy that influence patterns of variation, their identification should inform the design of future studies. It can also point to important features of the evolutionary history of human populations [10].

It is commonly quoted that 85% of human genetic diversity is found within populations [11], a finding usually interpreted as evidence that human populations are genetically very similar to one another [12]. Although this is certainly true (if only because, on average, two humans are identical at 99.9% of their DNA), this level of population structure is sufficient to have profound effects on levels of linkage disequilibrium in some contexts [5,13]. Furthermore, a high proportion of alleles seem to be specific to samples from single populations [14]. It therefore seems plausible that the geographic sampling scheme influences the allele-frequency spectrum as well as levels of allelic associations. To investigate this

 0168-9525/02/$ – see front matter PII: S0168-9525(02)02781-6