BIOINFORMATICS

Vol. 20 no. 18 2004, pages 3398–3405
doi:10.1093/bioinformatics/bth414

Mining HIV protease cleavage data using genetic programming with a sum-product function

Zheng Rong Yang[1,], Andrew R. Dalby[2] and Jing Qiu[2]*

[1]Department of Computer Science and [2]Department of Biology Sciences, Exeter University, UK

Received on August 13, 2003; revised on June 10, 2004; accepted on July 7, 2004
Advance Access publication July 15, 2004

ABSTRACT

Motivation: In order to design effective HIV inhibitors, studying and understanding the mechanism of HIV protease cleavage specification is critical. Various methods have been developed to explore the specificity of HIV protease cleavage activity. However, success in both extracting discriminant rules and maintaining high prediction accuracy is still challenging. The earlier study had employed genetic programming with a min–max scoring function to extract discriminant rules with success. However, the decision will finally be degenerated to one residue making further improvement of the prediction accuracy difficult. The challenge of revising the min–max scoring function so as to improve the prediction accuracy motivated this study.

Results: This paper has designed a new scoring function called a sum–product function for extracting HIV protease cleavage discriminant rules using genetic programming methods. The experiments show that the new scoring function is superior to the min–max scoring function.

Availability: The software package can be obtained by request to Dr Zheng Rong Yang.

Contact: z.r.yang@ex.ac.uk

INTRODUCTION

The human immunodeficiency virus (HIV) is the main causative agent of acquired immunodeficiency syndrome (AIDS) as a kind of retrovirus infecting several types of cells in the body. The main difference between a normal virus and a retrovirus is in the direction of flow of genetic information (Dahlberg, 1988). In a retrovirus, genetic information is stored in the form of RNA and the flow of genetic information is not in the routine order. During the life cycle of the retrovirus, the RNA is converted into DNA by a unique virus-specified enzyme called reverse transcriptase (Dahlberg, 1988).

The most-effective prevention of HIV infection would be a vaccine that blocks virus infection. Such a vaccine is difficult to develop because of the expense and complexity in advancing new candidate vaccines. A model that could achieve the goals of a more efficient and integrated HIV vaccine research enterprise was proposed (Klausner *et al*., 2003), but there is little hope that an HIV vaccine would be developed before 2009 (Kathryn, 2003).

Stopping viral replication in people who are already infected with HIV is then the most important alternative for fighting against HIV. Protease is one of the enzymes that HIV uses to reproduce itself and is a digestive enzyme that breaks down proteins. Enabling HIV protease cleavage inactive is therefore the major concern in medicine till now and protease inhibitor is a relatively recent form of an anti-viral agent.

The HIV protease has a crab-like shape, and is consisting of two molecules that are loosely associated. It has an extended binding region with eight consecutive residues of the polypeptide substrate in contact with active-site cleft (Miller *et al*., 1989). These eight consecutive residues are denoted by P_4–P_3–P_2–P_1–$P_{1'}$–$P_{2'}$–$P_{3'}$–$P_{4'}$ corresponding to the substrate S_4–S_3–S_2–S_1–$S_{1'}$–$S_{2'}$–$S_{3'}$–$S_{4'}$ in protease. A peptide of these eight residues is referred to as an 8mer in this study for convenience.

To design effective HIV protease inhibitors, accurately identifying cleaved HIV 8mers is very crucial. This identification process is based on the study of HIV protease specificity. However, the potential number of 8mers is 20^8 as there are 20 amino acids. This makes exhaustive experimental search impossible. On the other hand, it would be helpful and would expedite our pace in search of the proper inhibitors of HIV protease if we could find an accurate and rapid method for predicting the HIV protease cleavage sites in proteins (Chou, 1993a,b, 1996). In view of this, various computer prediction methods have been developed, such as the *h* function (Poorman *et al*., 1991), the vector-projection method (Chou, 1993), back-propagation neural networks (Cai and Chou, 1998), decision tree algorithms (Narayanan *et al*., 2002), bio-basis function neural networks (Thomson *et al*., 2003) and bio-support vector machines (Yang and Chou, 2004).

Despite the success of those algorithms, they suffer from some problems. The statistical methods bear a great dependence on data. The prediction accuracy was degraded due to the deficiency of data. The neural network models share the

*To whom correspondence should be addressed.

problems of having complicated structures and being affected greatly by noises. Furthermore, most of these models are all approximated by a black box approach and the underlying cleavage specificity can hardly be acquired for knowledge acquisition. The use of Genetic Programming (GP) was therefore proposed as a method of extracting discriminant rules (Yang *et al*., 2003).

