

# On the Use of MDL Principle in Gene Expression Prediction

Ioan Tabus and Jaakko Astola

Signal Processing Laboratory  
Tampere University of Technology  
P.O Box 553, SF-33101 Tampere, Finland

## Abstract

The structure and biological behavior of a cell are determined by the pattern of gene expressions within that cell. The so-called gene prediction problem refers to finding rules, or sets of possible rules, on how certain genes expressions determine the expression level of a given target gene. In this paper we investigate the gene prediction problem and propose the use of new predictors, selected according to the minimum description length (MDL) principle. We compare the use of Boolean predictors, ternary predictors and perceptron predictors. We resort to MDL as a tool for selecting the proper size of the prediction window. MDL is also well suited for comparing predictors having different complexities. We show that the best description can be achieved by the Boolean and ternary predictors, since they obtain better fitting of the data with a lower complexity of the model. To illustrate the comparison, both synthetic and experimental data are used.

## 1 Gene prediction problem

The concept of gene expression was introduced four decades ago with the discovery of messenger RNA, when the theory of genetic regulation of protein synthesis was described [5]. Each human cell contains approximately three billion base pairs, which encode about 30 000 to 40 000 genes. In any given cell only a small fraction of these genes is being actively transcribed. Genes carry the essential information for protein synthesis. Cell metabolism, in both healthy or malignant states, is governed by complex interactions between genes. A gene may have a varying level of expression in the cell at a certain time. Many biological crucial processes can be probed by measuring the gene transcription levels, and models of regulatory pathways can be inferred from these measurements [10].

The availability of cDNA microarrays [11] makes it possible to measure simultaneously the expressions level for thousands of genes. The huge amount of data provided by cDNA microarray measurements is explored in order to answer fundamental questions about gene functions and their inter-dependencies, and it is hoped to provide answers to questions like what is the type of the disease affecting the cells and what are the best ways to treat it. Large-scale study of gene expression is expected to provide the transition from structural to functional genomics [4], where knowing the complete sequence of a genome is only the first step in understanding how it works.

---

<sup>1</sup>This work has been supported by Academy of Finland, project No. 4876 (Finnish Center of Excellence Program (2000-2005))

As a first step in deciphering complex biological interactions at a biomolecular level it is necessary to study the microarray data in order to provide sets of possible rules on how the expressions of certain genes determine the expression level of a target gene.

Gene expression data obtained in microarray experiments may often be discretized as ternary data, the values 1,0,-1 carrying the meanings of overexpressed, normal, and repressed, respectively, which are the needed descriptors when defining regulatory pathways [2].

The basic problem addressed in this paper is the processing of ternary valued cDNA microarray data in order to find groups of genes, or factors, which are very likely to determine the activity of a target gene, as opposed to the oversimplified model of one (gene) factor influencing a single (gene) target. Previous solutions to this problem were given by using perceptron predictors, see [2], [6], which are shown to provide more reliable information than the determination of dependencies based solely on the correlation coefficient.

Our approach to prediction seeks for flexible classes of models, with good predictive properties, but also considers the complexity of the models as a penalizing factor, the balance between the two modelling aspects being set by the MDL principle.

## Modelling and data compression

Is a model telling cheaply enough the data generation mechanism? If yes, it will be cheaper to use the model and its residuals to describe the data instead of using the data in raw form.

A well established fact is the ability of the "codelength" to be a universal yardstick to measure the complexity of models. The MDL principle [1][7][8][9] considers the following axiom as basis to the theory of modeling: given the data and the model class, select the model in the model class which achieves the shortest codelength for the data and the model. The codelength necessary for encoding the data may be given by a two part code: first encode the model parameters, second encode the model residuals.

By applying compression techniques we seek for the (universal) code giving the shortest codelength of the available data. Modelling (data analysis) seeks for a description of the degree of dependency of one gene given the other genes and the conditions in the available data.

We postulate a parametric model for the dependency  $g_\ell = f(g_1, \dots, g_n)$ , fit its parameters to the available data, and find the codelength for sending the parameters and the residuals. We then claim that a dependency is present according to the reduction in the codelength obtained by a given model class. If different model classes are consistently signaling dependencies, they are very likely to be rules of the nature.

## Data available

The available data is organized in a matrix where row  $i$  is the set of measurements at instant  $i$ , and the column  $j$  ( $j = 1, \dots, n$ ) represents the activity of gene  $g_j$ . The entries  $M(i, j)$  take values in the set  $\{-1, 0, 1\}$ . In Table 1 we list the experimental data set, the same as the one used in [6].

