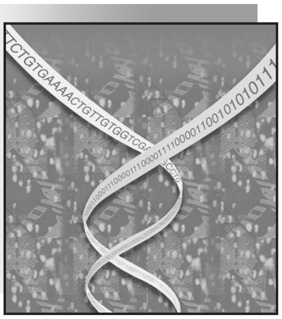


© EYEWIRE

Finding Large Domains of Similarly Expressed Genes

A Novel Method Using the MDL Principle and the Recursive Segmentation Procedure

BY DANIEL NICORICI, OLLI YLI-HARJA, AND JAAKKO ASTOLA

The advent of microarray technology enables us to measure simultaneously levels of thousands of genes for entire genomes in a single experiment—producing daunting amounts of data and genetic information. After the Human Genome Project ended in 2003 with the successful completion of the human genetic sequence [1], attention is turning to functional genomics. With gene expression data available for different organisms and their genomes already sequenced, a major goal is to understand the regulation of genes at the gene level and at the chromosomal level. Thus, an essential task is to define the role of the regulation mechanism and to understand how the regulation of a set of adjacent genes functions at the chromosomal level. In order to make biological sense of the sequenced genomes and the gene expression data that are available, all of the data must be organized in a manner that allows 1) the discovery of patterns that may arise, and 2) the establishment of relations between the gene expressions and the regulation mechanisms, e.g., the transcriptional regulation.

Recent studies [2]–[5] confirm that the genomes are divided into large domains that are composed of adjacent genes on chromosomes with similar expression profiles. There is evidence from budding yeast that some similarly expressed genes are found in adjacent pairs or triplets on chromosomes [3]. Larger domains are found to exist in the human genome [2], [4] due to the strong clustering of highly expressed genes in nearly all tissues. Also in their study, Spellman et al. [14] have found by analyzing the *Drosophila* genome and high-density oligonucleotide microarrays that its genes are clustered into coregulated groups of adjacent genes on chromosomes. The mechanism underlying the large domains is not yet known, but the observed similarities in the expression of adjacent genes are consistent with regulation at the level of chromatin structure [5]. The method used by Spellman et al. [5] does not provide a very good criterion for evaluating the quality of segmentation into large domains.

We describe a new method for finding large domains of similarly expressed genes using the minimum description length (MDL) principle and a recursive segmentation procedure. For the recursive segmentation, we introduce a new stopping criterion based also on the MDL principle. Based on the MDL principle, we give a rigorous definition of the quality of the segmentation of genomic profiles into large domains.

Intuitively, a large domain can be considered a group of adjacent genes on a chromosome, where the expression profiles of the genes are similar. This can be described in a succinct way by using the MDL principle, which has been introduced by Rissanen [6], [7]. The MDL principle has been used in statistics, machine learning, data mining [8], and genomic signal processing [9]–[11]. According to the MDL principle [6], the model is selected based on its fitting performance, but it also penalizes a very high complexity of the model.

Genomes can be divided into large domains that are important in controlling the expression of groups of adjacent genes [5]. The recursive segmentation can be used for finding their borders. The recursive segmentation methods have been applied to DNA segmentation into homogeneous domains; for finding the borders between coding and noncoding regions in DNA; for detecting the existence of the isochores, CpG islands, and replication origin and terminus; for detecting complex patterns such as telomers; and for evaluating the genomic complexity [12]–[14]. The criterion for continuing the recursive segmentation process can be based on 1) statistical significance [12], 2) the Bayesian information criterion (BIC) [12]–[14], or 3) the MDL principle [11], [15]. Our approach uses only the general properties of the large domains and, in this way, prior training on data sets is not necessary. The training data sets that contain the positions of large domains are not available. Also throughout this study, we define the genome data as the data containing measurements of gene levels versus experimental conditions, and the gene profiles are ordered according to their position on the chromosomes.

MDL Principle and Coding of Genome Data

Let X be a genome data represented as a $n \times m$ matrix where the row i represents the activity of the gene i (gene profile i) over different experimental conditions, and the column j represents the set of measurements for the experimental condition j. The genes in the genome data X are ordered according to their positions along the chromosomes and its entries $x_{i,j}$ take value in the set $\{0, 1, \ldots, q-1\}$ due to quantization of the genome data to q levels.

0739-5175/06/$20.00©2006IEEE

Our new segmentation method allows us to find large domains of similarly expressed genes without any a priori data for training.

MDL Principle

The MDL principle by Rissanen [6], [16] considers the description length of the data and the model as follows

$$\mathcal{L}(\mathcal{M}, X) = \mathcal{L}(\mathcal{M}) + \mathcal{L}(X|\mathcal{M}), \qquad (1)$$

where $\mathcal{L}(\mathcal{M})$ is the length of the description of the model and $\mathcal{L}(X|\mathcal{M})$ is the length of the description of the data, where the data X is described using the model $\mathcal{M}$. According to the MDL principle, the best model that fits the data is the model with the shortest length of the total description $\mathcal{L}(\mathcal{M}, X)$. Such a reduction indicates that the model $\mathcal{M}$ is able to capture the patterns and the dependencies within the data X. The goal is not to write down the encoded data but to compare the code length of the encoded data for a class of models. Thus, the model with the best fitting performance, which gives the shortest overall code length, is selected but in a balanced way; the models with a very high complexity are penalized. The MDL principle has been used in various applications [6], [9], [10].

Coding of Genome Data

The compression of a given genome data Y is done using the model $\mathcal{M}_1$, which takes into consideration the similarity between all the gene profiles from Y. The genome data Y is a $n^* \times m^*$ matrix that contains n^* genes across m^* different experimental conditions, where the entries $y_{i,j}$ take value in the set $\{0, 1, \ldots, q-1\}$ due to quantization of Y to q levels. The genes are ordered within Y according to their position on the chromosomes. Further, Y is considered to be a submatrix of the matrix X. We apply several transformations to the matrix Y such that (matrix Y) → (matrix Z) → (string $w^{n^{**}}$). The probability of the observed genome data Y is computed using the string $w^{n^{**}}$, which takes into consideration the similarity between all gene profiles within the genome data Y.

