Congress on Evolutionary Computation 2003, Canberra, 8–12 Dec 2003, pp807–814, IEEE press

Predicting Biochemical Interactions – Human P450 2D6 Enzyme Inhibition

W. B. Langdon
Computer Science, University College, London. Gower Street, London, WC1E 6BT, UK
http://www.cs.ucl.ac.uk/staff/W.Langdon

S. J. Barrett
Data Exploration Sciences, GlaxoSmithKline, Research and Development, Greenford, Middlesex, UK

B. F. Buxton
Computer Science, University College, London. Gower Street, London, WC1E 6BT, UK
http://www.cs.ucl.ac.uk/staff/B.Buxton

Abstract- *In silico* screening of chemical libraries or virtual chemicals may reduce drug discovery and medicine optimisation lead times and increase the probability of success by directing search through chemical space. About a dozen intelligent pharmaceutical QSAR modelling techniques were used to predict IC50 concentration (three classes) of drug interaction with a cell wall enzyme (P450 CYC2D6). Genetic programming gave comprehensible cheminformatics models which generalised best. This was shown by a blind test on GlaxoWelcome molecules of machine learning knowledge nuggets mined from SmithKline Beecham compounds. Performance on similar chemicals (interpolation) and diverse chemicals (extrapolation) suggest generalisation is more difficult than avoiding over fitting.

Two GP approaches, classification via regression using a multi-objective fitness measure and a direct winner takes all (WTA) or one versus all (OVA) classification, are described. Predictive rules were compressed by separate follow up GP runs seeded with the best program.

1 Introduction

GlaxoSmithKline (GSK) have evaluated a number of Quantitative Structure Activity Relationship (QSAR) computational modelling techniques. The merger of two major pharmaceutical companies, GlaxoWelcome (GW) and SmithKline Beecham (SKB) to form GSK, has given an opportunity to test generalisation of QSAR approaches. Both companies developed sizable libraries of chemicals, with surprisingly little overlap and occupying different areas of chemical space. A blind trial was held, in which more than a dozen different groups/techniques were invited to train classifiers on a labelled data set of former SKB chemicals. A non-overlapping unlabelled data set of former GW chemicals was also given. The task being to predict the activity of the former GW data set. Twelve entries were received.

A complete description of all of the techniques is being prepared for publication, however we give details of two genetic programming techniques (Sections 3 and 4) including mutation operators (Section 3.1) and simplification runs (Section 4.2). This is followed by a general discussion (Section 5) and conclusions (Section 6) but we start in the next section with a description of the problem and data sets.

2 Human P450 Cytochrome 2D6 IC50 data

Over the last few years GlaxoSmithKline has been measuring the *in vitro* activity of thousands of chemicals (drawn from both former SKB and former GW libraries) against Cytochrome P450. P450 are a family of important cell membrane enzymes which interact with many drugs [Ball and Borman, 1997; Cupp and Tracy, 1998]. An adverse reaction with a P450 enzyme may prevent the use of a potential drug. If the reaction of proto-drugs with P450 and other important molecules where known in advance then huge savings in drug discovery lead time could be made. Also directing the search to more fruitful areas of chemical space may increase the chance of finding a successful drug treatment. Computer based screening has the advantage that not only can it be applied to existing chemicals but it can, in principle, be applied to virtual chemicals. I.e. *in silico* screening can be applied to chemicals before they are synthesised.

P450 CYC2D6 is a large and complex molecule. However these machine learning prediction techniques are not specific to it. We may hope that a successful ML technique will be readily applied, by retraining, not only to other P450 molecules, but to other pharmaceutically important biomolecules. In contrast, techniques based on three dimensional crystallographical chemical models (such as [Segall *et al.*, 1998]) are specific to a crystal structure. Not only do they require the biomolecule structure (which in many cases is not known) but considerable human endeavour may be needed to re-engineer them for another molecule.

Initialy high throughput screening (HTS) was used [Langdon *et al.*, 2001; Langdon *et al.*, 2002; Langdon *et al.*, 2003] however more accurate IC50 measurements are now available. The correspondence between HTS and IC50 measurements has been disappointing, so we will concentrate on IC50.

The IC50 is the concentration at which a compound reduces the effectiveness of P450 by 50%. A chemical which interacts very little will need to be present in high concentration to have a large affect. In order to find a chemical's IC50 value, repeated measurements must be made at different concentrations. If a chemical is found to be inactive even at high concentrations, this is all that need be known and it is common not to continue testing at even higher concentrations. Instead, its IC50 value is recorded as "at least x". Figure 1 shows this effect in the training data, where 552 chemicals have an IC50 value of "100", meaning that it is at least 100.