If we had good estimates of the parameters of the stochastic model that corresponds to the data we could immediately derive estimates for the coding costs. However, collecting counts to evaluate frequencies, and from these to evaluate probabilities, is not a very safe method, because of the modest number of occurrences of the events of interest for us. Lacking the ability of accurate probability estimation, the entropies will not be good measures of average coding cost (which is the basic interpretation making entropy a measure of information content). Therefore it becomes natural to evaluate the coding length of a sequence by explicitly using a coding method.

## Encode without a model

We take as example the case of  $T = 30$  measurements, which we use in our experiments. Encoding "in clear" the column for gene  $g_\ell$  requires  $N = \log_2(3^{30}) = 47.55$  bits. Other codes for the target gene alone may be used, but we find it convenient to make no assumption about the distribution of  $-1, 0, 1$  at this point.

## Encoding the residuals

When we use a predictive model we have to decide on a code for the residuals. At a first glance, the residuals may be more difficult to code than the original sequence, since the dynamic range is now expanded to  $\{-2, -1, 0, 1, 2\}$ . However, there is an obvious reason why residuals may be easier to code: a good prediction will make the sequence of residuals to contain the symbol 0 many times. Using arithmetic coding, or simply run length coding, will produce, in the case of good predictors, a smaller codelength than that of the original sequence. However we use even a simpler code for prediction residuals, described next.

The residuals will be encoded in such a way to penalize the errors. For each error we send in clear the actual (correct) value, using the code "0" = 01, "1" = 10, "−1" = 11, the code 00 signaling the end of the residual sequence. After the actual value, we also transmit the location of the error using  $\log_2 31 = 5$  bits. An error will therefore be transmitted using 7 bits.

## 2 Comparison of prediction capabilities of several predictors

We consider here three predictor classes for ternary valued gene expressions: hard Boolean predictor, hard ternary predictor and finally, perceptron predictor. For all of them we discuss the predictor parameter estimation and the predictor order selection based on MDL principle. We are in the case of ternary valued data, with the ternary values  $\{0, 1, 2\}$  (if  $x \in \{-1, 0, 1\}$ , first use the mapping  $x \leftarrow x + 1$  to transform to the set  $\{0, 1, 2\}$ ).

### Hard Boolean predictor specification and design

Denote  $\underline{x} = [x_1, \dots, x_n]$  the prediction window, having dimension  $n$ , and  $x_i \in \{0, 1, 2\}$ . Define the thresholded vectors  $\underline{x}^{b_1} = [x_1^{b_1}, \dots, x_n^{b_1}]$ , and  $\underline{x}^{b_2} = [x_1^{b_2}, \dots, x_n^{b_2}]$ , with

$$x_i^{b_1} = \begin{cases} 0 & \text{if } x_i = 0 \\ 1 & \text{if } x_i \geq 1 \end{cases} \quad (1)$$

$$x_i^{b_2} = \begin{cases} 0 & \text{if } x_i \leq 1 \\ 1 & \text{if } x_i = 2 \end{cases} \quad (2)$$

The prediction is defined as

$$\hat{y} = f(\underline{x}^{b_1}) + f(\underline{x}^{b_2}) \quad (3)$$

where  $f(\cdot)$  is a Boolean function with  $n$  variables.

To design the Boolean predictor we quantize to two intervals the conditional expectation in the binary planes, and assign  $f^*(\underline{x}^b) = 0$  if  $\frac{E(y^{b_1}|\underline{x}^{b_1}=\underline{x}^b)+E(y^{b_2}|\underline{x}^{b_2}=\underline{x}^b)}{2} \leq 0.5$ , and  $f^*(\underline{x}^b) = 1$  otherwise.