We construct a $q \times m^*$ matrix M such that entries $m_{i,j}$ are the counts of the symbol $(i-1)$ within the column j of the matrix Y. Thus, one has $\sum_{i=1}^{q} m_{i,j} = n^*$, where $j = 1, \ldots, m^*$. The symbols observed at each column j of the matrix Y are reordered by their counts from the matrix M using a permutation $v_j(\cdot)$, where $j = 1, \ldots, m^*$. The permutations are used because their coding requires a relatively short code length. A permutation aligns the histograms of symbols of each Y's column, such that all histograms are monotonically decreasing, and collapses all histograms into a single one [9]. A permutation $v_j(\cdot)$ maps $k \to v_j(k)$, where $k = 1, \ldots, q$, as follows

$$\begin{pmatrix} 0 & 1 & \ldots & q-1 \\ v_j(0) & v_j(1) & \ldots & v_j(q-1) \end{pmatrix}. \qquad (2)$$

The transformed matrix Z is obtained from the matrix Y by using a set of permutations $\boldsymbol{v} = (v_1(\cdot), \ldots, v_{m^*}(\cdot))$, where $z_{i,j} = v(y_{i,j})$. Such a transformation is reversible due to the use of permutation [9], i.e., one can recover Y from Z knowing $\boldsymbol{v}$. The matrix Z is transformed further into the string $w^{n^{**}}$ of length n^{**}, where $n^{**} = n^* \times m^*$, by concatenating its rows. The entries W of the string $w^{n^{**}}$ take value in the set $\{1, \ldots, q-1\}$. The entries of the matrices X, Y, and Z take value also in the same set $\{1, \ldots, q-1\}$. The transformed string $w^{n^{**}}$ is modeled further as a multinomial trial process with parameters $P(W=0) = \theta_0, \ldots, P(W=q-1) = \theta_{q-1}$. The symbol l is observed $\sum_{j=1}^{m^*} m_{v_j(l),j}$ times in the matrix Z and in the string $w^{n^{**}}$. Also, one can recover Y from $w^{n^{**}}$ knowing $\boldsymbol{v}$.

For instance, one has for $q = 2$, $n^* = 3$, $m^* = 2$, and

$$Y = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \text{ that } M = \begin{pmatrix} 1 & 3 \\ 2 & 0 \end{pmatrix}, v_1 = (1 \quad 0), v_2 = (0 \quad 1),$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, w^{n^{**}} = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0), \text{ and } n^{**} = 6.$$

To conclude, the probability of the genome data Y is given by

$$\begin{aligned} P(Y; \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}}) &= P(w^{n^{**}}(\widehat{\boldsymbol{v}}); \widehat{\theta}(w^{n^{**}}), \widehat{\boldsymbol{v}}(Y)) \\ &= \widehat{\theta}_0^{\sum_{j=1}^{m^*} m_{v_j(0),j}} \cdot \ldots \cdot \widehat{\theta}_{q-1}^{\sum_{j=1}^{m^*} m_{v_j(q-1),j}}, \qquad (3) \end{aligned}$$

where the set of permutations $\widehat{\boldsymbol{v}}(Y) = \{\widehat{v}_i(\cdot) : i = 1, \ldots, m^*\}$ determines the string $w^{n^{**}}$, and the multinomial parameters of the string $w^{n^{**}}(\widehat{\boldsymbol{v}})$ are $\widehat{\theta}(w^{n^{**}}) = (\widehat{\theta}_0(w^{n^{**}}), \ldots, \widehat{\theta}_{q-1}(w^{n^{**}}))$. The string $w^{n^{**}}$ contains m_i^* values of i, where $\widehat{\theta}_i(w^{n^{**}}) = m_i^* / n^{**}$ and $i = 0, \ldots, q-1$. Clearly, one has $m_i^* = \sum_{j=1}^{m^*} m_{v_j(i),j}$.

The overall code length of the encoded $n^* \times m^*$ matrix Y using the model $\mathcal{M}_1$ based on the MDL principle is as follows

$$\begin{aligned} \mathcal{L}(\mathcal{M}_1, Y) = & -\log_2 P(Y; \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}}) + m^* \log_2(q!) \\ & + \log_2 N_\theta(n^{**}, q), \qquad (4) \end{aligned}$$

where the first part encodes the data Y given the model $\mathcal{M}_1$ with the parameters $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{v}}$, the second part encodes the optimal permutations $\widehat{\boldsymbol{v}}$, and the third part encodes the maximum likelihood (ML) estimates of $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{v}}$. In order to encode efficiently, the set of probabilities, $N_\theta(n^{**}, q)$ is used because the set of pairs $(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}})$ is redundant, and one can restrict $\widehat{\boldsymbol{\theta}} = (\widehat{\theta}_0, \ldots, \widehat{\theta}_{q-1})$ such that $\widehat{\theta}_0 \geqslant \widehat{\theta}_0 \geqslant \ldots \geqslant \widehat{\theta}_{q-1}$ [9]. The length of the list containing all possible q-tuples

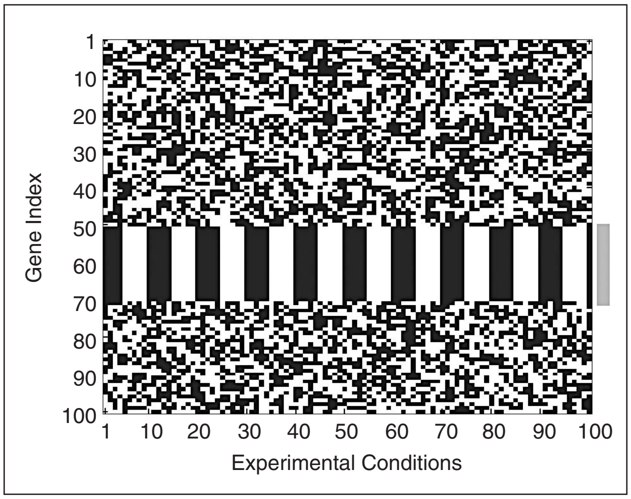


Fig. 1. Synthetic genome data containing 100 gene profiles across 100 experimental conditions. The first 50 gene profiles are randomly generated, the next 20 gene profiles are identical, and the last 30 gene profiles are randomly generated. The expression of genes have binary values.