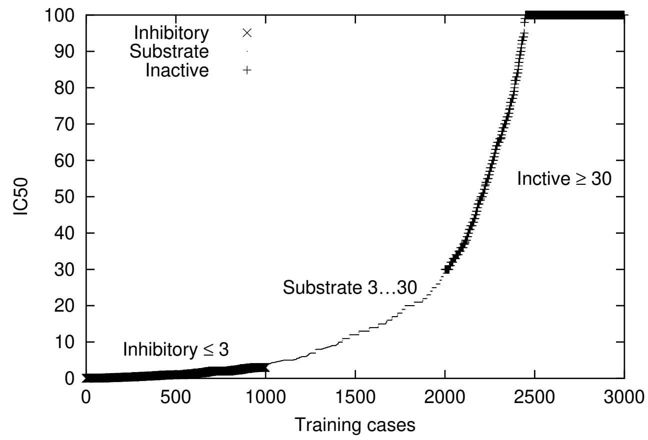


Figure 1: Human P450 CYC2D6 IC50 training data

Figure 1 also shows that the continuous IC50 data is divided into three classes. Inhibitory (IC50 $\leq$ 3), Substrate ($3 <$ IC50 < 30) and Inactive (IC50 $\geq$ 30). (The middle class is called "Substrate" on the assumption that chemicals with intermediate activity depress P450 metabolism of the standard substrate by being an alternative substrate with which the P450 molecules also react, thus leaving less P450 free for the original reaction.) Note also the training data has been specifically prepared for use with other machine learning techniques and so has been balanced to have exactly 1000 chemicals in each of the three classes.

The supplied training data includes the compound id, its "SMILES" representation[1], its true class and IC50 value. We did not used the compound's id or SMILES directly, instead (like most of the 12 groups taking part in the trial) we used 121 "features" pre-calculated by GSK. (Note these features are not the same as the 699 features we previously used with HTS data [Langdon *et al.*, 2001, 2002, 2003].) Many of the features are proprietary and represent considerable domain knowledge. A simple feature is charge imbalance. Most of the participating approaches used these features but a few used the SMILES representation.

In addition to the 3,000 training compounds, features for 4,570 former SKB compounds were provided (without IC50 values). This served as the first test set. Similarly, features for 1932 former GW compounds (again without IC50 values) were given. This served as the extrapolation set. Both unlabelled sets have large class imbalances, with a very much lower proportion of active chemicals than the training data (see Tables 4 and 5). That is, the former SKB test set contains $69 + 19 + 3 = 91$ inhibitory chemicals out of 4570 (2%). While the former GW test set contains $41 + 32 + 41 = 114$ P450 inhibitors out of a total of 1932 chemicals (6%).

[1] SMILES represents chemicals as a graph of atoms connected by bonds. E.g. alcohol is CCO. Hydrogen atoms are not explicitly given.

Table 1: IC50 Prediction (Regression and Classification)

Objective:	Predict Human P450 CYC2D6 IC50 class (Inhibitory, Substrate, Inactive)
Function set:	Max Min MaxA MinA MUL ADD DIV SUB IFLTE
Terminal set:	121 features, 0 1 2 3 4 5 6 7 8 9, plus 90 unique constants -34..582.
Fitness:	Fitness $= 20000$ hits $- \sum \lvert f_i - \mathrm{IC50}_i \rvert^2$ or hits
Selection:	generational (non elitist), tournament 7
Initial pop:	Each individual is either one or three trees. Each tree created by ramped half-and-half (2:6) (each initial tree limited to 300)
Parameters:	Population 5000. Max program size 1000. 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

3 Genetic Programming Configuration

We tried two GP approaches, classification via regression and direct classification. In regression each individual in the population consists of a single tree which yields a floating point value f. (Figure 6 gives an example tree.) The first component of fitness is the square of the difference between f and the measured IC50 value summed over all 3,000 training compounds. f is also converted to one of three classes using the same cut points as had been applied to the IC50 value. I.e. $f \leq 3.0$ indicates an active compound, $f \geq 30.0$ predicts the compound to be inactive, while between 3 and 30 is a substrate. Each compound where the individual correctly predicts the class is known as a hit. The fitness measure combines error squared and hits by multiplying the number of hits by 20,000 and subtracting the sum of error squared, Fitness $= 20000$ hits $- \sum |f_i - \mathrm{IC50}_i|^2$. (A certain amount of experimenting was required in order to choose 20,000 as a reasonable weighting between the two factors. Pareto multi-objective approaches [Langdon, 1998] might also have been tried.)