RCH1	BCL3	FRA1	REL-B	ATF3	IAP-1	PC-1	MBP-1	SSAT	MDM2	p21	p53	AHA
-1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	0	0	0	1	1	1	1
-1	0	0	1	1	0	1	0	0	1	1	1	1
0	0	1	0	1	0	0	0	0	0	1	1	1
-1	0	0	1	1	1	1	1	0	1	1	1	1
0	0	0	0	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	1	0	0	0	0	0	1	1	1
0	0	0	0	1	0	0	0	0	0	1	1	1
-1	0	1	1	0	0	0	0	0	1	1	1	0
0	0	1	0	1	0	0	0	0	1	1	1	1
0	0	1	1	1	0	0	0	0	1	1	1	1
0	1	0	1	1	1	1	0	0	1	1	1	1
0	0	0	0	1	0	0	0	0	0	1	1	1
0	0	0	0	1	0	0	0	0	0	1	1	1
-1	1	1	1	1	0	1	0	0	0	0	-1	-1
0	0	0	0	1	0	0	0	0	0	0	-1	0
-1	1	0	1	1	0	1	0	1	0	1	-1	-1
0	0	1	0	1	0	0	0	0	1	1	-1	0
0	0	0	0	0	0	0	0	0	0	0	-1	0
0	0	0	0	1	0	0	0	0	0	0	-1	0
0	0	0	1	0	0	1	0	0	0	0	-1	0
0	0	0	0	1	0	0	0	0	0	1	-1	0
0	0	0	0	1	0	1	0	0	0	1	-1	0
-1	1	0	1	0	1	1	0	0	0	0	-1	-1
-1	0	0	0	1	0	0	0	0	0	1	-1	-1
-1	0	0	0	1	0	0	0	0	0	1	-1	-1
0	0	0	1	0	0	0	0	0	0	1	-1	0
0	0	0	0	1	0	0	0	0	0	1	-1	0
0	0	0	0	1	0	0	0	0	0	1	-1	0

Table 1: “12 Gene” Experiment Ternary expression data

### Hard Boolean predictor: Description length

We consider the two part code: first we encode the model “parameters” (we have to specify the function  $f^*(\cdot)$ ), and secondly, we need to encode the residuals (prediction errors).

The model cost for encoding the function  $f^*(\cdot)$  is

$$C_M(n) = 2^n \text{ bits} \tag{4}$$

if we assume a uniform apriori distribution over all Boolean functions. Other selections are also possible, i.e. to favor Boolean functions which are more active (e.g. depending on the absolute difference between number of ones and zeros in  $f$ ). Therefore if there are  $N_e$  nonzero prediction errors, the overall coding length will be the model length plus the prediction error length:

$$L(n) = 2^n + 7N_e + 2 \text{ bits} \tag{5}$$

By starting from a data set of several possible gene factors, we can select the best combination by designing sequentially the predictors with prediction window sizes  $n=2, 3, 4, 5$  and computing  $L(n)$  for each predictor. The MDL principle selects as optimal predictor size that  $n$  for which  $L(n)$  is minimum.

## Ternary predictor specification and design

The optimal Ternary predictor is found by quantizing to three intervals the conditional expectation:

$$\hat{y} = h^*(\underline{x}) = \begin{cases} 0 & \text{if } E(y|\underline{x}) \leq 0.5 \\ 1 & \text{if } 0.5 < E(y|\underline{x}) \leq 1.5 \\ 2 & \text{if } 1.5 < E(y|\underline{x}) \end{cases} \quad (6)$$

### Ternary predictor: Description length

To encode the model we have two possibilities.

1. The possible values of the prediction window  $\underline{x}$  can be ordered in lexicographic order as  $[0\ 0\ 0], [0\ 0\ 1], [0\ 0\ 2], \dots$ . We may specify the optimal ternary predictor by the string of values  $h^*([0\ 0\ 0]), h^*([0\ 0\ 1]), h^*([0\ 0\ 2]), \dots$ , which needs  $2 \cdot 3^n$  bits, therefore the cost of encoding the ternary model is  $C_M = 2 \cdot 3^n$  bits, the same for all models, irrespective of our data.

However, we note that we do not know the optimal values  $h^*(\underline{x})$  for those prediction windows  $\underline{x}$  which we have not seen.

2. Therefore, in the second alternative the cost of the model depends on the actual data, since we have to specify to the encoder the optimal predictions only for the seen prediction windows. Denoting  $n_x$  the number of different prediction windows found in the data set, the cost of encoding the model becomes  $C_M = 2n_x$  bits, and it is upper bounded by  $2T$  where  $T$  is the number of measurements. This variant is used in our experimental determinations. Note that a similar variant can be used for the Boolean predictor.

The prediction errors will be encoded the same way as in the case of Boolean prediction. Therefore if  $N_e$  nonzero prediction errors are obtained with the Ternary predictor, the code length for the prediction residuals is  $7N_e + 2$ .