$(n_0^{**}, \ldots, n_{q-1}^{**})$ is $N_\theta(n^{**}, q)$ such that $n_0^{**} + \cdots + n_{q-1}^{**} = n^{**}$ and $n_0^{**} \geqslant n_1^{**} \geqslant \ldots \geqslant n_{q-1}^{**}$ [9]. The model $\mathcal{M}_1$ considers that the entire given genome data Y is encoded as a single part, and no large domains of similarly expressed genes exist within Y. Thus encoding the genome data Y with few similarities between genes profiles will be penalized with a larger code length than in the case when there are more similarities between the gene profiles. We note that our approach of computing the code length of encoded data Y, especially of (3) and (4), is a modification of the approach used by Tabus et al. [9] for computing the code length of encoded class labels as a two-part code. The code length computed by Tabus et al. [9] is based on gene expressions of each patient, and it is used in the problem of class discrimination. The major difference between our approach and the approach of Tabus et al. [9] is that we compute the code length of the encoded genome data Y in such way to take into consideration the similarity between the gene profiles (the matrices M and Z are constructed in this way). Furthermore, the computed code length is used in the problem of segmentation into large domains of similarly expressed genes.

Coding of Genome Data with Large Domain of Similarly Expressed Genes

The encoding of a given genome data X is done also using the model $\mathcal{M}_3(i, j)$, which considers the existence of a large domain (i, j) containing similarly expressed genes. A large domain (i, j) starts with the ith gene profile and ends with the jth gene profile within matrix X, and it splits the matrix X into three submatrices $X^{(a)}$, $X^{(b)}$, and $X^{(c)}$, which contain the X's gene profiles from 1 to $i-1$, i to j, and $j+1$ to n, respectively. The submatrix $X^{(b)}$ is considered to be the only one that represents the large domain (i, j), and it cannot contain more than the a priori established maximum number of gene profiles. The number of genes contained in a large domain has been determined previously using biological experiments or data [2]–[5]. A large domain with similarly adjacent gene profiles gives a submatrix $X^{(b)}$ that is encoded very effectively using (4) based on the MDL principle.

According to the MDL principle, the overall code length of the encoded genome data X using model $\mathcal{M}_3(i, j)$ that considers the existence of a large domain (i, j) is

$$\mathcal{L}(\mathcal{M}_3(i, j), X) = \mathcal{L}\left(\mathcal{M}_1, X^{(a)}\right) + \mathcal{L}\left(\mathcal{M}_1, X^{(b)}\right) + \mathcal{L}\left(\mathcal{M}_1, X^{(c)}\right) + 2 \cdot \log_2 n, \qquad (5)$$

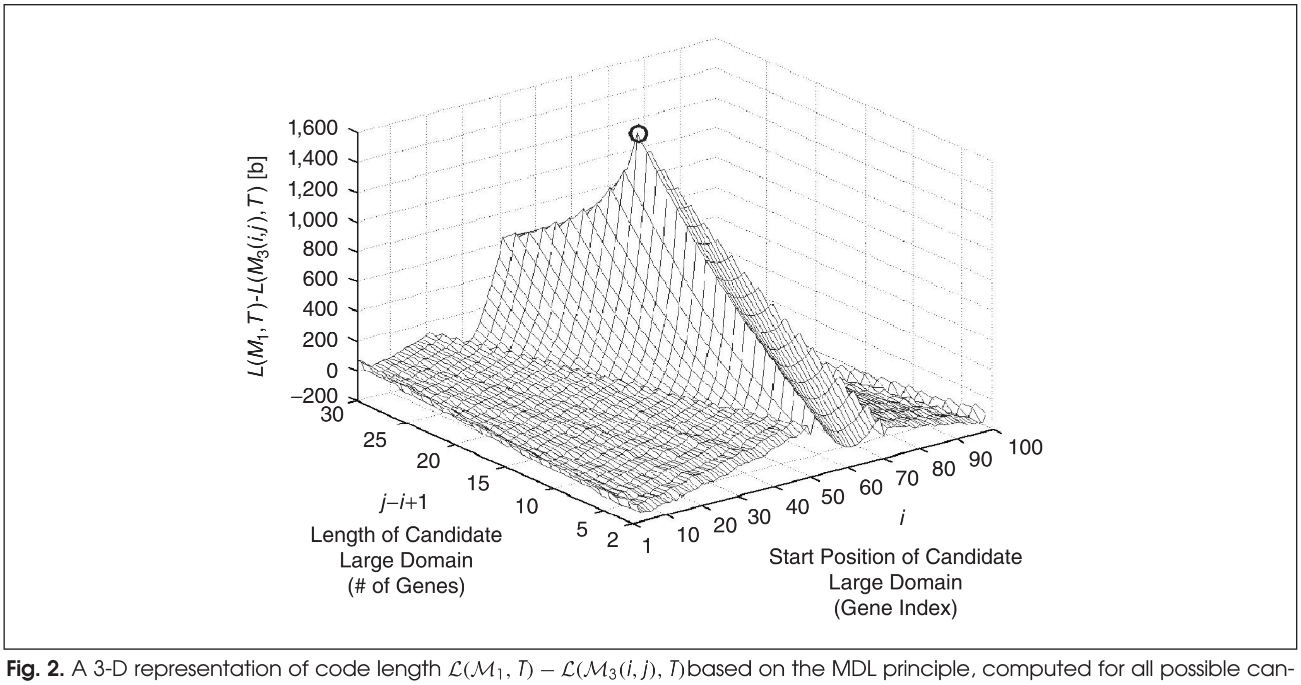


Fig. 2. A 3-D representation of code length $\mathcal{L}(\mathcal{M}_1, T) - \mathcal{L}(\mathcal{M}_3(i, j), T)$based on the MDL principle, computed for all possible candidate large domains (maximum length of 30) for synthetic genome data of 100 genes from Figure 1. The maximum value for the computed code length is circled on the graph and it corresponds to the large domain (51, 70).