GP is a branch of Genetic Algorithm (GA) (Goldberg, 1989). The GA is a model of machine learning which derives its behaviour from a metaphor of the processes of evolution in nature. This is done by the creation within a machine of a population of individuals represented by chromosomes. The individuals in the population then go through a process of evolution. GAs have proved to be a useful technique for finding solutions in a wide range of problem domains. Although a substantial amount of research has been performed on variable-length strings and other structures, the majority of work with GA is focused on fixed-length character strings.

The critical aspects to distinguish GP from GA are the fixed-lengthiness and the need to encode the representation of the solution. GP does not have a fixed-length representation and there is typically no encoding of the problem. The use of GPs flexible coding system allows it to perform structural optimization. Koza's GP algorithm was coded in LISP and has been applied to a wide range of problems, including symbolic regression (Koza, 1992), robotics (Koza and Rice, 1992), games (Eskin and Siegel, 1999) and classification (Loveard and Ciesielski, 2001).

A GP algorithm works on a population of individuals, each of which represents a potential solution to a problem. Initially, a population of random compositions of the functions and terms of the problem is generated. Next, each individual in the population is assigned a fitness value, which is a numeric value used to provide a measure of the appropriateness of a solution, i.e. how good the individual is at competing in its environment. Having selected candidate members of the population, some basic genetic operators are applied, which include reproduction, crossover and mutation. Then a new population is created. The best individual that appeared in the last generation is designated as the result of genetic programming.

A number of early results have demonstrated the potential applicability of GP to the field of biology. For example, GP has used to evolve a computer program to classify a given protein segment as being a transmembrane domain or non-transmembrane area of the protein (Koza and Andre, 1996a). In another application, a two-way algorithm that was evolved using GP for determining whether a protein was an extracellular protein, a nuclear protein, a membrane protein or an anchored membrane protein (Koza and Andre, 1996b). Koza has concluded that the single most important area for future work in GP was to demonstrate the applicability of the technique to realistic problems, and GP was suitable for an area where there was a large amount of data, in computer readable form, which required examination, classification and integration (Koza, 1997). Therefore, achievements can be expected from the innovative method of using GP in predicting HIV protease sites in proteins.

The earlier study had employed GP with a min–max scoring function to extract discriminant rules with success (Yang *et al*., 2003). A rule is a logic function of amino acids in a peptide. However, the decision is finally degenerated to one residue making further improvement of the prediction accuracy difficult. In this paper, we propose a new sum–product scoring function to replace the min–max scoring function. The use of the sum–product scoring function focuses on specificity, although it may lose some extent of generosity, it is still expected to improve the prediction accuracy.

SYSTEMS AND METHODS

Problem specification and notation

An 8mer is denoted by a vector $\mathbf{x} \in C^8$, where C is a set of 20 amino acids. Each 8mer is labelled positive if there is a cleavage site in it, otherwise negative. A set of 8mers is denoted by Ω. The set is divided into two parts. Ω_α contains non-cleaved 8mers and Ω_β cleaved ones, where

$$\Omega = \Omega_\alpha \cup \Omega_\beta \text{ and } \Omega_\alpha \cap \Omega_B = \phi.$$

The chromosome of a rule is denoted using a Reversed Polish Notation (RPN) as, $\mathbf{r} = [[\mathrm{x}]\mathrm{y}]$, where $[\circ]$ means that '$\circ$' can be repeated a couple of times, 'x' indicates a term and $y \in \{+, *\}$ an operator ($+$ means *or* operation and $*$ means *and* operation). Note that 'x' can take two types. First, it can be a residue followed by an amino acid. The residue takes one element from the set {a, b, c, d, e, f, g, h} corresponding to the eight residues {P_4, P_3, P_2, P_1, $P_{1'}$, $P_{2'}$, $P_{3'}$, $P_{4'}$}. Second, 'x' can be a sub-rule acting as a term. In the earlier study, each operator always has two terms, hence no parentheses are needed (Yang *et al*., 2003). To relax this constrain, parentheses are used in this study to allowing more than two terms for each operator. For example, a rule $P_2 = \mathrm{A}$ *and* $P_{1'} = \mathrm{K}$ and $P_{3'} = \mathrm{L}$ is (cAeKgL$*$) in this study rather than cAeK$*$ gL$*$ using the min–max scoring function.

Population initialization

The initial population is designed to contain M rules. M is an integer number and is 100 in this study. The distinction of each single rule in the initial population is very important. The 'ramped-half-and-half' method (Koza, 1992) is used, which supports greater population diversity. The maximum depth is four. The population is divided equally among individual trees of depth 2, 3 and 4 (max depth).