The overall coding length will be the model length plus the prediction error length:

$$L(n) = 2n_x + 7N_e + 2 \text{ bits} \quad (7)$$

Apparently the ternary predictor is defined only for the prediction windows seen in the experiment. There is a obvious way to use the observed data to define  $h^*(\underline{x})$  for those prediction windows  $\underline{x}$  we have not seen. One simple way is to consider  $h^*(\underline{x}) = h^*(\underline{x}')$  where  $\underline{x}'$  is a seen prediction window, with size smaller than  $\underline{x}$ . However the point with the current gene expression problem is that our goal is to identify outstanding gene dependencies, given the observed data, which does not make necessary to specify the predictor for unseen windows.

### Perceptron predictor specification and design

The perceptron is found by quantizing to three intervals the best linear combination of samples in the predictor window:

$$\hat{y} = \begin{cases} 0 & \text{if } \underline{w}^T + w_0 \underline{x} \leq 0.5 \\ 1 & \text{if } 0.5 < \underline{w}^T \underline{x} + w_0 \leq 1.5 \\ 2 & \text{if } 1.5 < \underline{w}^T \underline{x} + w_0 \end{cases} \quad (8)$$

The parameters  $\underline{w}$  and  $w_0$  can be found using the Perceptron algorithm[6].

## Perceptron predictor: Description length

For ternary valued data, the numbers of all distinct perceptrons with two or three inputs are known[12]. Using a perceptron model with two input genes requires  $C_M = \log_2 471 = 8.88$  bits. A perceptron model with three input genes requires  $C_M = \log_2 85629 = 16.38$  bits.

### 3 An artificial example

We use first an artificial example, where, by knowing the true state of the nature, we will be able to check the success of our procedure.

We assume to have 30 measurements, at times  $i = 1, \dots, 30$ , of 40 variables (factors)  $x_1(i), \dots, x_{40}(i)$  and 1 target  $y(i)$ , and suppose the measurements are quantized to three levels: 0, 1, 2. The number of sequences of 30 symbols with ternary values is  $3^{30} = 2 \cdot 10^{14}$ , exceeding hugely the 40 sequences we have in the experiment. Suppose there is one function  $y(i) = f^*(x_{t_1}(i), x_{t_2}(i), x_{t_3}(i), x_{t_4}(i))$  which exactly describes the target, where  $f^* : \{0, 1, 2\}^4 \rightarrow \{0, 1, 2\}$ .

The data was generated by randomly choosing 40 sequences of 30 measurements, from all possible  $2 \cdot 10^{14}$  ternary sequences, with an uniform prior on all sequences. The target function  $f^* : \{0, 1, 2\}^4 \rightarrow \{0, 1, 2\}$  was selected in the following way: for any input window  $(j_1, j_2, j_3, j_4)$ , the value  $f^*(j_1, j_2, j_3, j_4)$  was sampled from the random variable with values 0, 1, 2 and mass  $p = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ . Then the target gene was constructed as the exact function  $y(i) = f^*(x_1(i), x_2(i), x_3(i), x_4(i))$ . Therefore the true window for the target is  $[x_1(i), x_2(i), x_3(i), x_4(i)]$ , referred for short as  $[1, 2, 3, 4]$ . In the following we show that Boolean and Ternary predictors can be used in conjunction with the minimum description length to select prediction window candidates.

**Target alone** The complexity of the target (no conditioning on other genes) is evaluated by three methods: (a) a priori uniform distribution on all sequences, giving  $L(y) = -\lceil \log_2 3^{30} \rceil = 48$  bits; (b) adaptive arithmetic coding, giving the value  $L(y) = 47$  bits; (c) Move to Front coding (which favors correlated sequence), giving the length  $L(y) = 52$  bits.

Since the target was generated as an uncorrelated sequence of ternary values, Move to Front gives rather different results than the first two methods, but we can safely assume that the complexity of the target alone is in the range 47–52 bits.

**Window size=2** No predictor using only two genes can achieve a description length lower than the complexity of the target. In the left Table 2 we list the best models of order two, those giving a descriptive length less than 72, which makes 17 windows in total. The "correct" window (which is of order four, and contains gene<sub>1</sub>, gene<sub>2</sub>, gene<sub>3</sub>, and gene<sub>4</sub>) has a "trace" in the list of the best models, occupying the second position with the window (gene<sub>1</sub>, gene<sub>3</sub>).

**Window size=3** The predictors using three genes becomes more successful in describing the target. In the middle Table 2 we list the best models of order three, those giving a descriptive length less than 55, which makes 28 windows in total.

**Window size=4** We list the best models of order four (those having length less than 49, which makes 23 windows in total). In bold we show the "true" model, which was found in the 19'th position, ranked out of 91390 possible prediction windows. Observe that several variations of the true window show up frequently as winners in the final list.