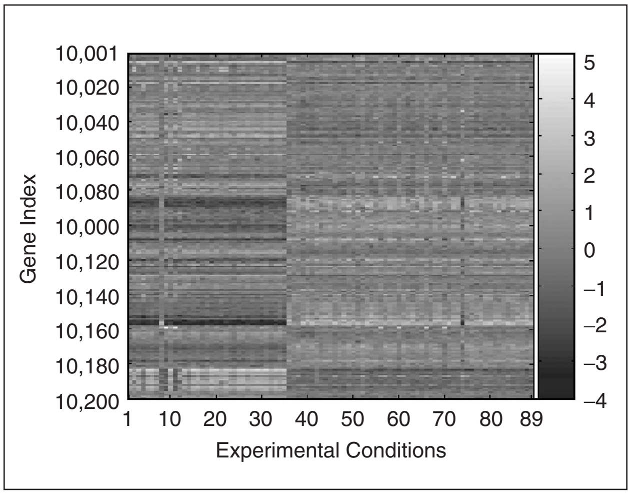


Fig. 3. Expression profiles of 200 adjacent genes on the right arm of the *Drosophila* chromosome 3 (3R).

where the first three terms $\mathcal{L}(\mathcal{M}_1, X^{(a)})$, $\mathcal{L}(\mathcal{M}_1, X^{(b)})$, and $\mathcal{L}(\mathcal{M}_1, X^{(c)})$ are computed using (4) based also on the MDL principle, and they give the cost in bits of encoding the submatrices $X^{(a)}$, $X^{(b)}$, and $X^{(c)}$, respectively. The last term $2 \cdot \log_2 n$ is needed to encode in bits the two positions where the large domain starts and ends within the matrix X. When it is assumed that no large domain exists within the genome the data X, the code length of the encoded matrix X is $\mathcal{L}(\mathcal{M}_1, X)$, and it is computed using the model $\mathcal{M}_1$ and relation (4).

Recursive Segmentation into Large Domains

In this study we use the recursive segmentation method proposed by Bernaola-Galvan et al. [12] and Li [13] for finding large domains of similarly expressed genes in the given genome data. The recursive segmentation of a given genome data X proceeds as follows. We sweep through the gene profiles of X and compute at every position i and j, where $i < j$, $i = 1, \ldots, n - h$ and $j = i + 1, \ldots, i + h$, that divide the matrix X into the upper submatrix $X^{(a)}$, the middle submatrix $X^{(b)}$, which represents the large domain (i, j), and the lower submatrix $X^{(c)}$, the code lengths of the whole matrix, the upper, the middle, and the lower submatrices. According to Spellman et al. [5], we choose the maximum length of a large domain to be $h = 30$ genes. The positions i and j are accepted as cutting points, representing the large domain (i, j), when the code length, $\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3(i, j), X)$, computed using (4) and (5), reaches its maximum. Further, we recursively apply the segmentation to the upper submatrix $X^{(a)}$ and to the lower submatrix $X^{(b)}$ until maximized code length $\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3, X)$ is above a certain threshold. In this approach, the threshold is based on the MDL principle, where

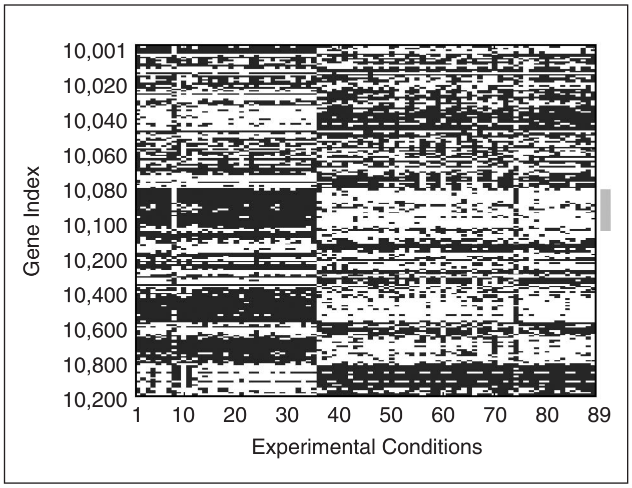


Fig. 4. Expression profiles of 200 adjacent genes (from Figure 3) on the right arm of the *Drosophila* chromosome 3 (3R) quantized to binary values. For each square, a black color denotes a lower relative expression than a white color for a gene in an experiment.

$$\begin{aligned} &\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3, X) \\ &\quad = \max_{(i,j)} (\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3(i, j), X)) \\ &\quad = \mathcal{L}(\mathcal{M}_1, X) - \min_{(i,j)} \mathcal{L}(\mathcal{M}_3(i, j), X). \end{aligned} \tag{6}$$

Clearly, one has $\mathcal{L}(\mathcal{M}_3, X) = \min_{(i,j)} \mathcal{L}(\mathcal{M}_3(i, j), X)$. If the maximized code length $\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3, X)$ is above the threshold, the genome data is segmented, and if not, the segmentation is stopped for the respective data. We note the

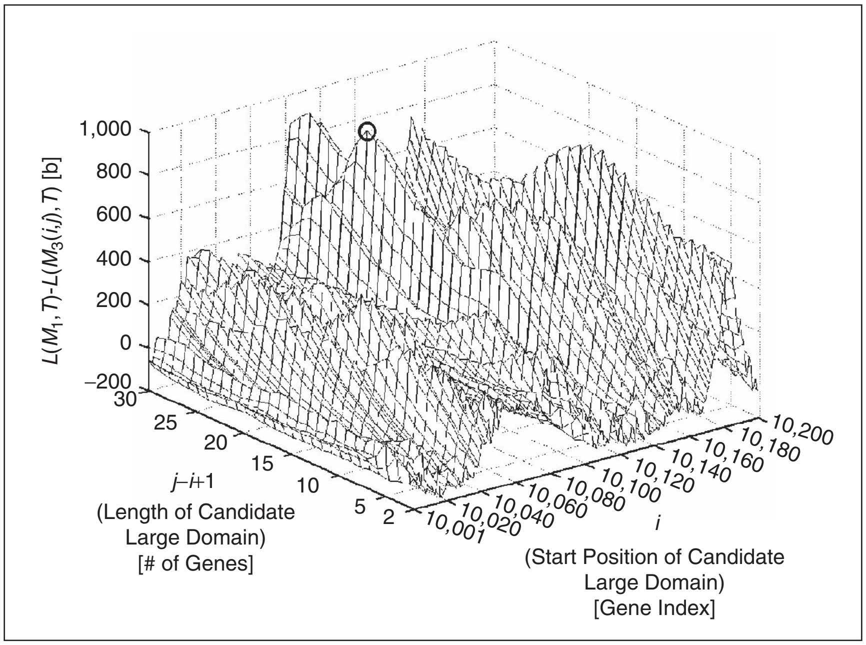


Fig. 5. 3-D representation of code length $\mathcal{L}(\mathcal{M}_1, \widetilde{X}) - \mathcal{L}(\mathcal{M}_3(i, j), \widetilde{X})$ based on the MDL principle, computed for all possible candidate large domains (maximum length of 30) for data of 200 adjacent genes, from the right arm of *Drosophila* chromosome 3 (3R), shown in Figure 4. The maximum value for the computed code length is circled on the graph, and it corresponds to the large domain (83, 104).