Quantitative method for measuring the relationship between 8mers and rules

To determine a proper quantitative method to measure the relationship between an 8mer and a rule is the most important

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
A	40	24	32	32	16	36	28	28	28	24	28	32	36	32	24	36	36	32	8	20
C	24	80	12	12	16	20	20	24	12	8	12	16	20	12	16	32	24	24	0	32
D	32	12	48	44	8	36	36	24	32	16	20	40	28	40	28	32	32	24	4	16
E	32	12	44	48	12	32	36	24	32	20	24	36	28	40	28	32	32	24	4	16
F	16	16	8	12	68	12	24	36	12	40	32	16	12	12	16	20	20	28	32	60
G	36	20	36	32	12	52	24	20	24	16	20	32	28	28	20	36	32	28	4	12
H	28	20	36	36	24	24	56	24	32	24	24	40	32	44	40	28	28	24	20	32
I	28	24	24	24	36	20	24	52	24	40	40	24	24	24	24	28	32	48	12	28
K	28	12	32	32	12	24	32	24	52	20	32	36	28	36	44	32	32	24	20	16
L	24	8	16	20	40	16	24	40	20	56	48	20	20	24	20	20	24	40	24	28
M	28	12	20	24	32	20	24	40	32	48	56	24	24	28	32	24	28	40	16	24
N	32	16	40	36	16	32	40	24	36	20	24	40	28	36	32	36	32	24	16	24
P	36	20	28	28	12	28	32	24	28	20	24	28	56	32	32	36	32	28	8	12
Q	32	12	40	40	12	28	44	24	36	24	28	36	32	48	36	28	28	24	12	16
R	24	16	28	28	16	20	40	24	44	20	32	32	32	36	56	32	28	24	40	16
S	36	32	32	32	20	36	28	28	32	20	24	36	36	28	32	40	36	28	24	20
T	36	24	32	32	20	32	28	32	32	24	28	32	32	28	28	36	44	32	12	20
V	32	24	24	24	28	28	24	48	24	40	40	24	28	24	24	28	32	48	8	24
W	8	0	4	4	32	4	20	12	20	24	16	16	8	12	40	24	12	8	100	32
Y	20	32	16	16	60	12	32	28	16	28	24	24	12	16	16	20	20	24	32	72

Fig. 1. Shows the Dayhoff matrix, where there are 20 rows and 20 columns. Each value means a mutation probability.

issue prior to using GP as it will be used to identify how good a rule is for classification. Since amino acids in these 8mers are non-numerical attributes, the relationship between an 8mer and a rule has to be measured using a scoring function based on biology measurements. A min–max function was therefore proposed by Yang *et al*. (2003) using amino acid similarity matrices (Dayhoff *et al*., 1978; Johnson and Overington, 1993; Henikoff and Henikoff, 1992, 1993).

The n-th 8mer is referred to as $\mathbf{x}_n$ and m-th rule $\mathbf{r}_m$. The d-th residue in $\mathbf{x}_n$ is referred to as x_{nd} and the d-th residue used in $\mathbf{r}_m$, r_{md}. In the min–max function, the minimum similarity score is found for *and* operation while the maximum similarity score is found for *or* operation

$$s(\mathbf{x}_n, \mathbf{r}_m) = \begin{cases} \min\{h(x_{nd}, r_{nd})\} & \textit{and} \\ \max\{h(x_{nd}, r_{nd})\} & \textit{or}, \end{cases} \quad (1)$$

where $h(x_{nd}, r_{nd})$ is the simialrity score between two amino acids x_{nd} and r_{md} using an amino acid similarity matrix with a table loo-up methods. Figure 1 shows a Dayhoff matrix (Johnson and Overington, 1993).

The min–max function enjoys a large extent of generosity, while it bears the problem of losing some specificity. In the sum–product function, all similarity scores calculated for each pair of amino acids from an input 8mer and a rule are summed together for *and* operation and the maximum similarity score is multiplied by the number of nodes (denoted as K) used in a rule for *or* operation.

$$s(\mathbf{x}_n, \mathbf{r}_m) = \begin{cases} \sum h(x_{nd}, r_{nd}) & \textit{and} \\ K h(x_{nd}, r_{nd}) & \textit{or}. \end{cases} \quad (2)$$

Table 1. Four artifical rules

No.	RPN expression	Rule
1	$\mathbf{r}_1 = (\text{aDcV}^*)$	$P_1 = D$ and $P_3 = V$
2	$\mathbf{r}_2 = (\text{aDcV}+)$	$P_1 = D$ or $P_3 = V$
3	$\mathbf{r}_3 = (\text{aDbEcV}^*)$	$P_1 = D$ and $P_2 = E$ and $P_3 = V$
4	$\mathbf{r}_4 = (\text{aDbEcV}+)$	$P_1 = D$ or $P_2 = E$ or $P_3 = V$

The RPNs are at the second column while the rules are at the 3rd column.