The second approach is a "winner takes all" or "one versus all" in which each GP individual contained three trees [Langdon, 1998], one per class. Each tree is like the single one used in the regression approach. They each return a floating point number. The class predicted is that corresponding to the tree which yielded the largest value. Fitness is simply the number of compounds correctly predicted. (cf. Figure 9). Apart from the number of trees and the fitness function, the two approaches are identical. The GP parameters and function and terminal sets are similar to our earlier work. Table 1 contains the details.

The functions within the evolved trees are binary except for the four argument IFTLE (If less than or Equal). IFLTE evaluates its first two arguments and then returns either the third (if the first is less than or equal to the second). Otherwise it returns its fourth argument.

In both approaches five GP runs were made. The best model for each was simplified by a separate GP run to give the final two models.

3.1 Mutation

As in our earlier work we used a high mutation rate and a mixture of different mutation operators [Angeline, 1998]. The four mutation operators are.

Subtree chooses a tree uniformly at random, then chooses uniformly at random a node within it. Replace the subtree rooted at that node (which may be a leaf or function) with a new random one. The new subtree is created in the same way as those in the initial population, i.e. ramped half-and-half. Ramped half-and-half's maximum depth is randomly chosen from one to the depth of the subtree being replaced. This may lead to either a decrease or increase in program size. Should the mutant be larger than the maximum program size, it is aborted and a new mutation is attempted.

Shrink replaces a subtree with part of itself. A node is chosen in the same way as in subtree mutation. If it is a leaf, no action is taken and the offspring is identical to its parent. If it is an internal node (function), a function amongst its argument subtree is chosen uniformly at random and is promoted up the tree replacing the whole subtree (cf. hoist [Kinnear, Jr., 1994]). It is possible the same internal node may be chosen a second time, in which case (as with a leaf) no change is made.

Point Mutation replaces a function with another with same number of inputs or replaces an input by another. Note if the Gaussian mutation of constants operator is enabled (as it was in these runs) then point mutation does not change leaves.

Point mutation differs from Subtree and Shrink mutations in that it can make multiple changes and because it applies to the whole of an individual, not just a single tree within it. (Remember, in the case of the classification approach, each individual consists of three trees.) Essentially point mutation scans the whole program from beginning to end. At each point within it, a random choice is made. With a probability of 100/1024 the point is replaced with a function (or a leaf) with the same number of inputs. (It is possible the function or terminal chosen for replacement is identical to the existing one, in which case there is effectively no change.)

Gaussian mutation replaces constants with another given by adding approximately Gaussian zero mean noise. If the constant is an integer, the noise standard deviation is 2.0, otherwise it is 5% of the constant's value.

All components of evolved programs (i.e. function and terminal sets), including constants, are fixed when the GP is started. Our GP has no ephemeral random constants. Hence the constant is replaced by the predefined constant nearest the randomly generated value.

Table 2: GP parameters used in run to simplify evolved P450 IC50 regression model (differences from Table 1)

Selection:	Keep best in population (elitism)
Pop Size:	500
Max Size:	68
Initial pop:	100% Seeded
Parameters:	10% size fair (crossover parameters as before). 90% mutation (point 5%, constants 5%, shrink 85%, subtree 5%). New subtrees created by subtree mutation created by ramped half-and-half (as before) but max depth 2 (min depth 1).

4 GP Results

4.1 GP Classification via Regression

Figures 2 and 3 show the performance of the best model from the last generation of each of the five runs. Of these the smallest and fittest was chosen. As a final stage the model was simplified, reducing its size from 65 to 45.

4.2 Simplification of Evolved Model

Except for the use of size fair crossover to avoid bloat [Langdon, 2000] we had not taken particular pains to evolve a compact model. A final stage was to perform a single small GP run in which the initial population was 100% seeded with 500 copies of the best model, and parameters (cf. Table 2) were adjusted with the aim of simultaneously reducing the size of the models in the population and keeping their performance high.

Mechanistic rules for transforming programs to exactly equivalent but smaller programs have been previously reported. E.g. [Hooper and Flann, 1996; Ekart, 2000; Ibarra *et al.*, 2002] all use explicit edit rules as special mutation operators which change the genetic material or program trees to yield equivalent offspring. High level languages such as Mathematica [Nachbar, 1995] and Maple include expression simplification within the language. Also [Olsson, 1995] produced a program to simplify programs which can be expressed as polynomials. However finding the smallest equivalent program is hard and so such rules have been heuristics. In using GP we also use a heuristic approach, however, we do not require the evolved models to be equivalent. Instead we rely on the fitness selection to try and find programs of similarly good performance. In fact after 50 generations a smaller model (51) with marginally improved performance was found. Figure 4 shows the evolution of the population's scores on the training data while Figure 5 show their size. On inspection six nodes in it were found which could be removed by hand to yield an even more compact model (45) with identical behaviour. It uses eight of the 121 available features (see Figure 6).