The model with the best fitting performance, which gives the shortest overall code length, is selected but in a balanced way.

similarity of this method with the approach when Jensen–Shannon divergence is used [13], [14].

Figures 1–2 shown synthetic data T of 100 gene profiles and the three-dimensional (3-D) representation of the code length $\mathcal{L}(\mathcal{M}_1, T) - \mathcal{L}(\mathcal{M}_3(i, j), T)$. The synthetic data T consist of 100 gene profiles that take binary values, where the first 50 gene profiles of data T are randomly generated, the next 20 gene profiles are identical, and the last 30 gene profiles are randomly generated. The maximum value for the computed code length is circled on Figure 2, and it corresponds to the large domain (51, 70). Figure 2 shows that the maximum value of the computed code length, $\mathcal{L}(\mathcal{M}_1, T) - \mathcal{L}(\mathcal{M}_3, T)$, finds exactly the start position and the length of the large domain with 20 identical gene profiles in the synthetic data T.

Stopping Criterion for Recursive Segmentation

The stopping criterion, in the case when relation (6) is used, can be considered from the point of view of hypothesis testing and the model selection framework. For the hypothesis testing framework, the probability that the value of $\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3, X)$ can be obtained by chance is

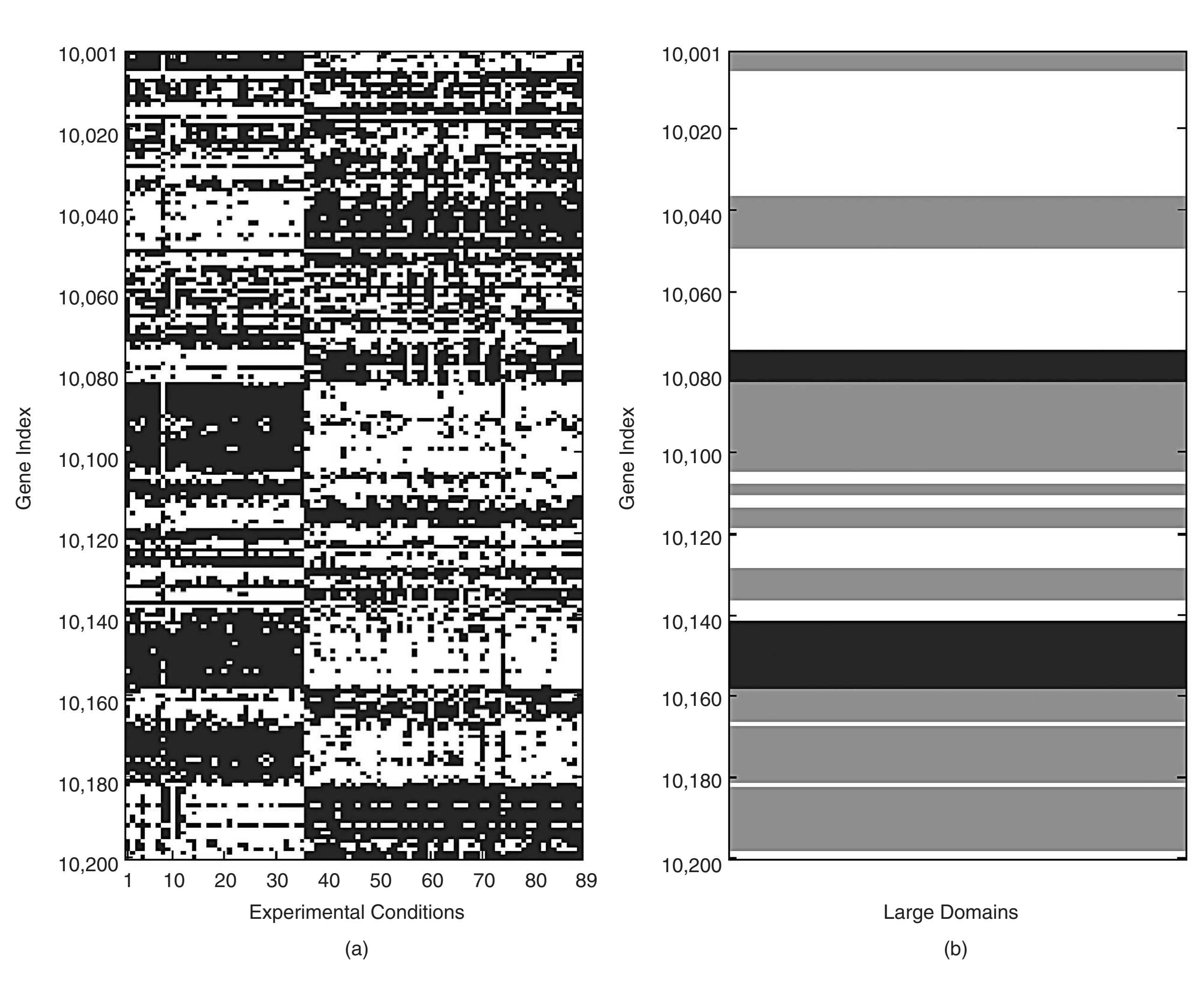


Fig. 6. Large domains of similarly expressed genes—gray or black rectangles in (b)—found using recursive segmentation applied to data of 200 adjacent genes from the right arm of *Drosophila* chromosome 3 (3R), shown in Figure 4.

computed by the null hypothesis that the genome data is homogeneous. The exact form of the null distribution is difficult to find [13].

The stopping criterion, based on Bayesian information criterion (BIC), for segmentation using Jensen–Shannon divergence has been introduced by Li in his study [13]. One can see the subtle relations between (6) and the Jensen–Shannon divergence or the Kullback-Leibler divergence [6]–[8], and also between the BIC and the MDL principle, which give an identical formula for certain cases [8].