Table 2. A comparison between two scoring functions

	$\mathbf{r}_1$	$\mathbf{r}_2$	$\mathbf{r}_3$	$\mathbf{r}_4$
$\mathbf{x}_1$				
Min–max	48	48	48	48
Sum–product	96	96	144	144
$\mathbf{x}_2$				
Min–max	28	32	28	32
Sum–product	60	64	92	96

The scores are calculated using the Dayhoff matrix shown in Figure 1.

Given two 3mers with different functions: $\mathbf{x}_1 = \text{DEV}$, $\mathbf{x}_2 = \text{ATG}$ and four rules shown in Table 1. A comparison between two scoring functions is shown in Table 2, where the sum–product function shows better discriminating capability between $\mathbf{x}_1$ and $\mathbf{x}_2$ than the min–max function.

Fitness function

There are two important factors for a rule to work well. The first is the discriminating capability. A rule with a high

discriminating score is expected to work well, but may be too specific and loss of the generalization capability. The second is the complexity. A simple rule may cover many peptides with a low-discriminating capability. The Fisher ratio has been used for the first factor and the minimal description length has been used for the second factor in (Yang *et al*., 2003).

The Fisher ratio of the rule $\mathbf{r}_m$ is defined as

$$J(\mathbf{r}_m)\frac{|u_{m\alpha}-u_{m\beta}|}{\sqrt{\sigma_{m\alpha}^2+\sigma_{m\beta}^2}} \quad (3)$$

Note that $u_{m^*} = E < s(\mathbf{x}_n, \mathbf{r}_m) >$ and $\sigma_{m^*}^2 = \text{var} < s(\mathbf{x}_n, \mathbf{r}_m) >$, where $\forall \mathbf{x}_n \in \Omega_*$ and * is either α or β. The minimum description length of a rule $[\ell(\mathbf{r}_m)]$ is defined as the equivalent length of a rule, which is the number of the residues used in the rule. We use the normalized equivalent length for convenience, i.e. dividing the equivalent length by the maximum number of residues in peptides (Yang *et al*., 2003). For instance, the normalized equivalent length of the rule '(P_1 = F and $P_{1'}$ = P) or P_1 = L' is 0.25.

The fitness function (goodness) of a rule is defined as

$$F(\mathbf{r}_m) = \delta J(\mathbf{r}_m) + (1-\delta)\ell(\mathbf{r}_m) \quad (4)$$

$\delta = 0.7$ favours the discriminant capability in this study.

Operations

The three evolutionary operations are the same as those used in Yang *et al*. (2003). They are reproduction, crossover and mutation. The top 50% rules (called elites) with the highest fitness values in the current generation are copied to the next generation for mating. New 50% rules are generated based on the mating operations on the elites. A probability referred to as the mating probability is assigned to each elite according to the fitness function value using a sigmoid function $1/\{1+\exp[-F(\mathbf{r}_m)]\}$. An elite with a higher probability will have a higher opportunity to take part in the mating operation. Whenever an elite is selected for mating, a random number is generated to make a decision between crossover and mutation. The random value is an integer between 1 and 10. When the random is 9 or 10, crossover takes place. This means that 80% operations are mutation.

IMPLEMENTATION

The program was coded in Java on a PC containing a 2 GHz Pentium and Windows operating system.

DISCUSSION

A total of 362 HIV 8mers were collected from the earlier study (Cai and Chou, 1998), of which 114 were positive and the rest negative. A 10-fold cross-validation is used to compare two scoring functions. Each run is limited to 100 generation to prevent possibleoverfitting. Each rule is assigned five measurements. They are the true negative fraction (TNf or specificity), true positive fraction (TPf or sensitivity), total accuracy (Total), the Mattew's coefficient (MC) (Matthews, 1975) and fitness function value.

Figure 2 shows the measurements through the evolutionary generations. It can be seen that all these five measurements, specificity, sensitivity, total accuracy, the MC and fitness values are consistently increasing until approaching steady state after about 50 evolutionary generations.

Figure 3 shows the diversities for total accuracy and the MC through evolutionary generations. It can be seen that the SD was big at the beginning and was small at break the end.