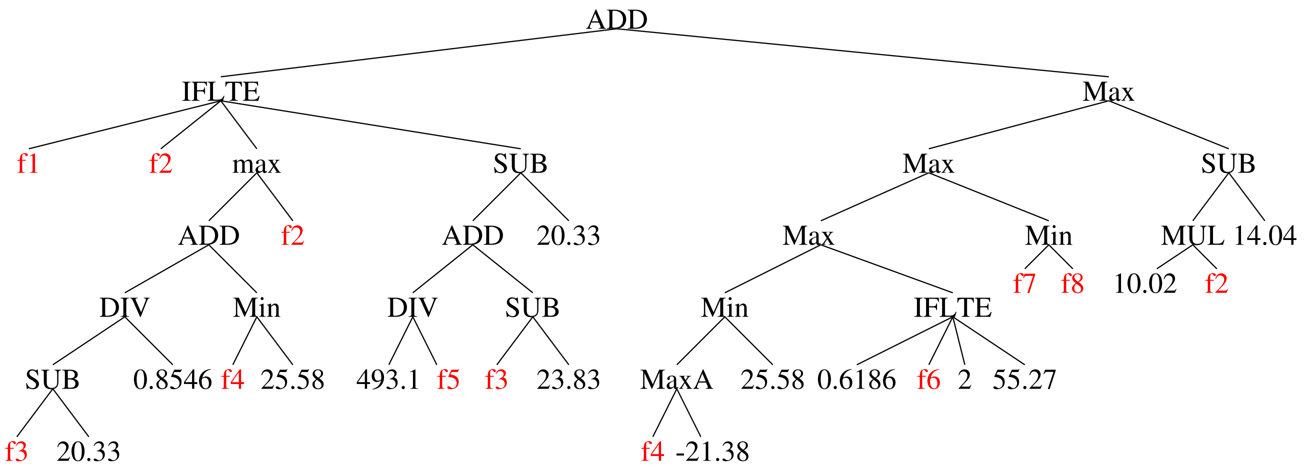


Figure 6: P450 IC50 (regression) model. f1, f2, ... f8 are GSK domain specific features calculated for each chemical from its SMILES representation. These 8 were chosen by GP from the 121 available.

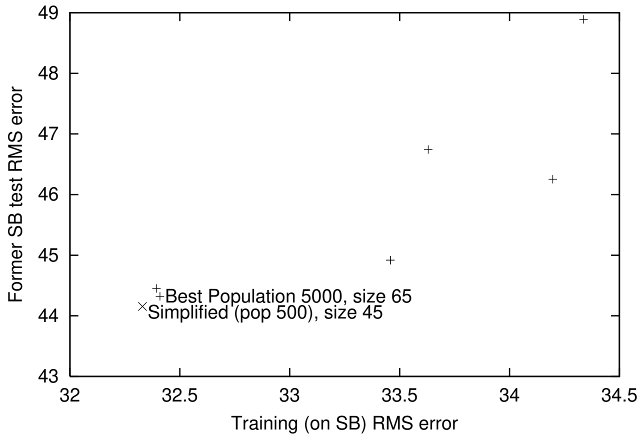


Figure 2: P450 CYC2D6 IC50 regression training v. test. (Remember the goal is to use regression as a route to classification. A poor RMS error need not imply a poor classification.)