In this study, we introduce a new stopping criterion for the recursive segmentation, when using the code length $\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3, X)$, based on model selection using the MDL principle. Thus, the stopping criterion tests if a three-random–submatrices model $\mathcal{M}_3$ gives a shorter code length than the one-random–matrix model $\mathcal{M}_1$. If the three-random–submatrix model has a shorter code length (it better fits the data), then the cuts are accepted (the large domain is accepted); otherwise, it is not. The MDL principle assures us of balancing the goodness-of-fit of the model with the complexity of the model in relations (4) and (5), and a very high complexity of the model is penalized. In order to continue the recursive segmentation procedure and to decide if the cuts *i* and *j* are significant or not (if the large domain (*i*, *j*) is significant), the three-random-submatrices model must fit the data better than the one-random model. This leads to a stopping criterion that is as follows

$$\mathcal{L}(\mathcal{M}_1, X) - \mathcal{L}(\mathcal{M}_3, X) > 0, \qquad (7)$$

where $\mathcal{L}(\mathcal{M}_1, X)$, $\mathcal{L}(\mathcal{M}_3, X)$ are computed using (4) and (6), respectively. Thus, the recursive segmentation continues, or the cuts *i* and *j*, which represent the large domain (*i*, *j*), are accepted as significant as long as criterion (7) is fulfilled.

Experimental Results

We illustrate the finding of large domains of similarly expressed genes based on the MDL principle and recursive segmentation using the *Drosophila* genome data of Spellman et al. [5], publicly available [17], and human genome data.

The microarray genome data of *Drosophila* contains 13,165 gene expression profiles, covering 89 distinct experimental conditions from 267 Affymetrix GeneChip *Drosophila* Genome Arrays [5]. The experimental conditions consist of adults and embryos which are visible in Figures 3 and 4 as a vertical line. The data are in $\log_2$ ratio format, all replicates are averaged, and the values are time zero corrected [5]. Data preprocessing and experimental conditions are described in detail in [5]. The genes in this dataset are organized according to their positions along the chromosome. Visual inspection of the data, as shown in Figures 3–4, reveals that groups of adjacent genes with similar expression patterns, which are not otherwise functionally related in any obvious way, appear frequently [5].

The starting point is the gene expression data X, also called genome data, where each entry $x_{i,j}$ indicates the expression level of gene i for experimental condition *j*. We make the assumption that the transcription machinery of a gene uses the expressed/not expressed or upregulated/downregulated states [18]. More precisely, we quantize each gene profile independently to binary states [18] by applying the Lloyd algorithm [9]. The quantization to discrete values of genome data can be viewed also as removing the noise from data [9]. In this study, the entries in X are quantized to $q = 2$ levels, but the newly introduced method for finding large domains can use more than two levels of quantization. For the remainder of the article, we assume that the genome data X is quantized to binary values (how many quantization levels are chosen is outside the scope of this article). The MDL principle can also be used to select an optimum q value as suggested in [9].

In order to illustrate the segmentation procedure, we apply the new segmentation method on a group, chosen arbitrarily from the chromosome 3R of *Drosophila*, of 200 adjacent genes, noted as $\widetilde{X}$. Figure 3 shows the original expression profiles of the group of 200 gene profiles from $\widetilde{X}$ that are ordered accordingly to their position along the chromosome. Also, Figure 4 shows the same 200 gene profiles from $\widetilde{X}$ after quantization to binary states using Lloyd algorithm, where a white color indicates a higher relative expression of a gene in an experiment than a black color. In Figure 3, and especially in Figure 4, are visible groups of adjacent genes that have similar expression profiles and the vertical separation between embryos and adults of *Drosophila*.

Figure 5 illustrates the 3-D representation of the code length $\mathcal{L}(\mathcal{M}_1, \widetilde{X}) - \mathcal{L}(\mathcal{M}_3(i, j), \widetilde{X})$, where the other two axes

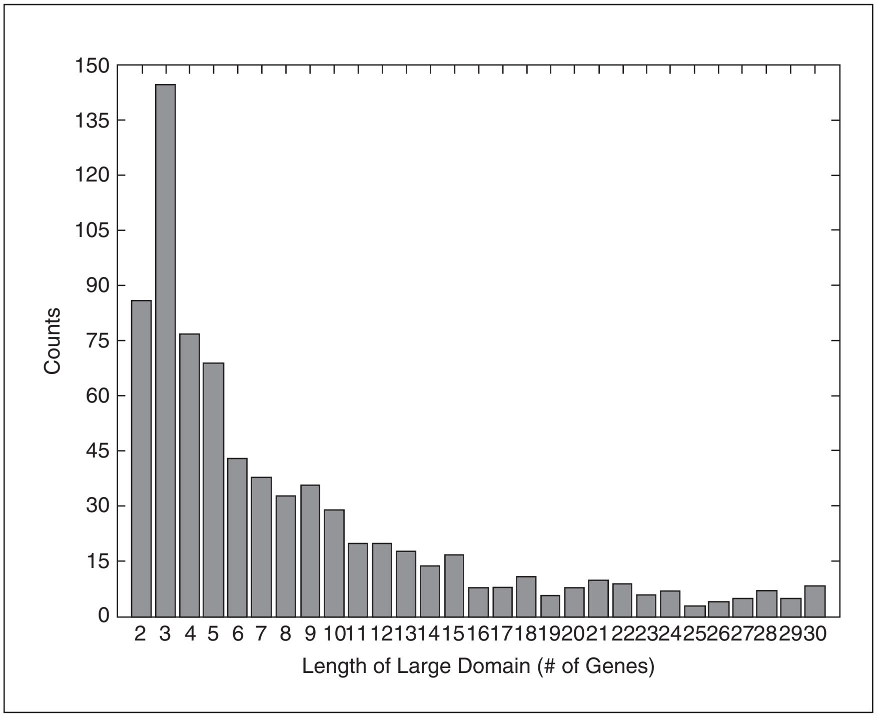


Fig. 7. The results of a recursive segmentation of *Drosophila* genome into large domains of similarly expressed genes.