<i>Descr.Length</i>	$g_1$	$g_2$	<i>Descr.Length</i>	$g_1$	$g_2$	$g_3$
60.00	15	40	44.00	1	3	12
65.00	<b>1</b>	<b>3</b>	47.00	10	32	36
67.00	4	13	48.00	16	20	31
67.00	9	40	48.00	30	32	38
67.00	10	11	49.00	12	15	40
67.00	10	34	49.00	14	19	23
67.00	14	20	50.00	2	10	32
67.00	20	31	50.00	10	31	34
67.00	20	32	50.00	15	20	32
67.00	20	40	50.00	22	33	34
67.00	26	34	52.00	6	20	25
67.00	27	40	52.00	8	10	12
67.00	38	40	52.00	8	20	32
72.00	3	34	52.00	13	18	19
72.00	11	34	53.00	<b>1</b>	<b>2</b>	<b>3</b>
72.00	27	32	53.00	10	11	34
72.00	32	36	53.00	10	15	40

  

<i>Descr.Length</i>	$g_1$	$g_2$	$g_3$	$g_4$
46.00	1	2	3	12
46.00	1	3	12	38
46.00	12	30	32	38
46.00	30	32	34	38
46.00	8	10	31	34
46.00	1	12	32	38
46.00	12	30	32	38
48.00	1	3	12	18
48.00	10	27	32	36
48.00	4	16	20	31
48.00	2	10	25	32
48.00	2	10	29	32
48.00	10	25	31	34
48.00	15	20	25	32
48.00	15	20	32	40
48.00	6	20	25	40
48.00	8	20	32	34
48.00	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
48.00	10	11	13	34
48.00	15	20	32	40
49.00	1	3	12	34
49.00	12	16	20	31

Table 2: **Artificial data** The prediction windows of the best predictors of lengths two, three and four (the framed one of length four was used to generate the data)

## 4 A comparison of the three classes of models in a “12 gene” experiment

We consider here a comparison of the three predictors under MDL criterion, by using the experimental data from [6], listed in Table 1. The target gene is *AHA*.

In Table 3 we present the description length values, and also the number of errors, obtained with the best predictors in each of the three classes, for various sizes of the prediction window, when we restrain only to the set of genes (*RCH1, p53, PC-1*). For the Boolean predictor, the best description length is obtained by the model *HB(RCH1, p53)*. Appending *PC-1* to the prediction window does not improve the descriptive power of the model. The same conclusion can be drawn with Hard ternary predictors, and perceptrons. The MDL criterion makes therefore a clear and consistent selection of the best prediction window, while the coefficient of determination  $c_d$ [6] simply shows a better fit of the model to the data when increasing the window size. The problem with  $c_d$  is that it expresses only the fit to the training data, to avoid overtraining one needs another (validation) data set. MDL is a method which implicitly penalizes too large models, being shown to be a consistent order estimator, while several cross-validation methods are not.

We extend now the experiment to predicting the values of the target *AHA* based on combinations of two, three, or four of the genes in the first 12 columns of Table 1. In Tables 4, 5, 6 we show the best predictors for each window size. The best overall predictor is indicated to be  $AHA = HB(p53, ATF3, RCH1)$ , which has the Boolean function  $f(x_1, x_2, x_3) = x_1x_2 + x_2x_3$ , and the description length  $L = 10$ , much lower than that of any other window size, or combination of genes. By comparing the description length of predictors of different sizes, we can decide now that most of the “good” predictors of order 3 or 4 are worse than corresponding predictors of lower order e.g. all predictors of size three with description length 24 are actually worse than the predictor of order two  $AHA = HB(p53, RCH1)$ , which has description length 20.

We are capable therefore to organize the huge number of “good predictors” in Tables 4, 5, 6 and discard all those overly complex, to conclude that the relevant predictors in this experiments, in the light of the data. are only: *HB(p53, RCH1)*, *HB(p53, ATF3, RCH1)*, and *HT(p53, ATF3, BCL3)*.

## References

- [1] A. Barron, J. Rissanen, Y. Bin. The minimum description length principle in coding and modeling. *IEEE Trans. on Information Theory*, vol.44, no. 6, 2743–2760, Oct. 1998.
- [2] S. Kim, E.R. Dougherty. Coefficient of determination in nonlinear signal processing. *Signal Processing*, 80:2219–2235, 2000.
- [3] C.D. Giurcaneanu, I. Tabus, J. Rissanen. MDL based digital signal segmentation, *EUSIPCO-2000*, Tampere, Vol. 1, pp. 339-342, September 4-8, 2000.
- [4] P. Hieter, M. Boguski. Functional genomics: its all how you read it. *Science* 278, 601-602, 1997.
- [5] F. Jacob, J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, Vol. 3, 318-356, 1961.
- [6] S. Kim, E.R. Dougherty, Y. Chen, K. Sivakumar, P. Meltzer, J.M. Trent, M. Bitnner. Multivariate measurement of gene expression relationships. *Genomics*, 67, 201–209, 2000.