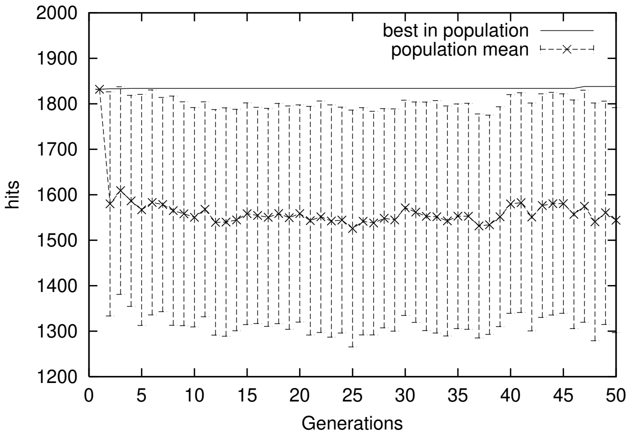


Figure 4: Using GP to shrink P450 regression model

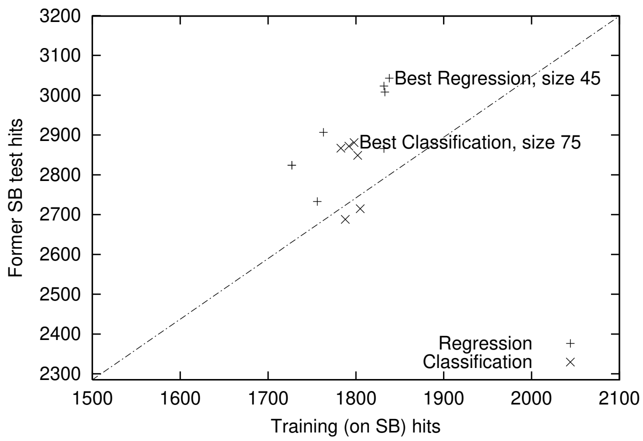


Figure 3: P450 CYC2D6 IC50 training v. test. Best of 5 regression runs and 5 classifications runs are plotted together with best simplified model of both GP approaches.

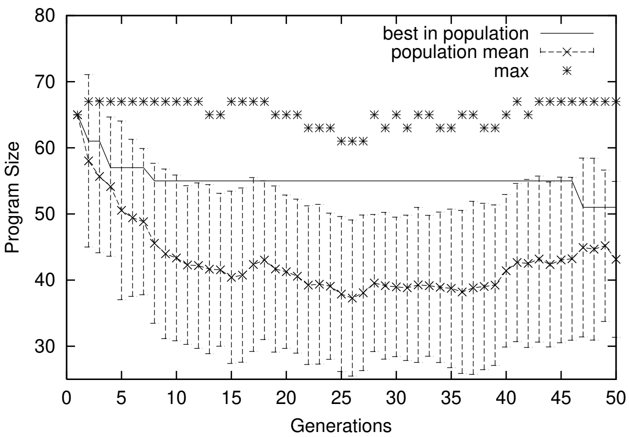


Figure 5: Using GP to shrink P450 regression model

Table 3: GP (regression) Training (on SKB) Performance

	Inhibitory	Substrate	Inactive	Accuracy
Inhibitory	579	294	127	58%
Substrate	188	504	308	50%
Inactive	45	200	755	76%
				Overall 1838/3000 = 61%

Table 4: GP (regression) former SKB test set

	Inhibitory	Substrate	Inactive	Accuracy
Inhibitory	69	19	3	76%
Substrate	306	778	479	50%
Inactive	153	566	2197	75%
				Overall 3043/4570 = 67%

Table 5: GP (regression) former GW extrapolation set

	Inhibitory	Substrate	Inactive	Accuracy
Inhibitory	41	32	41	36%
Substrate	92	119	235	27%
Inactive	103	242	1027	75%
				Overall 1187/1932 = 61%

The predictive accuracy, in terms of confusion matrices, of the final simplified model (using regression to obtain a classification) are shown on the training and two tests sets in Tables 3–5.

4.3 GP Direct Classification

Figure 3 gives the performance of the best classifier from the last generation of each of the five runs. Of these the smallest and fittest was chosen. Again the classifier was simplified, using the GP procedure described in the previous section with the same parameters (cf. Table 2), except the maximum size of the classifiers was 100 rather than 68. This reduced the classifier size from 93 to 75. See Figures 7 and 8. The final three way classifier uses 21 of the 121 available features (see Figure 9). Features f1 and f6 are used by both GP approaches.

The confusion matrices of the final simplified classifier as measured on the training and two tests sets are given in Tables 6–8.

5 Discussion

One of the biggest problems in applying machine learning to Quantitative Structure Activity Relationship (QSAR) has been over fitting. However both GP approaches do not appear to have suffered too badly, in that performance on the former SKB test set has been similar to that on the (SKB) training set (cf. Figure 3, Table 3 v. Table 4 and Table 6 v. Table 7). In other words, both GP approaches have generalised to unseen compounds from the same library. However when we look at performance (particularly on predicting active compounds) on the former GW compounds performance falls (cf. Table 5 and 8).