Our new segmentation method permits us to find successfully large domains of similarly expressed genes without any use of a priori data for training.

represent i (position where the candidate large domain starts) and $j+i-1$ (length of the candidate large domain given as number of genes). The maximum value of $\mathcal{L}(\mathcal{M}_1, \widetilde{X}) - \mathcal{L}(\mathcal{M}_3(i, j), \widetilde{X})$ is circled on Figure 5 and it corresponds to the large domain (83, 104), which is marked in Figure 4 on the right-hand side as a gray-filled rectangle. This large domain is accepted as significant because $\mathcal{L}(\mathcal{M}_1, \widetilde{X}) - \mathcal{L}(\mathcal{M}_3(83, 104), \widetilde{X}) = 987.14$ b and $\mathcal{L}(\mathcal{M}_1, \widetilde{X}) - \mathcal{L}(\mathcal{M}_3(83, 104), \widetilde{X}) > 0$, where criterion (7) is fulfilled.

Figure 6 shows the results of the recursive segmentation applied to the same 200 adjacent gene profiles $\widetilde{X}$ from the *Drosophila* chromosome 3R. The new method introduced in this study, based on the MDL principle and recursive segmentation, is able to find successfully the large domains of similarly expressed genes in $\widetilde{X}$ as shown in Figure 6.

The new recursive method based on the MDL principle is applied to the genome of *Drosophila* [5] containing 13,615 genes, and it finds 750 large domains of similarly expressed genes together with their exact positions on the chromosomes. From these 750 large domains, 223 large domains are the domains that contain between 10–30 similar gene profiles. Figure 7 shows a histogram of the sizes of similar gene-profile segments that result when the new recursive segmentation method, based on the MDL principle, is applied to the *Drosophila* genome data.

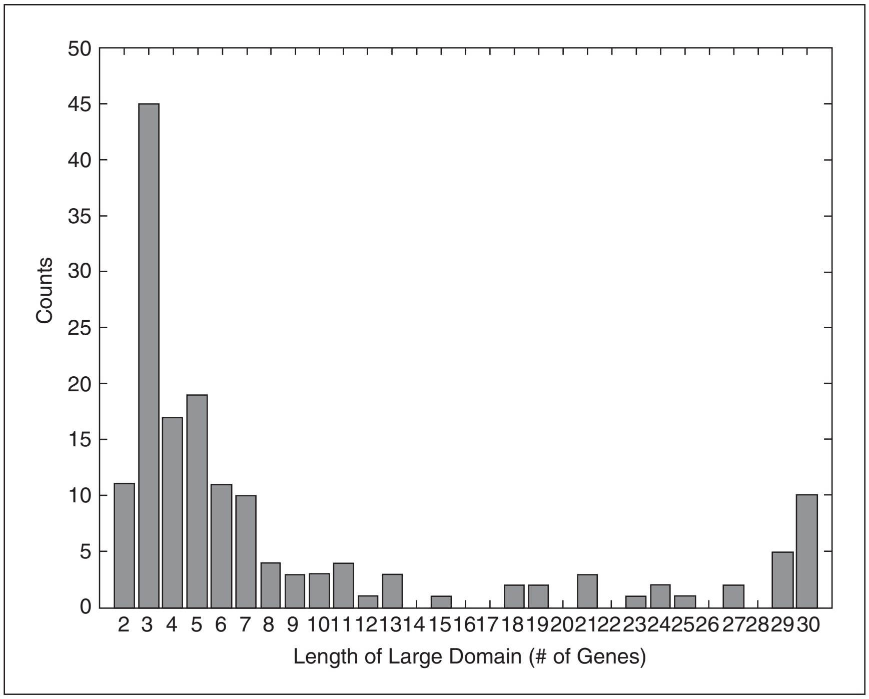


Fig. 8. The results of a recursive segmentation of the human genome into large domains of similarly expressed genes.

The human genome data contain 21,810 genes ordered according to their position on the chromosomes versus 50 patients with colorectal tumors. The Affymetrix HG-133A chips have been used for gene measurements. The gene profiles are quantized to binary using the Lloyd algorithm as done for the *Drosophila* genome data. The new recursive method finds 160 large domains of similarly expressed genes in human genome data. From these 160 large domains, 40 large domains are the large domains that contain between 10–30 similar gene profiles. Figure 8 shows a histogram of the sizes of similar gene-profile segments that result when the new recursive segmentation method is applied to the Human genome data.

Even though the biological significance of the detailed results for the *Drosophila* genome data and the human genome data remains to be later investigated and the underlying mechanism of the large domains is unknown [5], our new segmentation method permits us to find successfully large domains of similarly expressed genes without any use of a priori data for training.

The novel method introduced in this study, based on the MDL principle, for finding large domains of similarly expressed genes is different in several aspects from the method introduced in [11] for finding the large domains based on the MDL principle and normalized maximum likelihood (NML) model. The major differences are that in the current method, the quantization is done using the Lloyd algorithm and the similarity between all genes from a large domain are taken into consideration, which is closer to the biological knowledge available. In our previous study [11], only the similarities between the first gene profile and the rest of the gene profiles from a given large domain were taken into consideration.

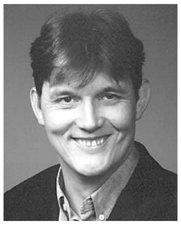
Concluding Remarks

In this study, we have introduced a new method for finding and defining large domains of adjacent genes on a chromosome with similar expression profiles based on the use of the MDL principle and the recursive segmentation procedure. For the recursive segmentation, we used a newly introduced stopping criterion using the MDL principle. Together they offer a novel method to view the large domains of similarly expressed genes in genome data. The description of the genome data and of the large domain is done according to the MDL principle, which selects the model based on its fitting performance and also penalizes a very high complexity of the model. The success of segmentation comes from the observation that the more similar the gene-expression profiles are in a large domain, the shorter the description of the data that represents the large domain. We have applied the new recursive segmentation method to the microarray measurements of the *Drosophila* genome and human genome in order to demonstrate the ability of the new method to find large domains successfully.

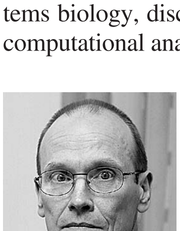
Acknowledgments

The authors thank Prof. Lauri A. Altonen and Diego Arango from the Department of Medical Genetics, Biomedicum Helsinki, Finland, for their valuable support and for providing the human genome data. Special thanks are also addressed to Christophe Roos from MediCel Ltd., Finland, for insightful discussions and references on large domains of similarly expressed genes.