Hard Boolean Predictor			
	AHA=HBP(RCH1,p53,PC-1)	AHA=HBP(RCH1,p53)	AHA=HBP(p53)
$N_e$	2	2	12
$Length$	24	20	88
Hard Ternary Predictor			
	AHA=HT(RCH1,p53,PC-1)	AHA=HT(RCH1,p53)	AHA=HT(p53)
$N_e$	1	2	7
$Length$	25	24	55
Perceptron Predictor			
	AHA=Per(RCH1,p53,PC-1)	AHA=Per(RCH1,p53)	AHA=Per(p53)
$N_e$	2	2	7
$Length$	30.38	22.87	53.24
$c_d$ (MSE)[6]	0.946	0.785	0.624

Table 3: “**3 Gene**” **Experiment** Description length for three predictors : Hard Boolean Predictors (HBP), Hard Ternary Predictor(HT) and Perceptron Predictor (Per)

- [7] J. Rissanen. Modelling by shortest data description. Automatica, vol. 14, 465–471, 1978.
- [8] J. Rissanen. Universal coding, information, prediction and estimation. IEEE Trans. on Information Theory, vol.30, 629–636, Jul. 1984.
- [9] J. Rissanen. Stochastic complexity and modelling. Ann. Statist., vol. 14, pp. 1080-1100, 1986
- [10] P.J. Russel. Fundamentals of genetics. 2nd edition, San Francisco: Addison Wesley Longman Inc., 2000.
- [11] M. Schena, D. Shalon, R.W. Davis, P.O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 1995 October 20; 270: 467-470.
- [12] Y. Yamamoto and M. Mukaidono. Meaningful special classes of ternary logic functions–Regular ternary logic functions and ternary majority functions. IEEE Trans. Computers, Vol.37,no.7,799–806, July 1988.

Length	20	38	45
Gene <sub>1</sub>	p53	p53	p53
Gene <sub>2</sub>	RCH1	BCL3	ATF3
Method	HB	HT	HT

Table 4: “**12 Gene**” **Experiment** The Description length of the best predictors with window size =2.

Length	10	24	24	24	24	24	24	24	24	24	30	39	42
Gene <sub>1</sub>	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53
Gene <sub>2</sub>	ATF3	BCL3	FRA1	RELB	IAP1	PC-1	MBP1	SSAT	MDM2	p21	ATF3	ATF3	RELB
Gene <sub>3</sub>	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	BCL3	RELB	BCL3
Method	HB	HB	HB	HB	HB	HB	HB	HB	HB	HB	HT	HT	HT

Table 5: “12 Gene” Experiment The Description length of the best predictors with window size =3.

Length	18	18	18	18	18	18	18	18	18	27	27	29	29
Gene <sub>1</sub>	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53
Gene <sub>2</sub>	ATF3	ATF3	ATF3	IAP1	PC-1	MBP1	SSAT	MDM2	p21	PC-1	MBP1	PC-1	MBP1
Gene <sub>3</sub>	BCL3	FRA1	RELB	ATF3	ATF3	ATF3	ATF3	ATF3	ATF3	BCL3	PC-1	IAP1	FRA1
Gene <sub>4</sub>	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1
Method	HB	HB	HB	HB	HB	HB	HB	HB	HB	HT	HT	HT	HT

Length	29	29	29	30	30	30	30	31	31	31	32	32	32
Gene <sub>1</sub>	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53
Gene <sub>2</sub>	SSAT	SSAT	MDM2	MBP1	SSAT	MDM2	p21	FRA1	PC-1	p21	RELB	RELB	IAP1
Gene <sub>3</sub>	FRA1	PC-1	PC-1	IAP1	MBP1	MBP1	MBP1	BCL3	RELB	PC-1	BCL3	FRA1	BCL3
Gene <sub>4</sub>	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1
Method	HT	HT	HT	HT	HT	HT	HT	HT	HT	HT	HT	HB	HB

Table 6: “12 Gene” Experiment The Description length of the best predictors with window size =4.