Table 3 shows a summary of the comparison between two scoring functions based on the testing performance using the top 10 rules in cross-validation. It can be seen that the GP model using the sum–production function outperformed the GP model using the min–max function. The worst and the best rules obtained from the min–max function are `dFgLhD+dFdY++*` and `fFaW*aLbW*dF++aYbW* dF+fFaW*hFdD*cL*fFaW *fFaW*hFfD*cL*+cL*dF+++ dF+cL*dF+++hL*dF+cL *cI*dF+gFdD*cL*++`, respectively. Their equivalent lengths are 0.375 and 0.75, respectively. On the other hand, the worst and the best rules obtained from the sum–product function are `((fEdLgLfL*)((fEdLgVfL*)(cIaL(dYaP*)*)cC+)+)` and `(dFbM(cV(((gLcL+)aTdF*)fLdL*)+)+)`, respectively. Their normalized equivalent lengths are 0.625 and 0.75, respectively. The depths of the best rules are 18 and 4 for the min–max and sum–production scoring functions, respectively. Although the operators used by the best rules are 32 and 5 for the min–max and sum–production scoring functions, respectively. It can be seen that the sum–production function has generated much simpler rule structures compared with the min–max function. Figure 4 shows the best rule generated using the sum–production scoring function. It has also been noticed that the sum–product scoring function did not present higher total accuracy for the worst case as shown in Table 3. This is due to the fact that the sum–product scoring function presented more balanced prediction performance, i.e. the difference between the worst sensitivity and the worst specificity was 6.66% rather than 36.36% as presented using the min–max scoring function. As 69% of the HIV protease 8mers are negative (with no cleavage sites), higher specificity naturally leads to higher total accuracy. By losing 2.57% in total prediction accuracy for the worst case, the sum–product scoring function won 16.67% of the sensitivity.

The biological studies have indicated that P_1 and $P_{1'}$ are very important for cleavage activity. For instance,

- Tyrosine (Y) and Proline (P) are more conserved at P_1 and $P_{1'}$ for cleavage as observed among the matrix and capsid proteins (Hong, 1998). The replacement of the Tyrosine–proline bond with methionine–methionine reduces the cleavage efficiency (Cheng *et al*., 1991).

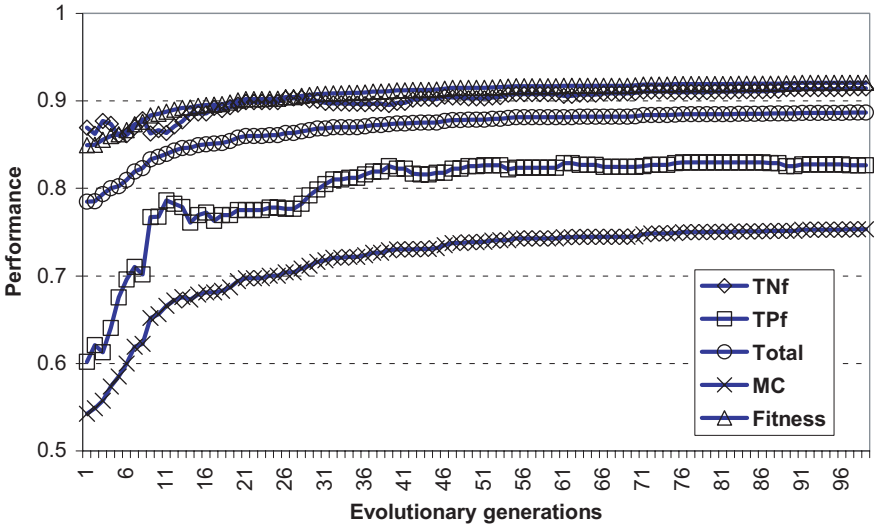

Fig. 2. Shows the five measurements through evolutionary generations. The horizontal axis represents the evolutionary generation while the vertical axis the performance. The diamonds, squares, circles, crosses and triangles are for specificity, sensitivity, total accuracy, the MC and fitness.

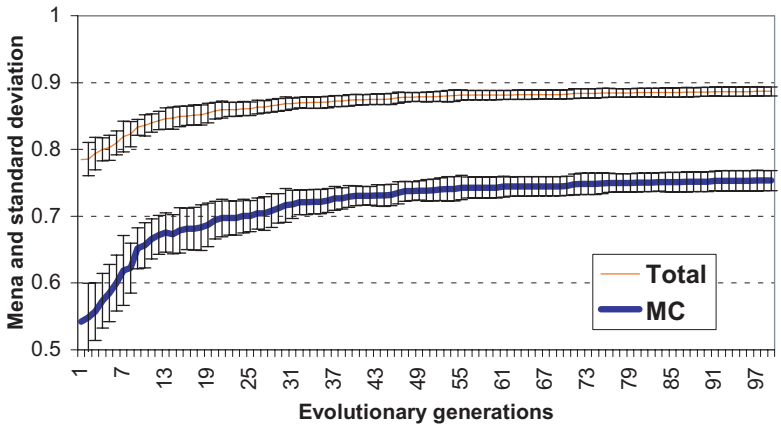

Fig. 3. Shows the diversities of two measurements through the evolutionary generations. The horizontal axis means the evolutionary generation while the vertical axis, the mean measurements with SD. The thin and the thick lines denote the total accuracy and the MC.