Daniel Nicorici received his B.S. and M.S. degrees in electrical engineering from the Technical University of Cluj-Napoca, Romania, in 1999 and 2000, respectively. He received the Ph.D. degree in signal processing from Tampere University of Technology, Finland, in 2005. Since 2001, he has been with the Institute of Signal Processing, Tampere University of Technology, as a researcher. He is currently pursuing his postdoctoral studies at Tampere University of Technology. His research interests include genomic signal processing, bioinformatics, and computational systems biology.



Olli Yli-Harja received an M.Sc. in energy technology (1985) and a Ph.D. in computer science and applied mathematics (1989) from Lappeenranta University of Technology, Finland. His professional experience involves research and teaching in signal and image processing, computer science, and computational systems biology. Currently, he is a professor at the Institute of Signal Processing, Tampere University of Technology, Finland, leading a research group in computational systems biology. His research interests involve signal processing methods for systems biology, nonlinear signal processing, computational systems biology, discrete dynamic networks, image analysis, and computational analysis of music.



Jaakko Astola received his B.Sc., M.Sc., Licentiate, and Ph.D. degrees in mathematics (specializing in error-correcting codes) from Turku University, Finland, in 1972, 1973, 1975, and 1978, respectively. From 1976–1977 he was with the Research Institute for Mathematical Sciences of Kyoto University, Kyoto, Japan. Between 1979 and 1987, he was with the Department of Information Technology, Lappeenranta University of Technology, Finland, holding various teaching positions in mathematics, applied mathematics, and computer science. In 1984, he worked as a visiting scientist in Eindhoven University of Technology, The Netherlands. From 1987–1992 he was an associate professor in applied mathematics at Tampere University, Tampere, Finland. Since 1993, he has been professor of signal processing and director of Tampere International Center for Signal Processing, leading a group of about 60 scientists. He was nominated as an academy professor by the Academy of Finland (2001–2006). His research interests include signal processing, coding theory, spectral techniques, and statistics. He is a Fellow of the IEEE.

Address for Correspondence: Daniel Nicorici, Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere, Finland. E-mail: Daniel.Nicorici@tut.fi.

References

[1] F.C. Collins, M. Morgan, and A. Patrinos, "The human genome project: Lessons from large-scale biology,'' *Sci.*, vol. 300, no. 5617, pp. 286–290, 2003.

[2] H. Caron et al., "The human transcriptome map: Clustering of highly expressed genes in chromosomal domains,'' *Sci.*, vol. 291, no. 5507, pp. 1289–1292, 2001.

[3] B.A. Cohen, R.D. Mitra, J.D. Hughes, and G.M. Church, "A computational analysis of whole genome expression data reveals chromosomal domains of gene expression,'' *Nature Genetics*, vol. 26, no. 2, pp. 183–186, 2000.

[4] M.J. Lercher, A.O. Urrutia, and L.D. Hurst, "Clustering of housekeeping genes provides a unified model of gene order in the human genome," *Nature Genetics*, vol. 31, no. 2, pp. 180–183, 2002.

[5] P.T. Spellman and G.M. Rubin, "Evidence for large domains of similarly expressed genes in *Drosophila* genome,'' *J. Biol.*, vol. 1, no. 5, pp. 1–5, 2002.

[6] J. Rissanen, "Modeling by shortest data description,'' *Automatica*, vol. 14, pp. 465–471, 1978.

[7] J. Rissanen, "Strong optimality of the normalized ML codes as universal codes and information in data,'' *IEEE Trans. Information Theory*, vol. IT-47, no. 5, pp. 1712–1717, 2001.

[8] M.H. Hansen and B. Yu, "Model selection and the principle of minimum description length,'' *J. Amer. Statistical Assoc.*, vol. 96, no. 454, pp. 746–774, 2001.

[9] I. Tabus, J. Rissanen, and J. Astola, "Classification and feature gene selection using the normalized maximum likelihood model for discrete regression,'' *Signal Processing*, vol. 83, no. 4, pp. 713–727, 2003.

[10] I. Tabus, G. Korodi, and J. Rissanen, "DNA sequence compression using the normalized likelihood model for discrete regression,'' in *Proc. Data Compression Conf.*, Snowbird, UT, 2003, pp. 253–262.

[11] D. Nicorici, O. Yli-Harja, and J. Astola, "An MDL method for finding large domains of similarly expressed genes," in *Proc. Workshop Genomic Signal Processing and Statistics* (GENSIPS), Baltimore, Maryland, 2004.

[12] P. Bernaola-Galvan, I. Grosse, P. Carpena, J.L. Oliver, R. Roman-Roldan, and H.E. Stanley, "Finding borders between coding and noncoding DNA regions by an entropic segmentation method,'' *Physical Rev. E*, vol. 85, no. 6, pp. 1342–1345, 2000.

[13] W. Li, P. Bernaola-Galvan, F. Haghighi, and I. Grosse, "Applications of recursive segmentation to the analysis of DNA sequences,'' *Computers and Chemistry*, vol. 26, pp. 491–510, 2002.

[14] D. Nicorici and J. Astola, "Segmentation of DNA into coding and noncoding regions based on recursive entropic segmentation and stop-codon statistics,'' *J. Applied Signal Processing*, vol. 1, no. 1, pp. 81–91, 2004.

[15] M. Koivisto, M. Perola, T. Varilo, W. Hennah, J. Ekelund, M. Lukk, L. Peltonen, E. Ukkonen, and H. Mannila, "An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries," in *Proc. Pacific Symposium Biocomputing 2003 (PSB'03)*, Hawaii, 2003, pp. 502–513.

[16] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling,'' *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2743–2760, 1998.

[17] P.T. Spellman, and G.M. Rubin, Web supplement to: "Evidence for large domains of similarly expressed genes in Drosophila genome," *J. Biol.* [Online]. Available: http://www.fruitfly.org/expression/dse/

[18] L.A. Soinov, M.A. Krestyaninova, and A. Brazma, Web supplement for "Towards reconstruction of gene networks from expression data by supervised learning,'' *Genome Biol.*, vol. 4, no. R4 [Online]. Available: http://genomebiology.com/2003/4/1/R6