All the techniques tested in the blind trial gave worse performance on the former GW compounds, indicating that there is a systematic difference between the former SKB

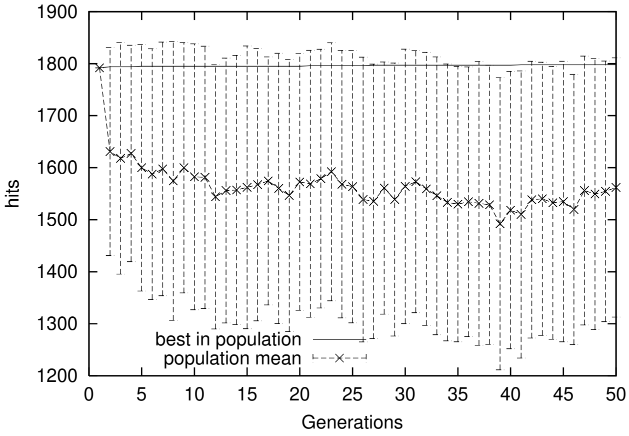


Figure 7: Using GP to shrink P450 classifier

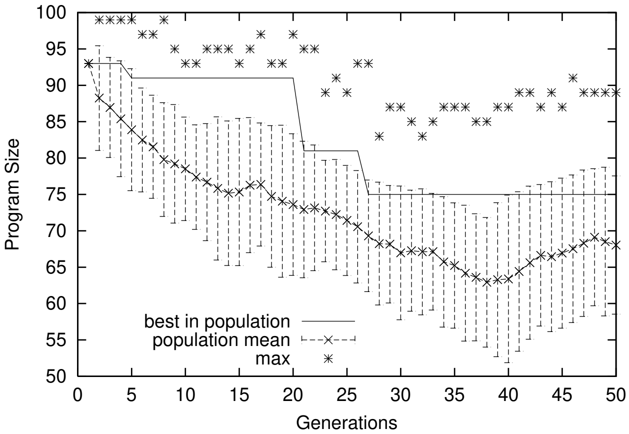


Figure 8: Using GP to shrink P450 classifier

Table 6: GP (classification) Training (SKB) Performance

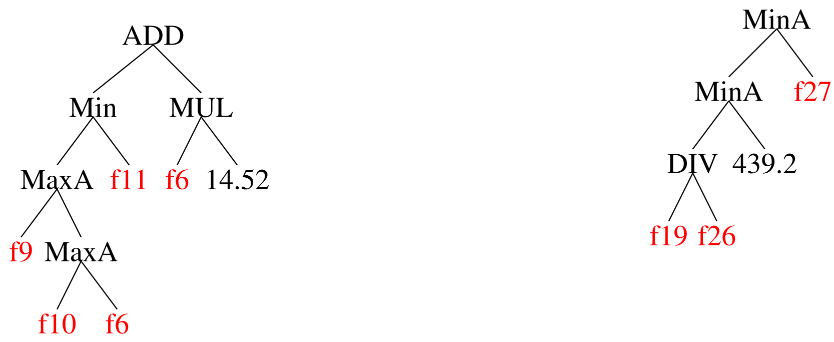
	Inhibitory	Substrate	Inactive	Accuracy
Inhibitory	568	352	80	57%
Substrate	187	569	244	57%
Inactive	64	275	661	66%
				Overall 1798/3000 = 60%

Table 7: GP (classification) former SKB test set

	Inhibitory	Substrate	Inactive	Accuracy
Inhibitory	56	31	4	62%
Substrate	342	813	393	53%
Inactive	179	740	2012	69%
				Overall 2881/4570 = 63%

Table 8: GP (classification) former GW extrapolation set

	Inhibitory	Substrate	Inactive	Accuracy
Inhibitory	45	26	43	39%
Substrate	62	148	236	33%
Inactive	96	360	916	67%
				Overall 1109/1932 = 57%



“Active” (left) and “inactive” (right) trees for P450 IC50 classifier.