Table 3. Comparison of two scoring functions for the HIV case

	FIT	TNf (%)	TPf (%)	Total (%)	MC
Min–max					
Worst–best	0.69–0.72	86.36–88.89	50.00–94.74	75.00–91.89	0.39–0.84
Mean (SD)	0.71 ± 0.01	85.76 ± 4.88	77.68 ± 13.32	83.61 ± 4.56	0.62 ± 0.13
Sum–product					
Worst–best	0.80–0.98	73.33–95.83	66.67–100	72.43–96.88	0.47–0.93
Mean (SD)	0.89 ± 0.06	88.23 ± 6.50	79.84 ± 10.30	85.42 ± 6.64	0.70 ± 0.12

The measurements are obtained from the testing using top 10 rules from each cross-validation model. 'Worst' means the lowest total testing accuracy. 'Best' means the highest total testing accuracy. 'Mean' means the average among top 10 rules.

Table 4. Shows the conserved amino acids from two scoring functions

	Min–max								Sum–product							
	P_4	P_3	P_2	P_1	$P_{1'}$	$P_{2'}$	$P_{3'}$	$P_{4'}$	P_4	P_3	P_2	P_1	$P_{1'}$	$P_{2'}$	$P_{3'}$	$P_{4'}$
A									0.29							
C																
D																
E										0.20				0.29		
F	0.21			**0.37**	**0.22**	0.21		0.20				**0.55**	**0.43**			0.23
G																
H																
I			0.23								0.40					
K																
L			0.37	**0.24**								**0.23**				
M							0.20							0.20	0.21	0.23
N										0.20						
P					**0.26**								**0.21**			
Q		0.27														
R																
S					0.22											
T																
V										0.20					0.36	
W	0.20						0.28									
Y				**0.23**								**0.20**				

The numbers are frequencies of the amino acids occurred at the specified residues for the top 10 rules from 10 cross-validation models.

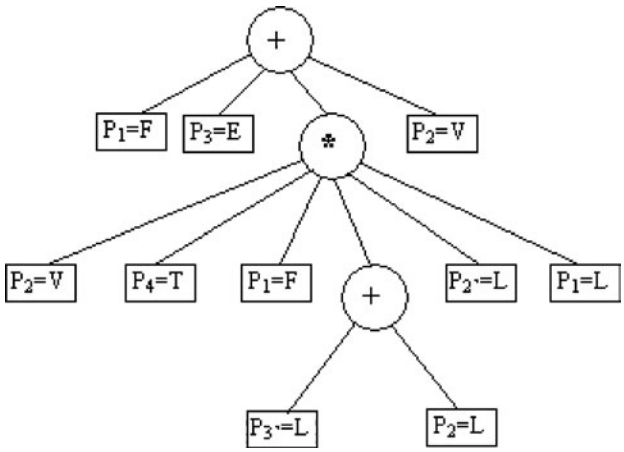

Fig. 4. Shows the best rule generated using the sum–product function. The rule shows the depth of 4 with the normalized equivalent length 0.75. Note that '+' means the *or* operation and '∗' the *and* operation.

- The elementary reaction properties of phenylalanine–proline (F–P) structure at P_1 and $P_{1'}$ have been studied (Reich *et al*., 1996; Fournout *et al*., 1997; Tran *et al*., 1997; Okimoto, 2000). The scissile bond (Phe–Pro) within the gag–pol polyprotein has been shown to be the competitive inhibitors of HIV-1 protease (Pivazyan *et al*., 2000).
- The specificity of phenylalanine–tyrosine (F–Y) bond has also been shown to play an important role for inhibition (Glenn *et al*., 2002; Kassel *et al*., 1995; Marastoni *et al*., 1998; Tossi *et al*., 1995).
- The other studies also addresses the importance of leucine–phenylalanine (L–F) bond in inhibitor design (Dreyer *et al*., 1992; Polgar *et al*., 1994; Carrillo *et al*., 1998; Hong, 1998).

After the rules are generated, the frequency that each amino acid occurs in top 10 rules is calculated for the investigation of whether the generated rules are consistent with biology science. Table 4 shows the frequencies with the values larger than 0.2. It can be seen that both scoring functions are able to explore the right conserved amino acids.

We have presented a new scoring function called the sum–product scoring function for extracting HIV protease cleavage discriminant rules using genetic programming. The simulation shows that this new scoring function is superior to the min–max scoring function. However, both the earlier study (Yang *et al*., 2003) and the current study aim to find a single rule to discriminate cleaved and non-cleaved 8mers. The principle of one-for-all may work for a small dataset where the heterogeneous is not too large. In order to enable the genetic programming working for a large dataset, multiple rules must be considered. In this case, the current single-population genetic programming method may not be suitable and complicated algorithms need to be investigated. On the other hand, a further comprehensive investigation of the comparison between discriminating rules and prototype rules as studied in

pattern recognition will bring this interesting subject deeper and wider in bioinformatics.

ACKNOWLEDGEMENT

The authors are grateful to the reviewers for their valuable advice and suggestion, which enabled the authors to strengthen the presentation of the work.

REFERENCES

Cai,Y.D. and Chou,K.C. (1998) Artificial neural network model for predicting HIV protease cleavage sites in protein. *Adv. Eng. Softw.*, **29**, 119–128.

Carrillo,A., Stewart,K.D., Sham,H.L., Norbeck,D.W., Kohlbrenner,W.E., Leonard,J.M., Kempf,D.J. and Molla,A. (1998) *In vitro* selection and characterization of human immunodeficiency virus type 1 variants with increased resistance to ABT-378, a novel protease inhibitor. *J. Virol.*, **72**, 7532–7541.

Cheng,T.R.J., Yin,Y.E. and Erickson-Viitanen,S. (1991) Mutagenesis of protease cleavage sites in the human immunodeficiency virus type 1 gag polyprotein. *J. Virol.*, **65**, 922–930.

Chou,J.J. (1993a) A formulation for correlating properties of peptides and its application to predicting human immunodeficiency virus protease-cleavable sites in proteins. *Biopolymers*, **33**, 1405–1414.

Chou,J.J. (1993b) Predicting cleavability of peptide sequences by HIV protease via correlation-angle approach. *J. Protein Chem.*, **12**, 291–302.

Chou,K.C. (1993) A vectorized sequence-coupling model for predicting HIV protease cleavage sites in proteins. *J. Biol. Chem.*, **268**, 16938–16948.

Chou,K.C. (1996) Prediction of human immunodeficiency virus protease cleavage sites in protens. *Anal. Biochem.*, **233**, 1–14.

Dahlberg,J.E. (1988) An overview of retrovirus replication and classification. *Adv. Vet. Sci. Comp. Med.*, **32**, 1–35.

Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1978) A model of evolutionary change in proteins. Matrices for detecting distant relationships. In Dayhoff, M.O. (ed.) *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Washington, DC, vol. 5, pp. 345–358.

Dreyer,G.B., Lambert,D.M., Meek,T.D., Carr,T.J., Tomaszek,T.A., Fernandez,A.V., Bartus,H. and Cacciavillani,E. (1992) Hydroxyethylene isostere inhibitors of human immunodeficiency virus-1 protease: structure–activity analysis using enzyme kinetics, X-ray crystallography, and infected T-cell assays. *Biochemistry*, **31**, 6646–6659.

Eskin,E. and Siegel,E.V. (1999) Genetic programming applied to Othello: introducing students to machine learning research. *Proceedings of 30th Technical Symposium of the ACM Special Interest Group in Computer Science Education*. New Orleans, LA, USA.

Fournout,S., Roquet,F., Salhi,S.L., Seyer,R., Valverde,V., Masson,J.M., Jouin,P., Pau,B., Nicolas,M. and Hanin,V. (1997) Development and standardization of an immuno-quantified solid phase assay for HIV-1 aspartyl protease activity and its application to the evaluation of inhibitor. *Anal. Chem.*, **69**, 1746–1752.

Glenn,M.P., Pattenden,L.K., Reid,R.C., Tyssen,D.P., Tyndall,J.D.A., Birch,C.J. and Fairlie,D.P. (2002) Beta-strand mimicking macrocyclic amino acids, templates for protease inhibitors with antiviral activity. *J. Med. Chem.*, **45**, 371–381.

Goldberg,D.E. (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, Reading, MA.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci., USA*, **89**, 10915–10919.

Henikoff,S. and Henikoff,J.G. (1993) Performance evaluation of amino acid substitution matrices. *Prot. Struct. Funct. Genet.*, **17**, 49–61.

Hong,L. (1998) Active-site mobility in human immunodeficiency virus, type 1, protease as demonstrated by crystal structure of A28s mutant. *Protein Sci.*, **7**, 300–305.

Johnson,M.S. and Overington,J.P. (1993) A structural basis for sequence comparisons—an evaluation of scoring methodologies. *J. Mol. Biol.*, **233**, 716–738.

Kassel,D.B., Green,M.D., Wehbie,R.S., Swanstrom,R. and Berman,J. (1995) HIV-1 protease specificity derived from a complex mixture of synthetic substrates. *Anal. Biochem.*, **228**, 259–266.

Kathryn,S. (2003) HIV vaccine still out of our grasp. *Lancet Infect. Dis.*, **3**, 457.

Klausner,R.D., Fauci,A.S., Corey,L., Nabel,G.J., Gayle,H., Berkley,S., Haynes,B.F., Baltimore,D., Collins,C., Douglas,R.G. *et al*. (2003) Medicine. The need for a global HIV vaccine enterprise. *Science*, **300**, 2036–2039.