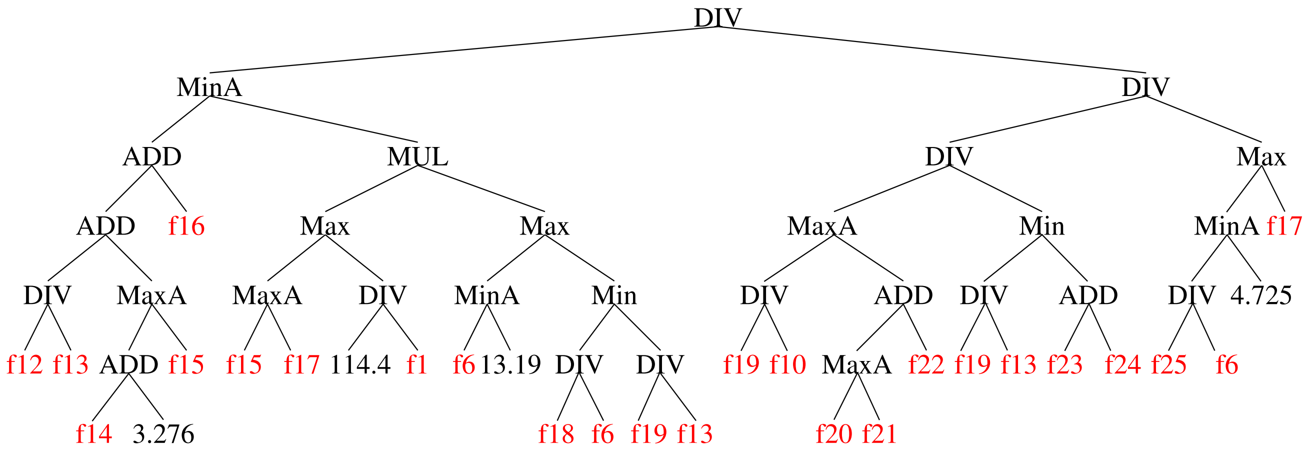


Figure 9: “Substrate” tree for P450 IC50 classifier. Each tree yields a value. The tree whose value is largest of the three indicates the predicted class. f1, f6, f9, ... f27 are proprietary features calculated from each chemical’s formula. 21 features were chosen by GP from the 121 available.

and former GW sets of compounds. One possible cause is that the IC50 values were measured differently, although it appears the same chemical assay was used for all the compounds. Instead it is thought that the systematic difference is due to the former SKB and former GW chemical libraries occupying different portions of chemical space. Indeed, this was part of the reason why the trial was set up this way. Computer models which can make predictions about chemicals similar to those within existing chemical libraries are useful in their own right. However it would also be very useful to be able to make predictions about chemicals (indeed even virtual chemicals) somewhat unlike those in use. The former GW compounds were thought to be different to those from SKB, and so it has proved. However none of the other techniques performed as well as GP on the second (former GW) extrapolation set.

In our earlier HTS experiments, GP performance had held up well when tested on outliers taken from the same library [Langdon *et al.*, 2003], so its GW performance is disappointing. Of course, that GP extrapolated well on HTS data, may have been due to the particular nature of the HTS dataset or perhaps due to the particular clustering technique used to designate chemicals as outliers.

From a machine learning point of view, it is easy to criticise the experiment, as it violates the underlying assumption that the training data is representative of the problem. The extrapolation set is from a different distribution to that used to train the classifier. However this is what industrial chemists want to do. Often they do not need a model of existing chemicals (usually they can look up their properties), they want to be able to make predictions about novel chemicals. They want to have some confidence that the chemical they are about to make will react in the body in the way they want.

We should remember that biochemistry occurs in three dimensions and so it should not come as a surprise that approaches based on treating complex molecules as two dimensional graphs (i.e. using their chemical formulae) are not 100% accurate. However real economic advantage could be obtained by using models with less than 100% accuracy provided they are predictive enough to be able to guide the drug discovery chemist.

Where proto-drugs are simple enough that their three dimensional structure is known or could be inferred, more complex three dimensional featured could be used. However the structure of biomolecules, including disease causing targets, may be unknown. This prevents the use of three dimensional modelling software which tries to find geometrical “docking” configurations between molecules. Nevertheless 3D features could still be used by adaptive machine learning approaches. Unfortunately the number of possible features is huge, so the first part of the data mining process becomes one of feature selection. I.e. deciding which features to use.

While the problem was posed as a three way classification, and most approaches were multi-class, it appears that the most important distinction is between active compounds and the rest. I.e. a binary classification. Perhaps binary classification would have been easier.

Our results suggest that perhaps using the IC50 measurement as part of the fitness function (via an error squared term) may give the classification via regression approach a modest advantage. However the small difference between the two GP approaches might also be due to using one tree instead of three. [Loveard and Ciesielski, 2001] investigates other ways to evolve multi-class classifiers.

Figure 1 shows the exponential nature (common in many chemical systems) of the distribution of IC50. It has been suggested that instead of IC50 values, we should have tried to model $-\log_{10}$ IC50 (pIC50) instead.

The models GP evolved (Figures 6 and 9) do provide some predictive ability. Another important aspect is that while evolved as "black box" classifiers, their inner workings are available to inspection, even alteration, by their users.

Since the GP classifiers are simple programs written in plain text, they can be readily translated to any format, be it C++, JavaScript or even a spreadsheet. Being simple they can be applied to even the largest chemical database with negligible overhead (compared to the database's own overheads and the cost of calculating the features).

6 Conclusions

Compact, comprehensible predictive models of the interaction between potential drugs and an important biological enzyme (human P450 CYC2D6) have been evolved using genetic programming. Their inputs are knowledge rich features which are readily computed from chemical formulae. They predict reasonably well on similar chemicals but have difficulty with compounds outside the chemical space on which they were trained. Nevertheless their extrapolation performance was found to be the best on a blind trial.

Acknowledgements

We would like to thank Sandeep Modi and Chris Luscombe.

Bibliography

[Angeline, 1998] Peter J. Angeline. Multiple interacting programs: A representation for evolving complex behaviors. *Cybernetics and Systems*, 29(8):779–806, November 1998.

[Ball and Borman, 1997] Simon Ball and Neil Borman. Pharmacogenetics and drug metabolism. *Nature Biotechnology*, 15:925–926, 1997.

[Cupp and Tracy, 1998] Melanie Johns Cupp and Timothy S. Tracy. Cytochrome P450: New nomenclature and clinical implications. *American Family Physician*, page 107, 1 January 1998.

[Ekart, 2000] Aniko Ekart. Shorter fitness preserving genetic programs. In C. Fonlupt, J.-K. Hao, E. Lutton, E. Ronald, and M. Schoenauer, editors, *Artificial Evolution. 4th European Conference, AE'99, Selected Papers*, volume 1829 of *LNCS*, pages 73–83, Dunkerque, France, 3-5 November 2000.

[Hooper and Flann, 1996] Dale Hooper and Nicholas S. Flann. Improving the accuracy and robustness of genetic programming through expression simplification. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, page 428, Stanford University, CA, USA, 28–31 July 1996. MIT Press.

[Ibarra *et al.*, 2002] Aitor Ibarra, J. Lanchares, J. Mendias, J. I. Hidalgo, and R. Hermida. Transformation of equational specification by means of genetic programming. In James A. Foster, Evelyne Lutton, Julian Miller, Conor Ryan, and Andrea G. B. Tettamanzi, editors, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 248–257, Kinsale, Ireland, 3-5 April 2002. Springer-Verlag.

[Kinnear, Jr., 1994] Kenneth E. Kinnear, Jr. Alternatives in automatic function definition: A comparison of performance. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 6, pages 119–141. MIT Press, 1994.

[Langdon *et al.*, 2001] W. B. Langdon, S. J. Barrett, and B. F. Buxton. Genetic programming for combining neural networks for drug discovery. In Rajkumar Roy, Mario Köppen, Seppo Ovaska, Takeshi Furuhashi, and Frank Hoffmann, editors, *Soft Computing and Industry Recent Applications*, pages 597–608. Springer-Verlag, 10–24 September 2001. Published 2002.

[Langdon *et al.*, 2002] William B. Langdon, S. J. Barrett, and B. F. Buxton. Combining decision trees and neural networks for drug discovery. In James A. Foster, Evelyne Lutton, Julian Miller, Conor Ryan, and Andrea G. B. Tettamanzi, editors, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 60–70, Kinsale, Ireland, 3-5 April 2002. Springer-Verlag.

[Langdon *et al.*, 2003] W. B. Langdon, S. J. Barrett, and B. F. Buxton. Comparison of adaboost and genetic programming for combining neural networks for drug discovery. In Günther R. Raidl, Stefano Cagnoni, Juan Jesús Romero Cardalda, David W. Corne, Jens Gottlieb, Agnès Guillot, Emma Hart, Colin G. Johnson, Elena Marchiori, Jean-Arcady Meyer, and Martin Middendorf, editors, *Applications of Evolutionary Computing, EvoWorkshops2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, EvoSTIM*, volume 2611 of *LNCS*, pages 87–98, University of Essex, UK, 14-16 April 2003. Springer-Verlag.

[Langdon, 1998] William B. Langdon. *Genetic Programming and Data Structures*. 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.

[Loveard and Ciesielski, 2001] Thomas Loveard and Victor Ciesielski. Representing classification problems in genetic programming. In *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1070–1077, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.

[Nachbar, 1995] Robert B. Nachbar. Genetic programming. *The Mathematica Journal*, 5(3):44–55, 1995.

[Olsson, 1995] Roland Olsson. Inductive functional programming using incremental program transformation. *Artificial Intelligence*, 74(1):55–81, March 1995.

[Segall *et al.*, 1998] M.D. Segall, M. C. Payne, S. W. Ellis, G. T. Tucker, and R. N. Boyes. First principles calculation of the activity of cytochrome p450. *Physical Review E*, 57(4):4618–4621, April 1998.