Koza,J.R. (1992) *Genetic Programming on the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.

Koza,J.R. and Rice,J.P. (1992) Automatic programming of robots using genetic programming. *Proceedings of Tenth National Conference on Artificial Intelligence*. Menlo Park, CA, USA pp. 194–207.

Koza,J.R. and Andre,D. (1996a) Classifying protein segments as transmembrane domains using architecture-altering operations in genetic programming. In Angeline,P.J. and Kinnear,K.E.,Jr (eds), *Advances in Genetic Programming II*. MIT Press, Cambridge, MA.

Koza,J.R. and Andre,D. (1996b) Automatic discovery of protein motifs using genetic programming. In Yao,X. (ed.), *Evolutionary Computation: Theory and Applications*. World Scientific Press, Singapore.

Koza,J.R. (1997) Future work and practical applications of genetic programming. In Barck,T., Fogel,D.B. and Michalewicz,Z. (eds), and Oxford University Press, New York. *Handbook of Evolutionary Computation*. Institute of Physics Publishing Bristol, UK.

Loveard,T. and Ciesielski,V. (2001) Representing classification problems in genetic programming. *Proc. Congress Evol. Comput.*, **2**, 1070–1077.

Marastoni,M., Bortolotti,F., Salvadori,S. and Tomatis,R. (1998) Structure–activity relationships of HIV-1 protease inhibitors containing gem-diaminoserine core unit. *Arzneimittel-Forschung*, **48**, 709–712.

Matthews,B.W. (1975) Comparison of predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta*, **405**, 442–451.

Miller,M., Schneider,J., Sathyanarayana,B.K., Toth,M.V., Marshall,G.R., Clawson,L., Selk,L., Kent,S.B. and Wlodawer,A. (1989) Structure of complex of synthetic HIV-1 protease with substrate-based inhibitor at 2.3 Å resolution. *Science*, **246**, 1149–1152.

Narayanan,A., Wu,X. and Yang,Z.R. (2002) Mining viral protease data to extract cleavage knowledge. *Bioinformatics*, **18**, 1–18.

Okimoto,N. (2000) Protein hydrolysis mechanism of HIV-1 protease investigation by the *ab initio* MO calculations. *RIKEN Rev.*, **29**, 100–102.

Pivazyan,A.D., Matteson,D.S., Fabry-Asztalos,L., Singh,R.P., Lin,P.F., Blair,W., Guo,K., Robinson,B. and Prusoff,W.H. (2000) Inhibition of HIV-1 protease by a boron-modified polypeptide. *Biochem. Pharmacol.*, **60**, 927–936.

Polgar,L., Szeltner,Z. and Boros,I. (1994) Substrate-dependent mechanisms in the catalysis of human immunodeficiency virus protease. *Biochemistry*, **33**, 9351–9357.

Poorman,R.A., Tommasselli,A.G., Heinrikson,R.L. and Kezdy,F.J. (1991) A cumulative specificity model for proteases from human immunodeficiency virus types 1 and 2, inferred from statistical analysis of an extended substrate data base. *J. Biol. Chem.*, **266** 14554–14561.

Reich,S.H., Melnick,M., Pino,M.J., Fuhry,M.A., Trippe,A.J., Appelt,K., Davies,J.F., Wu,B.W. and Musick,L. (1996) Structure-based design and synthesis of substituted 2-butanols as nonpeptidic inhibitors of HIV protease: secondary amide series. *J. Med. Chem.*, **39**, 2781–2794.

Thomson,R., Hodgman,T.C., Yang,Z.R. and Austin,K.D. (2003) Characterising proteolytic cleavage site activity using bio-basis function neural networks. *Bioinformatics*, **19**, 1741–1747.

Tossi,A., Antcheva,N., Romeo,D. and Miertus,S. (1995) Development of pseudopeptide inhibitors of HIV-1 aspartic protease, analysis and tuning of the subsite specificity. *Pept. Res.*, **8**, 328–334.

Tran,T.T., Patinoa,N., Condoma,R., Frogiera,T. and Guedja,R. (1997) Fluorinated peptides incorporating a 4-fluoroproline residue as potential inhibitors of HIV protease. *J. Fluor. Chem.*, **82**, 125–130.

Yang,Z.R., Thomson,R., Dry,J., Hodgman,T.C., Wu,X.K., Doyle,A.K. and Narayanan,A. (2003) Extracting decision rules from protein sequences using genetic programming methods. *BioSystems*, **72**, 159–176.

Yang,Z.R. and Chou,K.C. (2004) Bio-support vector machines for computational proteomics. *Bioinformatics*, **20**, 903–908.